

# Print Wizard

## User Guide and Technical Manual

*Version 4.1 and later*

*March 26, 2013 Edition*



**Rasmussen Software, Inc.**

10240 SW Nimbus Ave., Suite L9, Portland, Oregon 97223

(503) 624-0360

[www.anzio.com](http://www.anzio.com)

[rsi@anzio.com](mailto:rsi@anzio.com)

*Copyright © 2004-2013 by Rasmussen Software, Inc., All Rights Reserved*

## Table of Contents

<b>PRINT WIZARD USER GUIDE.....</b>	<b>7</b>
1 PRINT WIZARD INTRODUCTION .....	8
1.1 What is Print Wizard?.....	8
1.2 Concept.....	8
1.3 Profiles and Services .....	11
1.3.1 Introduction to print profiles.....	12
1.3.2 Introduction to services .....	13
2 THE USER INTERFACE .....	14
2.1 Beginning to use Print Wizard.....	14
2.1.1 First-time use.....	14
2.2 Print Wizard User Interface.....	21
2.2.1 The main dialog.....	21
2.2.2 The status bar .....	21
2.2.3 Done .....	21
2.2.4 File name and history .....	21
2.2.5 The tool bar .....	21
2.2.6 The Menus.....	21
2.2.7 File : Open.....	22
2.2.8 File : Close .....	22
2.2.9 File : Print.....	23
2.2.10 File : Preview .....	24
2.2.11 File : Email .....	25
2.2.12 File : Fax .....	26
2.2.13 File : Generate PDF.....	26
2.2.14 File : Generate then... ..	26
2.2.15 File : Printer setup .....	27
2.2.16 File : Exit.....	28
2.2.17 Edit : Smart printing .....	28
2.2.18 Edit : Edit settings .....	29
2.2.19 Edit : Reset settings .....	29
2.2.20 Edit : Load profile .....	29
2.2.21 Edit : Save settings as user profile.....	30
2.2.22 Edit : Save settings as master profile.....	30
2.2.23 Edit : Remove master profile.....	31
2.2.24 View : View log .....	31
2.2.25 View : Log messages.....	32
2.2.26 View : Debug .....	32
2.2.27 View : Start minimized.....	32
2.2.28 View : Show toolbar.....	32
2.2.29 Tools : Setup email.....	33
2.2.30 Tools : Setup faxing .....	33
2.2.31 Tools : Setup profiles.....	35
2.2.32 Tools : Setup services .....	37
2.2.33 Tools : Foreground Service Wizard.....	41
2.2.34 Tools : Add service to Windows Startup.....	42
2.2.35 Tools : Windows Services wizard .....	43
2.2.36 Tools : Show Windows Services status .....	45
2.2.37 Tools : Register file extension .....	45
2.2.38 Help : About.....	46
2.2.39 Help : Enter license .....	46
2.2.40 Help : Print Wizard help.....	47
2.2.41 Help : Print Wizard manual .....	47
2.2.42 Help : Print Wizard updates at Anzio.com .....	47
2.2.43 Help : Examples .....	48
3 FOR DEVELOPERS.....	49
3.1 The Print Wizard Engine .....	49
3.2 Print Wizard DLL .....	50

3.3 WēPO (Print Wizard ActiveX) .....	50
<b>PRINT WIZARD TECHNICAL DOCUMENTATION .....</b>	<b>51</b>
4 BEHIND THE USER INTERFACE.....	52
5 BEHIND THE PRINT WIZARD ENGINE .....	54
6 USING COMMAND-LINE PARAMETERS .....	56
7 “BANG” COMMANDS .....	62
7.1 How “Bang” Commands Are Used .....	63
7.2 !ACROBAT .....	64
7.3 !FAX32.....	64
7.4 !MAPISEND .....	65
7.5 !PREVIEW .....	65
7.6 !PRINT.....	65
7.7 !PDF .....	67
7.8 !RUN.....	67
7.9 !WINFAX .....	67
7.10 An Example.....	67
8 INSIDE PROFILES.....	70
8.1 Default Settings.....	70
8.2 Master Profile.....	70
8.3 User Print Profiles.....	70
8.4 User Interface Profile.....	70
8.5 Extension Profiles.....	71
8.6 Partial Profiles .....	71
9 INSIDE SERVICES .....	73
9.1 Despool.....	74
9.1.1 How does it work?.....	74
9.1.2 Rules.....	74
9.2 FTPDespool.....	76
9.2.1 How does it work?.....	76
9.2.2 Rules.....	76
9.3 Listen .....	78
9.3.1 How does it work?.....	78
9.3.2 Rules.....	78
9.4 LPD.....	80
9.4.1 How does it work?.....	80
9.4.2 Owners .....	80
9.4.3 Rules.....	80
10 THE PRINT WIZARD ENGINE.....	83
10.1 Beginning Assumptions.....	83
10.2 Detecting Special File Types .....	83
10.3 Print Level Switching.....	84
10.4 Detecting Special File Types By Initial Content.....	84
10.5 Basic Text Handling.....	84
10.6 Auto-fit Logic .....	86
10.7 Mini-page Logic.....	87
10.8 Auto-rotation Logic.....	87
10.9 Character Set Issues .....	89
10.10 Automatic Font Switching and Diacritics .....	89
10.11 Auto Tab and Tab Handling .....	91
10.11.1 Classic tab handling.....	91
10.11.2 Auto tab handling .....	91
10.12 Print Wizard’s Preview.....	93
10.13 Programming with Print Wizard.....	93
10.13.1 Column Alignment with FS Characters.....	94
10.13.2 Print Wizard’s markup language, briefly.....	94
10.14 Photo and Image Printing.....	95

10.15	Printing Metafiles .....	97
10.16	Capturing a Spool (SPL) File .....	97
10.16.1	Locating the SPL Directory .....	97
10.16.2	The capture process .....	98
10.17	Scanner as Input File .....	98
11	USING PRINT WIZARD IN ANZIOWIN .....	101
12	PDF GENERATION .....	103
12.1	The Built-in PDF Generator .....	103
12.2	PDF Encryption .....	103
12.3	Things You Can Do with PDF .....	104
13	PCL TRANSLATION .....	106
13.1	The Built-in PCL Translation .....	106
13.2	Where Do PCL Files Come From? .....	106
13.3	How You Can Use PCL Translation .....	107
13.4	When TranslatePCL is OFF .....	107
13.5	When TranslatePCL is MAIN .....	108
14	PWML SYNTAX .....	109
14.1	Recognizing a Marked-up Document .....	109
14.2	Tag Basics .....	109
14.3	End Tags .....	110
14.4	The Bare Minimum .....	110
14.5	Units of Measure .....	111
14.6	Positioning .....	112
14.7	Positioning Relative to Margins .....	112
14.8	Global Filtering with Regular Expressions .....	113
14.8.1	The IN string .....	113
14.8.2	The OUT string .....	114
14.9	Default Margins and Alignment .....	114
14.10	Setting Page Dimensions and Margins .....	115
14.10.1	Paper names, bin names, etc. ....	116
14.10.2	Changing pagesize parameters in the middle of a job .....	116
14.10.3	Usage patterns in pagesize tags .....	116
14.11	Pagination and Linespacing .....	117
14.12	Multi-column (Label) Printing .....	117
14.13	Colors .....	119
14.14	Fonts .....	120
14.15	Automatic Font Selection .....	121
14.16	Font Size, Pitch, and Line Spacing .....	122
14.17	Horizontal Alignment .....	123
14.18	Word Wrap .....	123
14.19	Character Entities .....	124
14.20	Variable Width Spaces .....	127
14.21	Line-drawing Characters .....	127
14.22	<ArcText> for text on an arc .....	127
14.23	Rectangles .....	128
14.24	Lines .....	128
14.25	Images .....	128
14.26	Barcodes .....	130
14.27	Tables .....	131
14.28	Pass thru .....	131
14.29	Tail – Data to be added at end of job .....	131
14.30	Special tags for PDF generation .....	131
14.30.1	<Attach> .....	131
14.30.2	<Index> .....	131
14.30.3	<A> for Anchor .....	131
14.30.4	<Signature> .....	131
15	INIT. FILES .....	132

15.1 Using initialization files.....	132
16 OVERLAYS.....	133
16.1 Bitmap Overlays .....	133
16.2 Scanned Overlays .....	134
16.3 EMF and WMF Overlays.....	134
16.4 SPL Overlays .....	134
16.5 PCL Overlays.....	134
16.6 PWML Overlays.....	135
16.7 PDF Overlays .....	135
16.8 Comparison of Methods.....	135
<b>INDIVIDUAL PROGRAM DOCUMENTATION.....</b>	<b>136</b>
17 THE UI - PWUI.EXE .....	137
18 THE PRINT WIZARD ENGINE – PRINTWIZ.EXE.....	139
19 DESPOOL SERVICE – PWDESPOOL.EXE .....	141
20 FTP DESPOOL SERVICE – PWFTPDESPOOL.EXE.....	144
21 LISTEN SERVICE – PWLISTEN.EXE.....	147
22 LPD SERVICE – PWLPD.EXE.....	149
23 WINDOWS SERVICES – PWSERVICES.EXE.....	151
23.1 Managing Windows Services Manually.....	151
23.2 Managing Windows Services Through Windows.....	152
24 PRINT WIZARD EMBEDDED .....	155
25 PRINT WIZARD DLL – PRINTWIZ.DLL.....	157
26 WĒPO .....	161
26.1 The Web Page.....	162
26.2 The Print Job File.....	164
26.3 Samples.....	164
27 MAPISEND.....	166
27.1 What Is MAPI?.....	166
27.2 How do you set/change MAPI options?.....	166
27.2.1 Mail.....	166
27.2.2 Internet options.....	168
27.3 Running MapiSend.....	168
27.4 How it works – Notes on Various Configurations .....	169
27.4.1 Microsoft Outlook e-mailing .....	169
27.4.2 Outlook Express e-mailing .....	170
27.4.3 Netscape Messenger e-mailing.....	170
27.4.4 Microsoft Outlook faxing .....	171
27.4.5 Outlook Express and Netscape Messenger faxing .....	171
27.4.6 Others .....	171
27.4.7 Examples .....	171
27.5 Errors.....	172
28 NETPRINT.....	174
<b>APPENDICES .....</b>	<b>176</b>
APPENDIX A PWML TECHNICAL REFERENCE.....	176
APPENDIX B OVERRIDING PRINT WIZARD DEFAULTS.....	190
APPENDIX C PRINT WIZARD EXAMPLES.....	191
APPENDIX D WĒPO EXAMPLES .....	195
APPENDIX E MORE DEVELOPER EXAMPLES .....	197
E.1. The Print Wizard Engine .....	197
E.1.1. Script example.....	197
E.1.2. “Bang” command examples .....	198
E.2. The Print Wizard DLL.....	202
E.2.1. Visual basic example.....	202
E.2.2. Delphi example.....	203
E.3. WĒPO ActiveX.....	205

APPENDIX F PRINT WIZARD SUPPORT FILES .....	207
APPENDIX G PRINT WIZARD SERVICES .....	208
G.1.    Program Calls .....	208
G.2.    Handling of work files .....	208
G.3.    FTP Despool Files.....	210
APPENDIX H WINDOWS SERVICES.....	211
H.1.    Program Calls .....	211
H.2.    Handling of work files .....	211
APPENDIX I LOG FILES AND DEBUG INFORMATION.....	212
I.1.    Debug Flags .....	212
I.2.    Log Files.....	212
APPENDIX J PRINT WIZARD SUPERFILES .....	213
APPENDIX K EMAILING .....	214
K.1.    CDO .....	215
K.2.    MAPI .....	215
K.3.    Outlook.....	216
K.4.    SMTP.....	216
APPENDIX L FAXING .....	218
L.1.    Fax32.....	218
L.2.    MAPI .....	219
L.3.    WinFax PRO.....	220
APPENDIX M WINDOWS DEFAULT PRINTER ISSUES .....	222
APPENDIX N WINDOWS FILE ASSOCIATIONS & SMART PRINT .....	224
N.1.    File Type Associations.....	224
N.2.    Smart Print .....	225
APPENDIX O APPLICATION NOTES .....	226
O.1.    Print Wizard and FilePro .....	226
O.1.1.    Installation and use – Windows .....	226
O.1.2.    Installation and use – Unix/Linux.....	226
O.1.3.    Usage notes .....	227
O.2.    Print Wizard and III Millennium and Innopac .....	228
O.2.1.    Connecting to Innopac via attached print .....	228
O.2.2.    Connecting to Millennium or Innopac via Workstation JetDirect .....	228
O.2.3.    Character sets .....	229
O.2.4.    What Print Wizard Will Do .....	229
APPENDIX P EXPANDABLE VARIABLES .....	230
APPENDIX Q PRINT PREVIEW, LABELS, INK & MORE.....	231
Q.1.    Drag to Resize .....	231
Q.2.    Labels .....	231
Q.3.    Ink and Drawing in Preview.....	233
APPENDIX R TROUBLESHOOTING .....	234

# **Print Wizard User Guide**

# 1 Print Wizard Introduction

## 1.1 What is Print Wizard?

Print Wizard is a printing utility for Windows. It comes in many forms and includes many features associated with processing the print output of other programs.

Print Wizard can receive a print job from a number of different sources, including local or network-derived files and UNIX/Linux-type systems. Print Wizard can output to many different targets, including printers, files, PDF files, and fax software. It can also email the print job or pass it off to another third-party application.

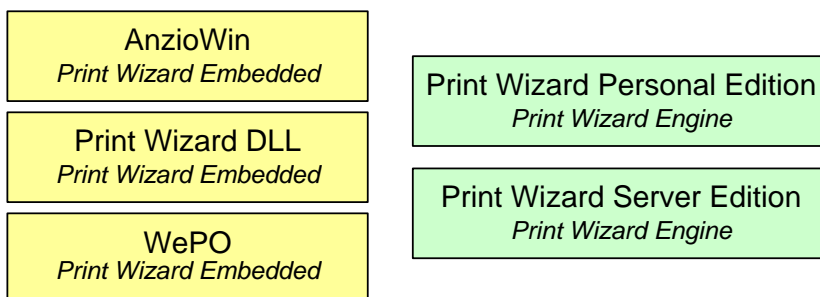
Print Wizard does all this while remaining totally printer independent, handling all the differences in printers and Windows printer drivers for you.

Print Wizard contains many advanced features such as automatically fitting print on a page, handling custom paper sizes and providing support for high-end printer features. It includes a printer markup language that is totally printer-independent, and bridges the gap between basic application output and modern-day printers.

## 1.2 Concept

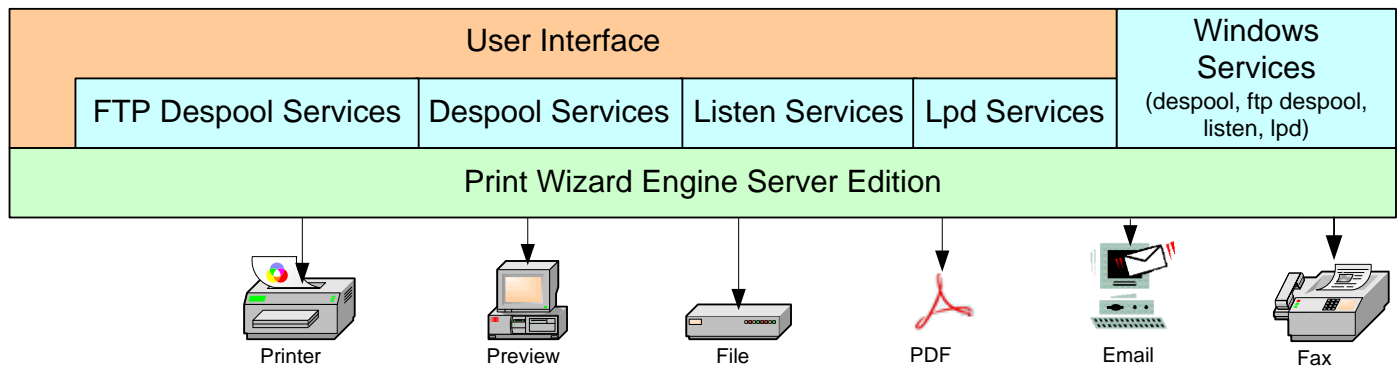
Print Wizard is a print processing engine that runs on Windows (95 and later). This engine, and its related support programs, take incoming print data, format it as necessary, and output it to a designated target.

There are two basic Print Wizard engines, Print Wizard Embedded and the Print Wizard stand-alone engine. Most of this document covers the Print Wizard Personal Edition and Service Edition software packages. However, features talked about for the Print Wizard engine also pertain to the embedded version.



There are several support programs which use the Print Wizard engine to produce quality output.





**User Interface** Print Wizard User's Interface.

**Despool Services** Watch a specific directory for the appearance of print files.

**FTP Despool Services** Watch a specified directory on a remote FTP server for files to print.

**Listen Services** Listen for incoming print jobs from a remote host, similar to the HP JetDirect protocol.

**LPD Services** Listen for print jobs over the standard UNIX/Linux LPR/LPD protocol.

**Windows Services** Run the "despool", "listen", and "LPD" procedures through a standard Windows service.

Print Wizard can process text that comes to it in a variety of ways:

- From a host system, via passthrough print (in AnzioWin).
- From a PC disk file.
- Fetched from an FTP or HTTP (web) server.
- Despooled from a PC directory.
- From Windows Explorer, as the designated print agent for a particular file type.
- Via Windows drag-and-drop.
- From a host system, via the LPR/LPD protocol.
- From a host system, using the "JetDirect" listen protocol.
- From a Windows printer driver (on the same or a different PC) using the LPR/LPD protocol.
- From a Windows printer driver (on the same or a different PC) using the "JetDirect" listen protocol.
- From a web page.
- From a controlling program feeding the Print Wizard DLL.

Print Wizard can take as input:

- Plain text files.
- Text files with form feeds, tabs, backspace-bolding, and backspace-underlining.
- Unicode text files.
- Text files with embedded PCL control (escape) sequences (PCL files).
- Text files with embedded IBM Proprinter, Epson, or Okidata control sequences.
- FBA files.
- PDF files.
- Text files with PWML markup tags.
- HTML files.
- Image files.

- Images directly from a scanner.
- EMF or WMF files.
- SPL files captured from the Windows spooler.
- Tab-separated values files.
- “Bang command” files.
- “Foreign” Windows file types.

#### Print Wizard can output:

- Plain text reports that always fit the page.
- With specified orientation, paper size, bin, and duplexing.
- Bolding, italics, and underlining.
- Font changes.
- Multilingual characters (Unicode).
- Rectangles, gray bars, and color bars.
- Lines.
- Color text, including white-on-black for security.
- Images (inline or precisely placed).
- Image form overlays.
- Various kinds of form overlays.
- Watermarks.
- Barcodes.
- With precise placement of print elements.
- Multi-column labels.
- One-at-a-time labels on continuous forms.
- Variable-length output, such as with receipt printers.
- Full-page photos.
- Through fax software to a remote fax machine.
- Through email, as plain text.
- Directly to a PDF file.
- Directly to a PDF file and email it.
- Directly to a TIF file.
- Directly to a TIF file and email it.

#### Print Wizard solves a wide variety of printing problems:

- Data processing reports that won't print correctly on a laser printer.
- Line wrap and page break problems.
- Multilingual printing requirements.
- Driving fancier printout from legacy programming environments.
- Printing from scripting languages.
- Printing from DOS programs in a Windows environment.
- Printing to “Windows only” printers from legacy environments.
- Adding form overlays to existing print environments.
- Programmatic faxing.
- Programmatic emailing.
- Printing “print-to-file” files from PC disk.
- Printing to serial, parallel, USB, and network-connected printers.
- Printing HTML pages that browsers don't print well.
- Photo printing that is complicated and has too many steps.
- The need to preview a print job and adjust output placement before printing.
- The need to print a PCL file on a non-PCL printer.
- Replacing multipart preprinted forms.

- Signing documents on-screen.

Where most Windows programs let you *interactively* design and print your output, Print Wizard lets you *program* your output. But there are also interactive options.

*Print Wizard Personal Edition* provides a set of features for taking output from a local file, application or script.

*Print Wizard Service Edition* provides a set of service options for receiving files from remote sources, including despooling print output from a local shared or network shared directory, listening for print jobs on a TCP/IP port (similar to Hewlett-Packard's JetDirect protocol), receiving print jobs over the LPR/LPD protocol, or taking output from an application or script.

*AnzioWin* is a very popular communications client supporting SSH (Secure Shell), telnet and modem communications. It includes Print Wizard Embedded for processing passthrough or remote printing from the host application.

*Print Wizard DLL* is Windows Dynamic Linked Library which can be called from any Windows application to push print jobs out to a Windows target device using Print Wizard Embedded.

*WēPO* is an ActiveX version of Print Wizard. The *WēPO*, Web Print Object, can act as a web page component to “push” a print job from any number of sources out to a local PC's printer. This can also be utilized through some third-party application to accomplish “push” printing.

Most of the rest of this first section of the manual deals with Print Wizard Personal Edition and Print Wizard Service Edition. If you are using *WēPO*, the Print Wizard DLL, or *AnzioWin*, see the *Print Wizard Technical Documentation* section for more information.

### **1.3 Profiles and Services**

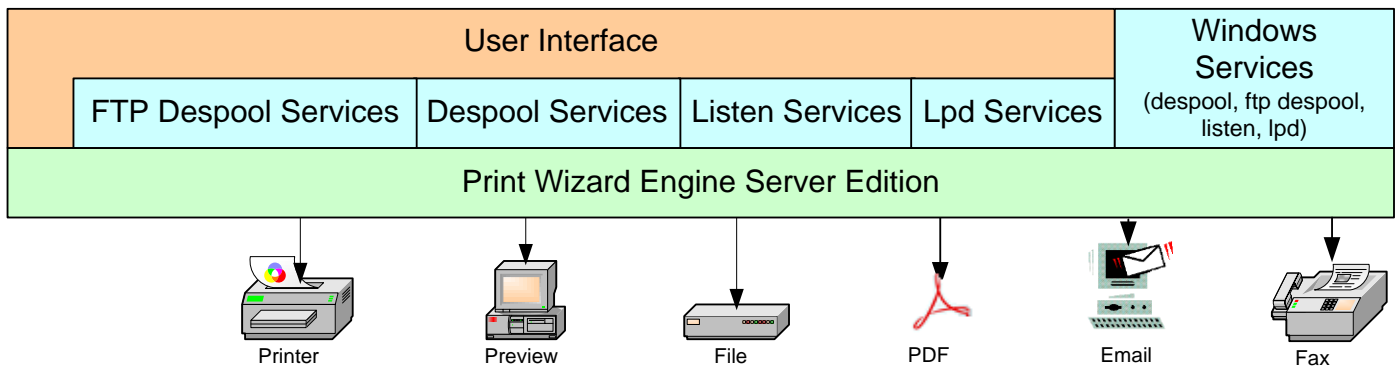
Print Wizard version 3.0 introduced two concepts that need to be discussed briefly before continuing. These are “profiles” and “services”.

A *profile* is a collection of printing (or similar) parameters. It determines the type of output, whether overlays should be added, orientation, character sets, etc. You will typically develop a profile for each kind of print job.

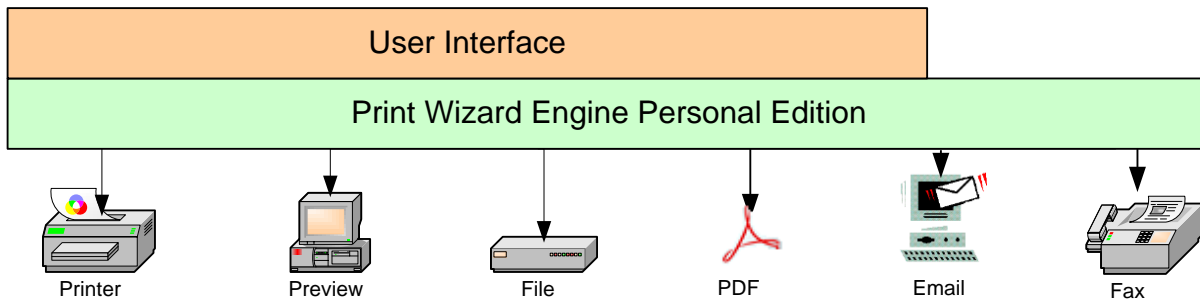
A *service* defines how Print Wizard will *receive* a job. A service runs in the background, either on the desktop but minimized to the tray notification area, or as a Windows service completely in the background.

Services are what distinguish Print Wizard Personal Edition from Print Wizard Service Edition.

A service may be tied to one or more profiles, but not necessarily.



### *Service Edition*



### *Personal Edition*

#### 1.3.1 Introduction to print profiles

The idea of “print profiles” is really not something new. In Windows you can set up your printer driver’s global defaults to print with certain settings. Likewise, some programs retain settings you have made in the Printer Setup dialog box from the last time you printed from that program.

However, there is often more to printing than just what is in the printer driver. Through Print Wizard “print profiles”, you can save off all the Print Wizard settings, including the printer settings, and apply those the next time you want to any specific job.

There is also the concept of a “master profile” whereby you can override your Windows defaults for every job run through the Print Wizard engine, without needing to change your printer driver global settings (which could influence the behavior of printing from other programs).

Here is a quick synopsis of how the Print Wizard engine looks at profiles:

1. Load factory default (some set during compile, some pulled from Windows)

2. If a “master.profile” file exists, load the master profile over the top of the default.
3. If the user specifies a “print profile”, load this over the top of the current settings.
4. Process the job...

Profiles are managed through the Print Wizard User Interface. They can be applied to individual print jobs started by the user through the user interface, they can be included when starting the Print Wizard engine from a third-party program or they can be established for a service to use. See the technical documentation below on *Inside Profiles* for additional information.

### 1.3.2 Introduction to services

Print Wizard version 3.0 also introduced the concept of “services”. There are two types of services supported by Print Wizard, “Foreground Services” (formerly called “Print Wizard Services”) and “Windows Services”.

“Foreground Services” are those services manually started by a user, independent of the Windows startup procedure. This can be from a script or a third-party program, or through a desktop shortcut, but they run under the user’s Windows login, dependent on the user’s permissions and on the user being logged in to run correctly.

“Windows Services” install in Windows just like any other background process. They run under a separate user (normally LOCALSYSTEM) and are dependent on the permissions of that user, not the logged-in user. They auto-start when Windows is brought up and do not depend on a user logging in to the Windows box.

Services are set up through the Print Wizard User Interface. They are also managed from the user interface, provided the logged-in user has sufficient permissions. See the technical documentation below on *Inside Services* for more information.

## 2 The User Interface

### 2.1 Beginning to use Print Wizard

When you install Print Wizard, you will see just one Print Wizard shortcut on your desktop. This program is the Print Wizard User Interface program (PWUI). PWUI is used for two primary purposes: 1) configuring Print Wizard operations, and 2) manually running Print Wizard to print, fax, or make a PDF from a file.

Once installed, you can start using Print Wizard by simply dragging and dropping a file to print onto the desktop shortcut for PWUI. With everything set for defaults, the file should print on your default printer with no other options necessary.

However, you will probably want to set up the various options in Print Wizard to handle your specific needs. To do this you will want to set up faxing, emailing, “print profiles” and “print services” to match your specific requirements. All of this is done through the Print Wizard User Interface.

#### 2.1.1 First-time use

The very first time you use the Print Wizard user interface, Print Wizard will walk you through the Setup Wizard. You can always rerun the Setup Wizard at a later time as well, from the Tools menu. The Setup Wizard will walk you through some initial steps in preparing Print Wizard for use. Most of these steps are simply to make sure you understand what is needed for Print Wizard to work, and you can cancel out of this routine at any time without consequences.

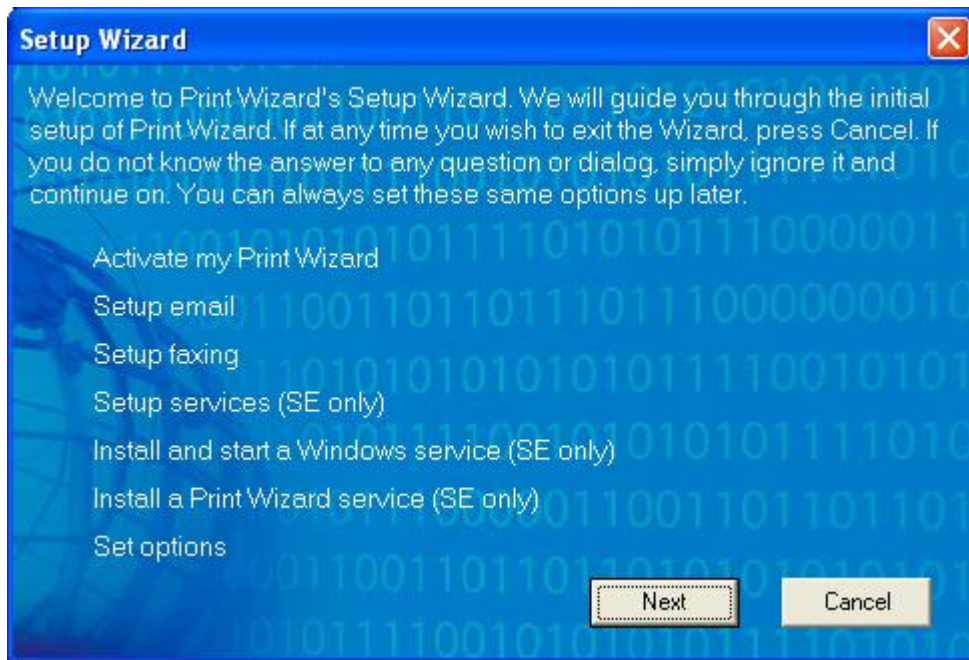
In order for Print Wizard to do its job, you must:

- Activate Print Wizard with a serial number and activation code.
- Provide email settings (if you are going to use Print Wizard for automatically sending emails)
- Provide faxing information (if you are going to use Print Wizard for faxing)
- Set up various services, and start them.

You can ignore dialogs for features you will not be using. The Wizard is designed to traverse several dialogs already available through the user interface. If a particular dialog is not applicable, you can simply hit Enter, or press the Done or Cancel button, to continue.

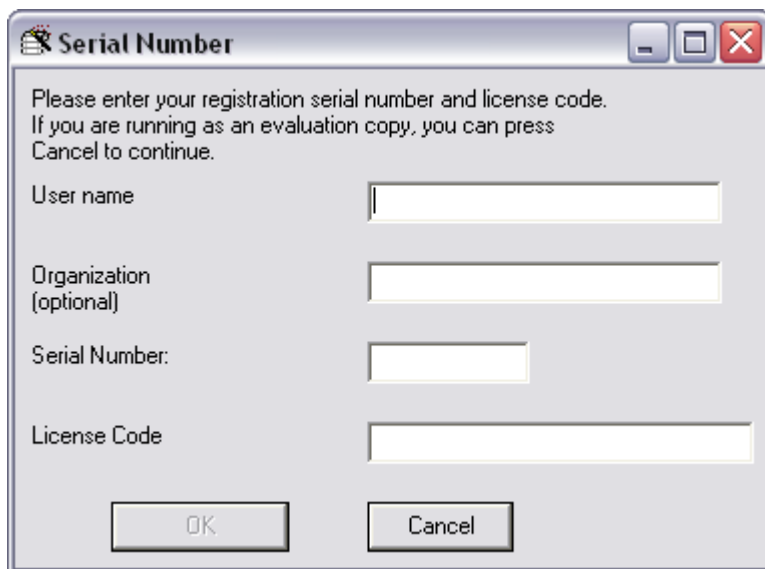
At any time, you can press Cancel from the main Wizard dialog to quit out of the Setup Wizard.

Most of these dialogs are also covered below in the user interface as well. But they are worth repeating here. Again, if a feature is not going to be used, you can simply step over that dialog.

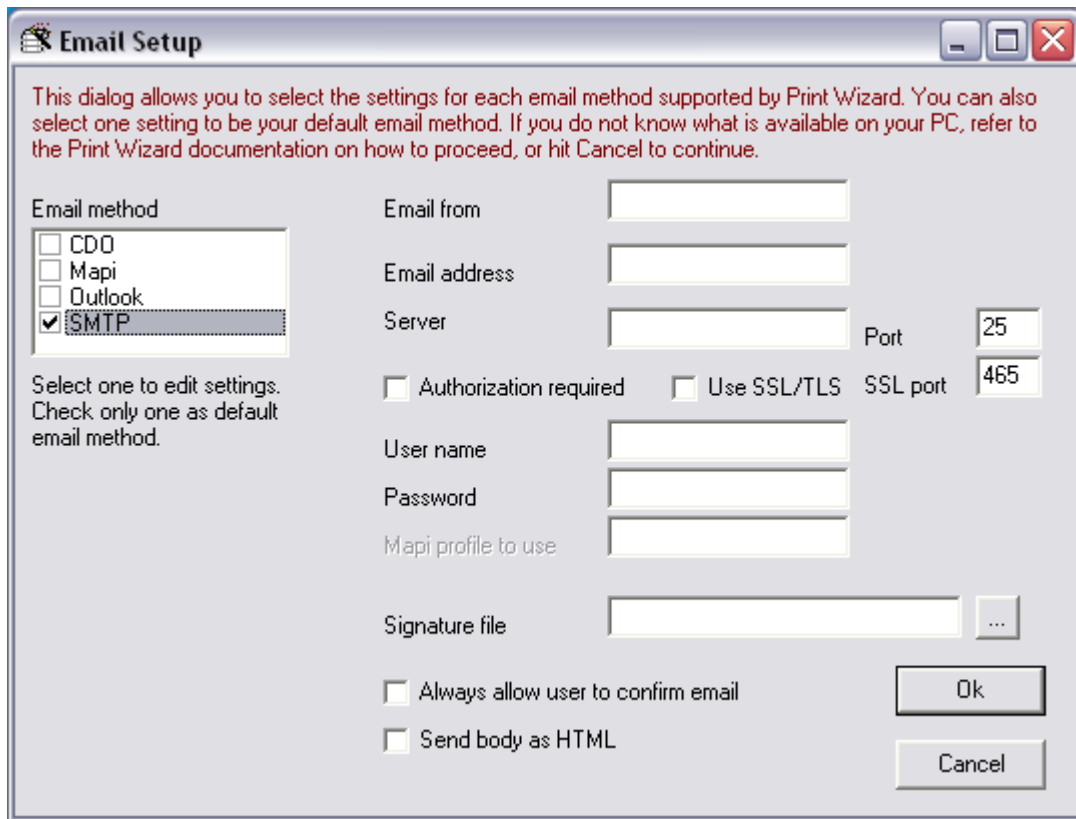


Click "Next" to begin the Wizard. Click "Cancel" to stop the Wizard and return to the main program.

You will first be asked to enter your license information (serial number and activation code) that was provided with your registered copy of the software. If you are running a demo, simply hit cancel or press the Enter key. Otherwise enter the provided serial number and activation information ("user name" and "organization" are free form fields).



Next you will be asked for information on what email method you wish to use. If you are going to use Print Wizard to email any print jobs, you need to set up at least one method as a default email.



You can select from the various supported email methods and set one of these as a default by checking the box. By clicking on the method name (not the check box), you can edit details for that method.

To do this set up, you need to know a little about what email methods are supported by your PC.

## MAPI

MAPI is Microsoft's "Messaging Application Program Interface". It is the key method of doing email for Microsoft Office, Outlook and Outlook Express. It is also included with various versions of Microsoft Exchange but may not be loaded by default. There are various levels of MAPI available, from "simple MAPI" that ships with Outlook Express, to standard MAPI with older version of Office and Outlook, to Extended MAPI shipped with the most recent version of Outlook. In order for Print Wizard to email through MAPI, a MAPI profile (not to be confused with Print Wizard Print Profiles) may need to be set up in Outlook (it will be required if you have extended MAPI installed).

## CDO

CDO stands for Collaboration Data Objects and simplifies the use of MAPI to do emails. MAPI does need to be installed for this to work. MAPI ships with Microsoft Office, Outlook and a limited version with Outlook Express.

## OUTLOOK

The Outlook method wraps around extended MAPI to handle emailing directly through Outlook calls.

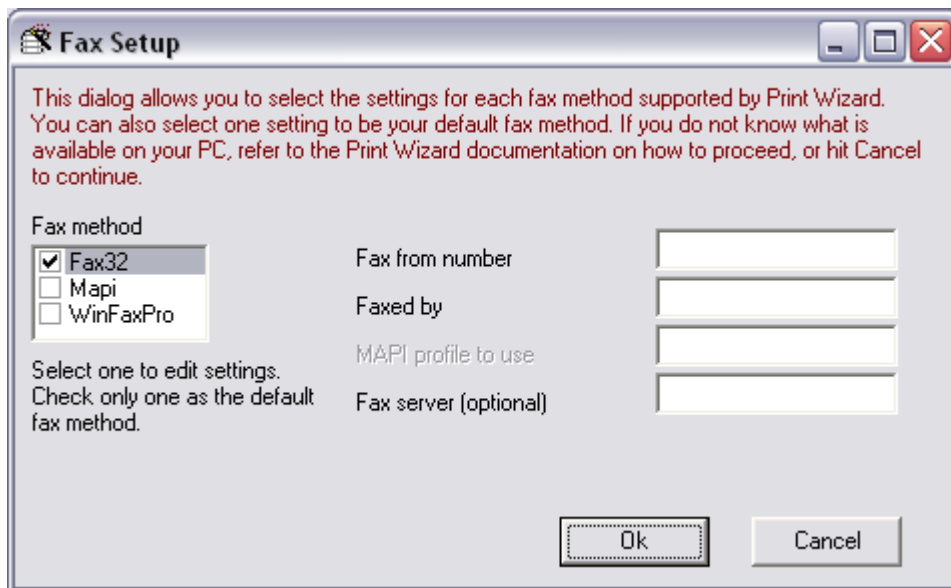


## SMTP

Simple Mail Transport Protocol is the common way non-Microsoft email applications work, including those found on UNIX and Linux. This method requires direct access to a mail relay server in order to work properly.

Additional information on the email methods can be found in the Appendix.

The next dialog allows you to set up faxing. If you are going to use Print Wizard to fax, you will need to set up at least one method as a default fax method.



Print Wizard supports 3 different faxing methods. Print Wizard can make calls into these methods.

## FAX32

Fax32 is the Microsoft Windows standard fax printer driver available with Windows XP or later.

## MAPI

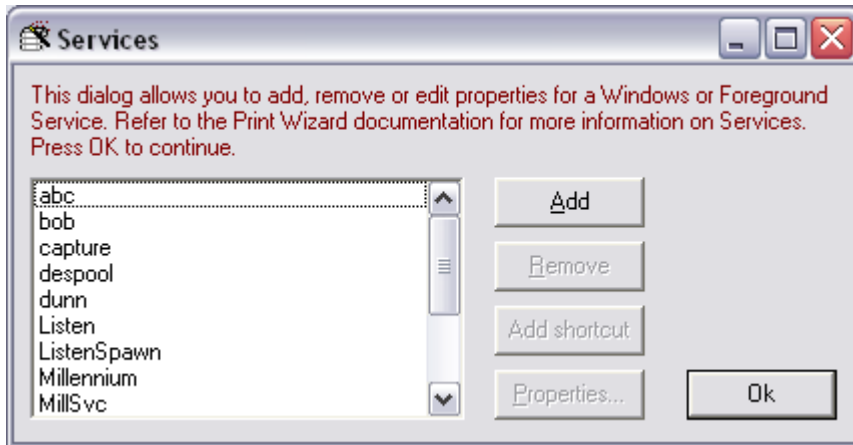
MAPI, as discussed above, is Microsoft's messaging API. Within MAPI, you can set up a MAPI profile that has faxing as its main "transport" option. You can then fax using MAPI and that profile.

## WINFAX PRO

Starting with Symantec's WinFax PRO version 10.0, Print Wizard can make calls direct to the WinFax PRO software for faxing. Note: this requires version 10.0 or 10.0.3 or later. Versions 10.0.1 and 10.0.2 have problems with third-party software products.

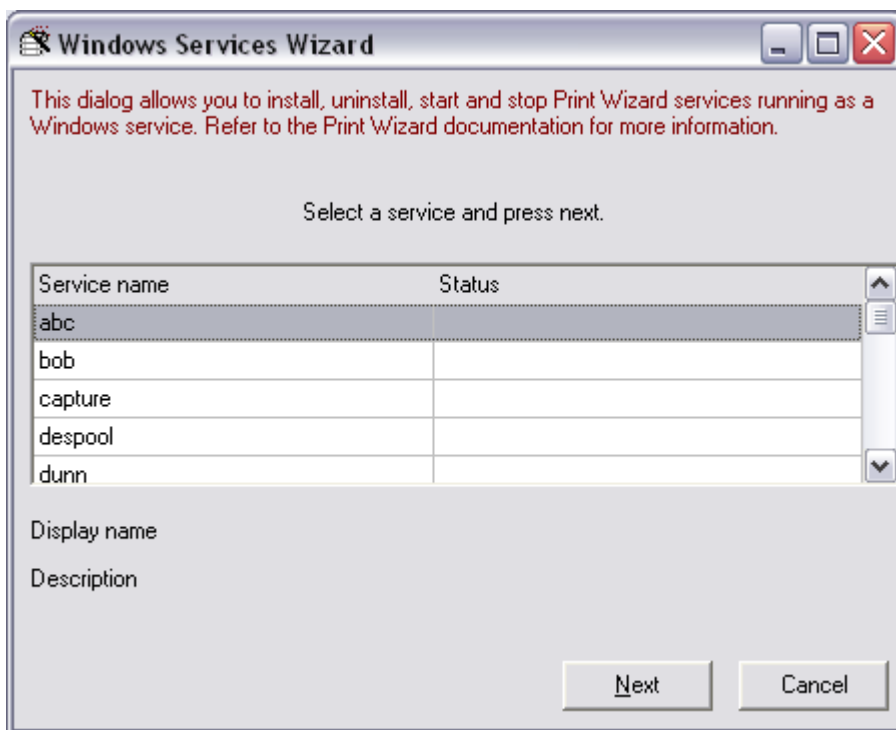
Additional information on the fax methods can be found in the Appendix.

The next dialog asks you if you want to set up any services. These can be services you intend to run as a Windows Service or as a Foreground Service, and include three basic types, Despool, Listen, and LPD.



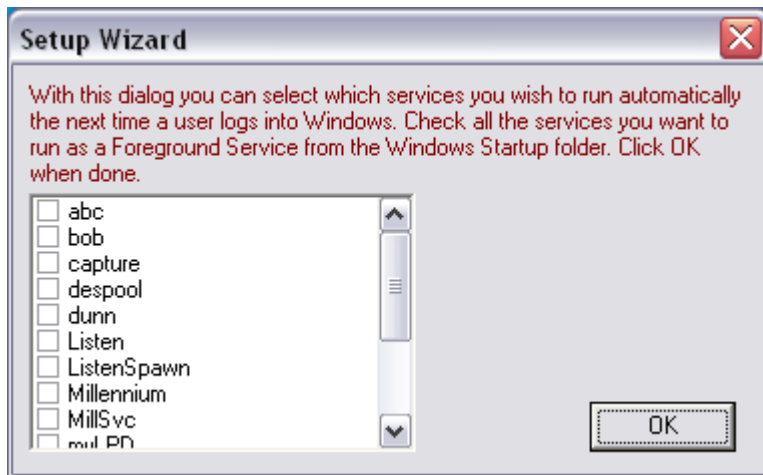
Refer to information below and in the Appendix on setting up and using Foreground Services.

The next dialog will allow you to install, uninstall, start, and stop Windows Services related to Print Wizard.



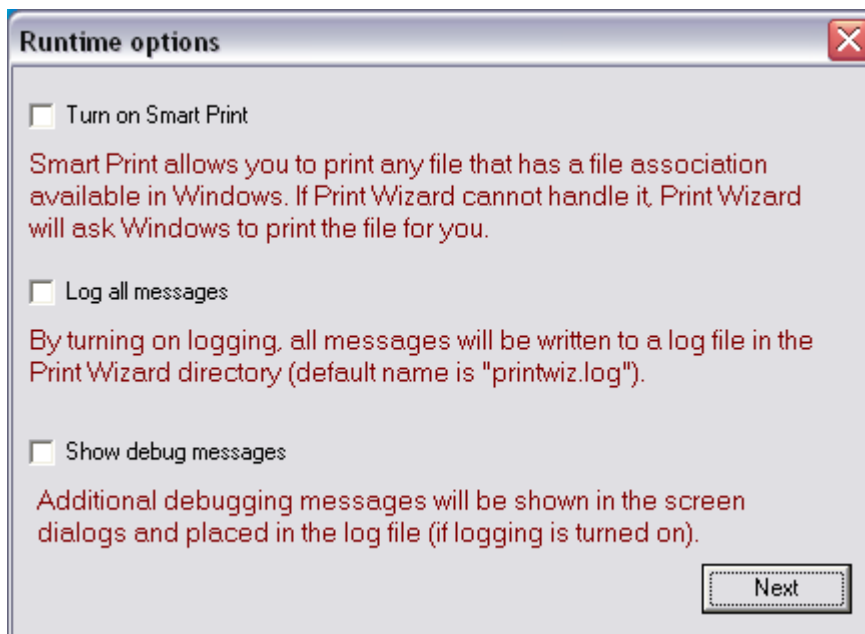
Refer to information below and in the Appendix on services for more information.

Next you will be given an opportunity to set a specific service up to run as a Foreground Service from the Windows Startup folder. This means that each time the Windows user logs in, the service will automatically start up for that user.



Refer to sections below on Foreground Services for more information.

The next dialog allows you to turn on or off certain messaging and debugging information that will be generated by each print job run through Print Wizard.



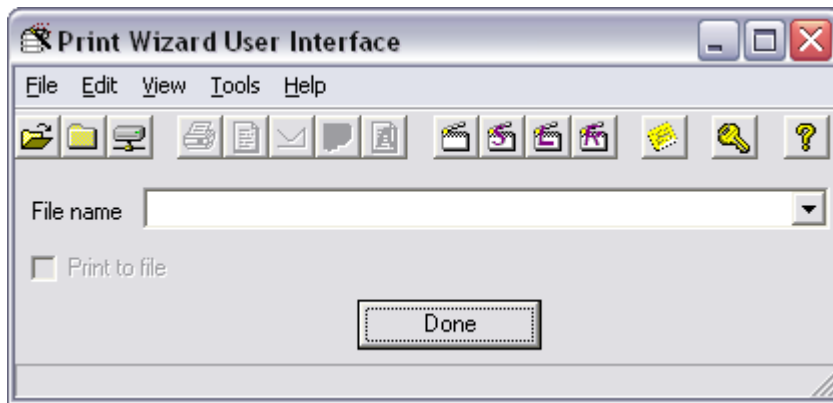
Once complete, simply press Next to start the user interface running.

If you wish to restart Print Wizard from its factory defaults, and force it to run through this Wizard on startup again, remove the file “printwiz.ini” from the Print Wizard directory.

## 2.2 Print Wizard User Interface

The Print Wizard User Interface (PWUI) provides you with a method by which you can open a file, apply settings to the printing of the file, and then print or preview the file using the Print Wizard engine. The following steps will walk you through the common procedures you would use.

### 2.2.1 The main dialog



The main dialog is made up of the pop-down menus, the toolbar, the file name and file name history, the “Done” button and the status bar.

### 2.2.2 The status bar

This shows you the current state of Print Wizard and the number of days remaining on your demo (if you are running a demo version). It also indicates when a file is being processed.

### 2.2.3 Done

The “Done” button, exits you from the program.

### 2.2.4 File name and history

The file name is that which the action will be performed on. Also note that this is a drop down list and will show you the last several print jobs you ran through the user interface.

### 2.2.5 The tool bar

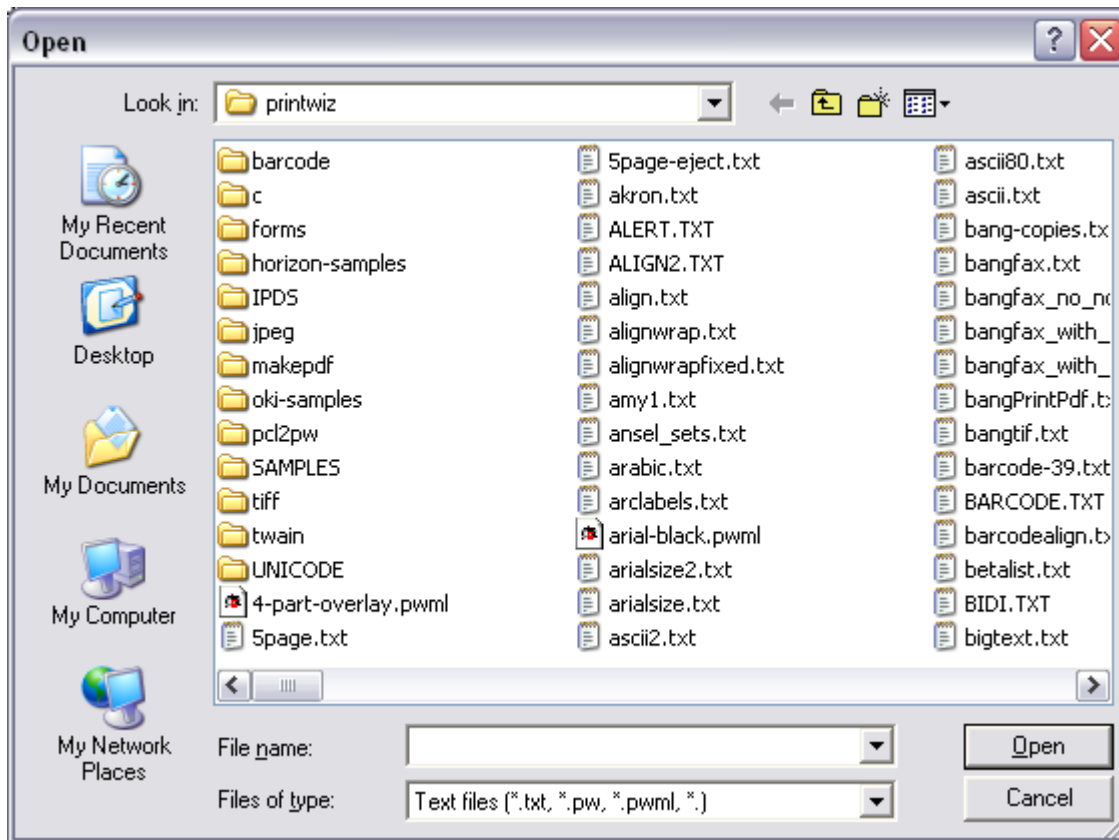
Items on the tool bar directly relate to the menu items. This simply provides a more direct approach to accessing menu items (see below for more information).

### 2.2.6 The Menus

These are standard Windows drop-down menus, providing access to various aspects of printing with Print Wizard. Each menu item will be covered in detail here.

Certain menu items may be grayed out. This indicates that either the action is not currently available or there is no file to perform specific actions on (see “File : Open”)

## 2.2.7 File : Open



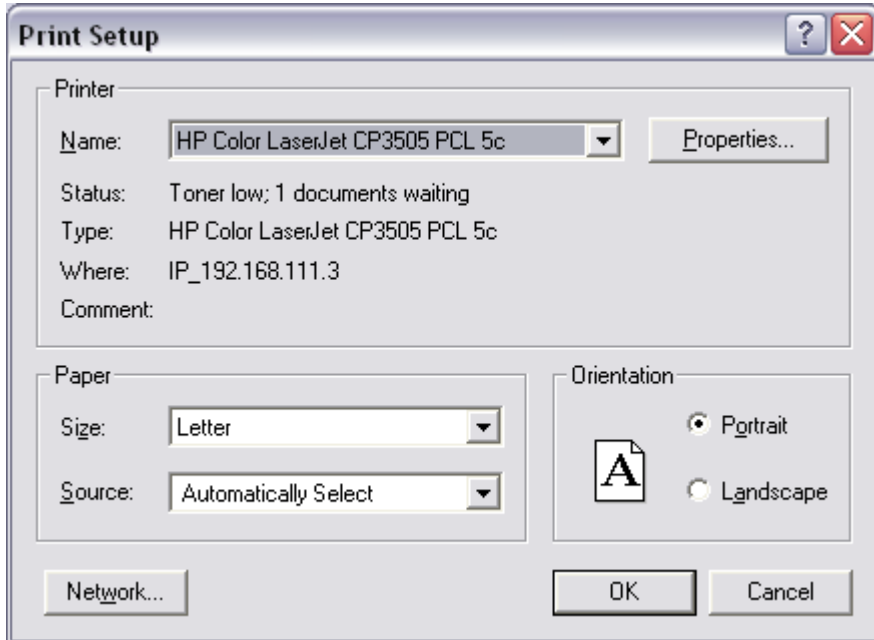
This retrieves the file name for the current settings to apply to.

The **Open** does not actually open the file, but rather verifies that it is available for the print process to run against. The file will be opened when the required action is started (print, preview, fax, email, etc.). However, you must select a file or type one in manually (in the File Name field) before certain menu items will be available.

## 2.2.8 File : Close

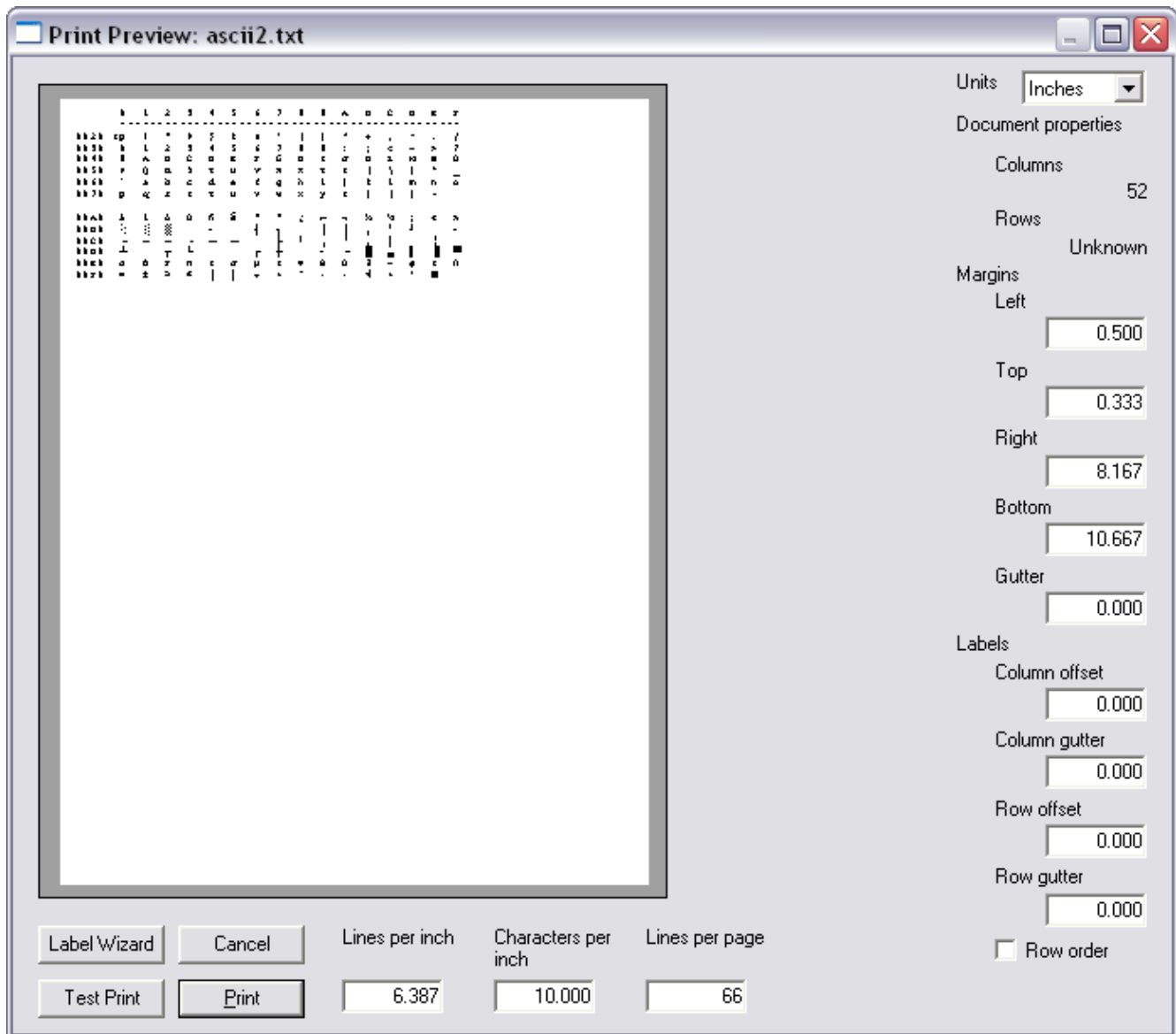
This clears out the File Name field.

## 2.2.9 File : Print



When a valid file name is given, this item will fire off the Print Wizard engine to print the selected file or files. It first takes you to the “Print Setup” dialog, allowing you to change various options.

## 2.2.10 File : Preview

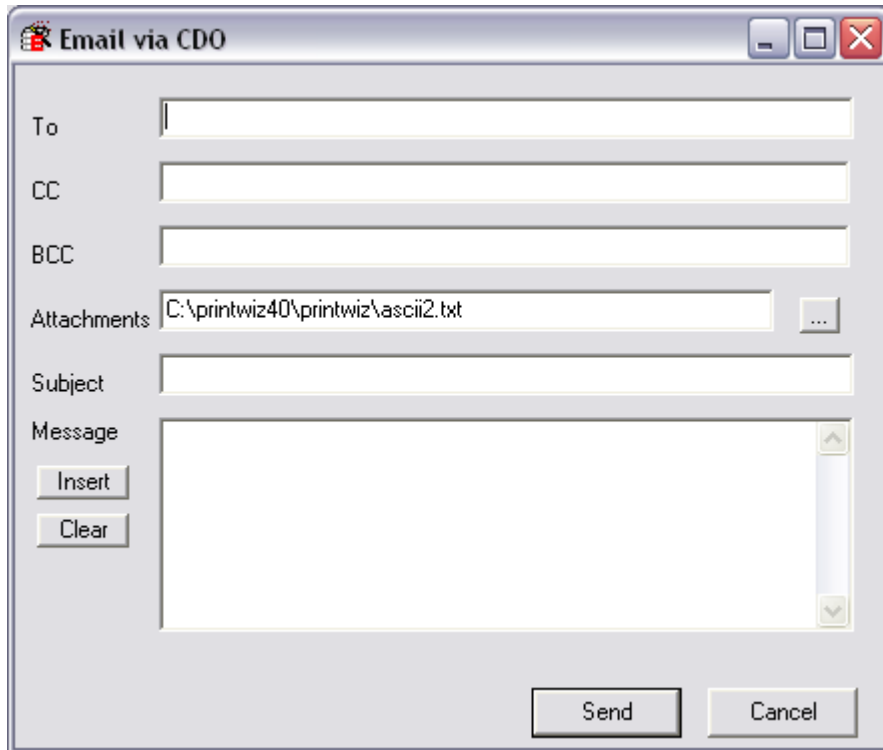


This menu item will display the Print Wizard Preview dialog. You can drag and expand this dialog to show you the print and adjust certain layout items before actually printing the page.

Many of the dialogs throughout Print Wizard and its support programs allow you to resize to see more content. You can “grab” any or edge of the dialogs by clicking with the left mouse button and drag it to resize. You can also “zoom” the dialog. This is especially useful with the Print Preview dialog.



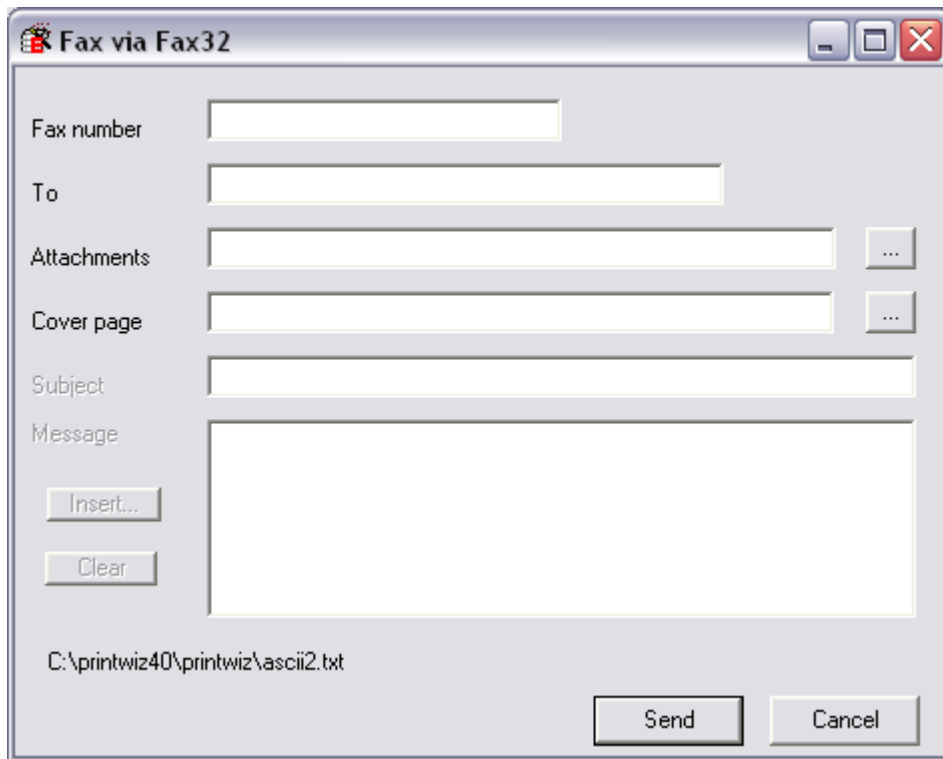
## 2.2.11 File : Email



When you want to email a selected print file, use this dialog. It is set to use the current email default transport method (see the Tools menu). Print files will be emailed as attachments.

From this dialog, you can also select additional attachments, include a “body” text file, or type in your own body text. Pressing “Send” will send the selected print file and all attachments to the recipient’s address.

## 2.2.12 File : Fax



When you want to fax a selected print file, use this dialog. It is set to use the current fax default transport method (see the Tools menu). The handling of print files and attachments, and which fields are enabled **may** depend on your selected fax program and how it handles faxing.

Pressing “Send” will send the selected print file and attachments, if applicable, to the fax number entered.

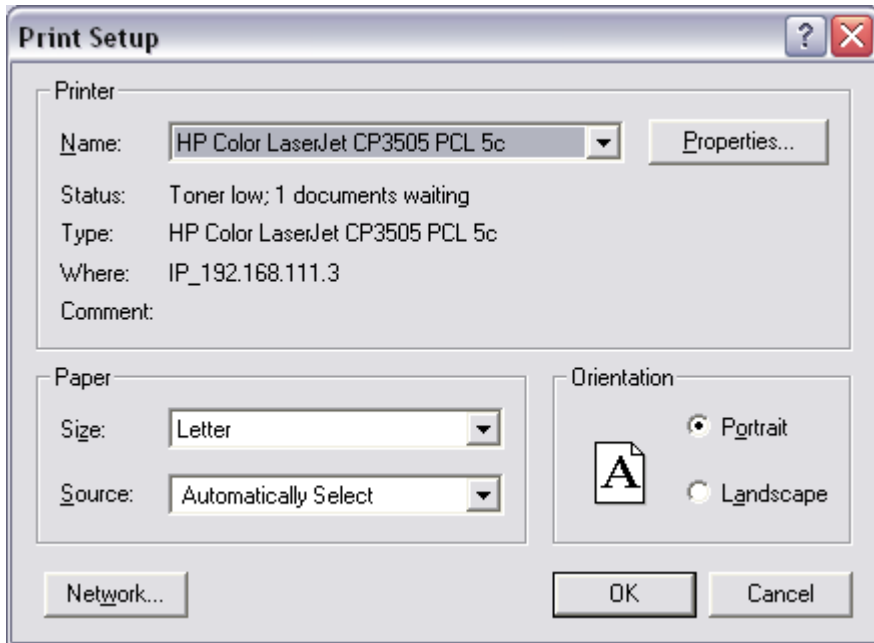
## 2.2.13 File : Generate PDF

When you select this item, the given print file will create a PDF file compatible with Adobe Acrobat Reader ®.

## 2.2.14 File : Generate then...

This item determines what you want to do with the generated PDF file once it is created. You can elect to do nothing, (“No action”), to view it in the default PDF viewer, (“View it”), or elect to “Email it”.

## 2.2.15 File : Printer setup



This is the standard Windows printer driver dialog box. This dialog allows you to set up various printer-specific items prior to printing the selected print file. The appearance of this dialog may vary depending on your printer and printer driver.

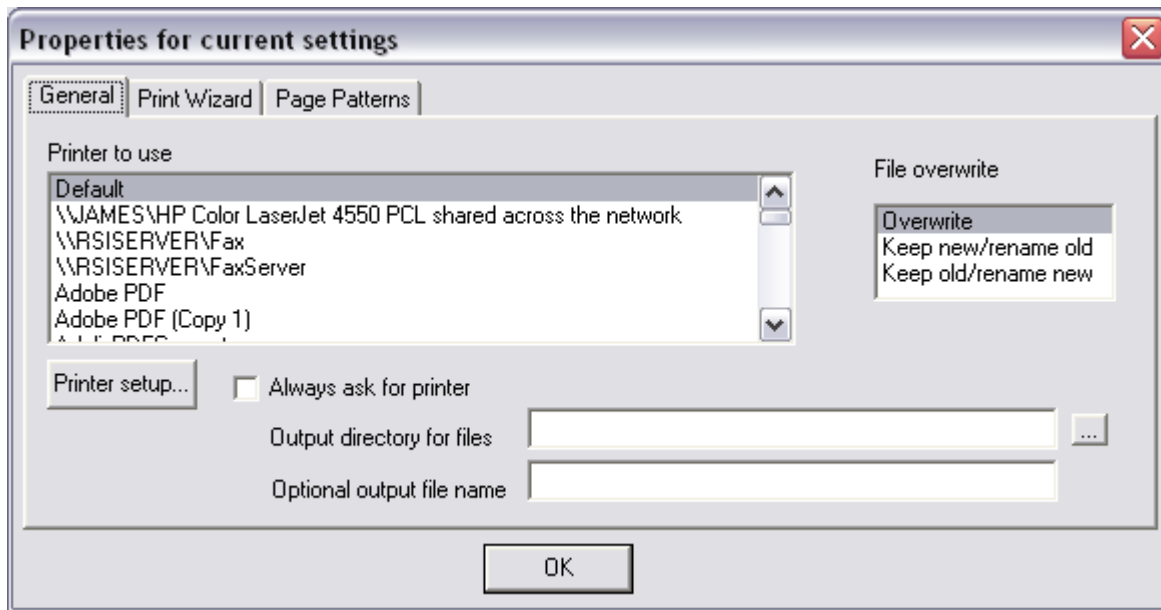
### **2.2.16 File : Exit**

Quit out of the Print Wizard program.

### **2.2.17 Edit : Smart printing**

When checked, Print Wizard can handle most any file type dropped on it, or selected from the Open dialog. This feature pays attention to the file extension and if the file extension is NOT one of the supported Print Wizard file extensions, Print Wizard will attempt to print it or view it through the extensions parent program (the Windows file association or file type).

## 2.2.18 Edit : Edit settings



This dialog shows you the current settings you have for Print Wizard. If you have not loaded any printer profile (discussed later) then this shows the current default settings. You can change these settings and hence the current default behavior of Print Wizard printing from here.

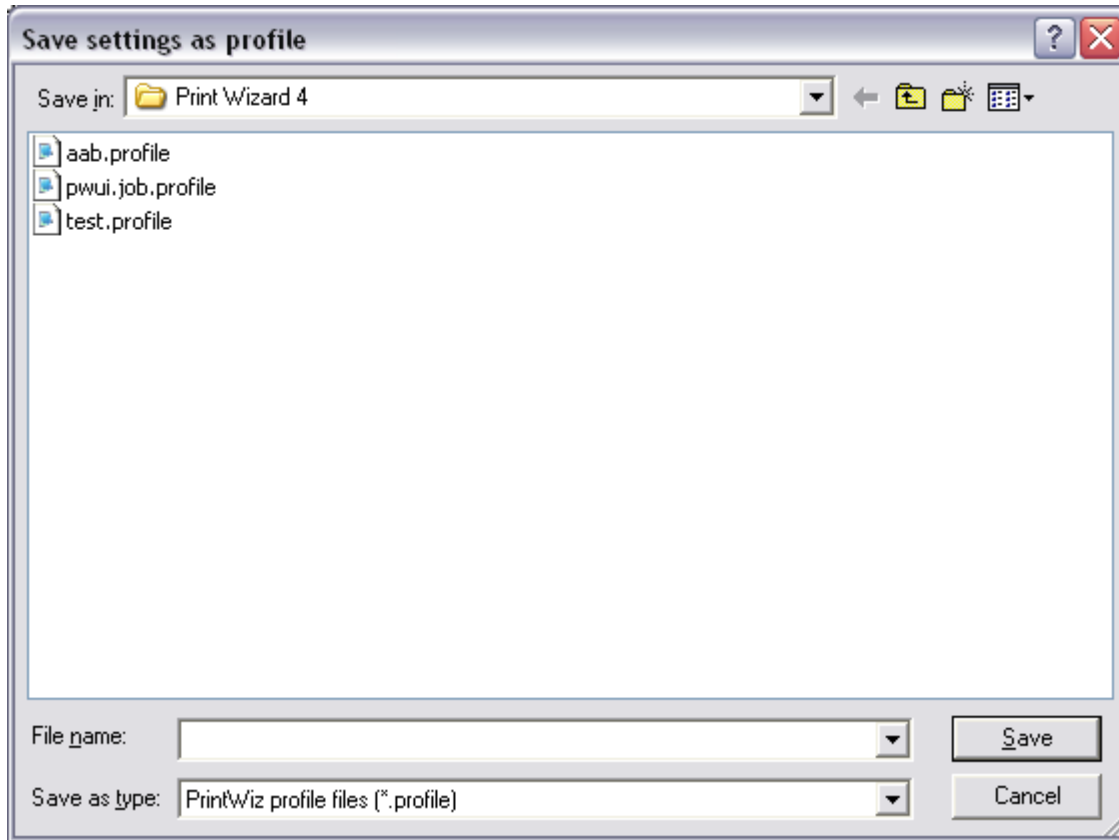
## 2.2.19 Edit : Reset settings

This option will reset the current Print Wizard settings to either the “factory defaults” or to the “master defaults” if a master print profile exists (discussed later).

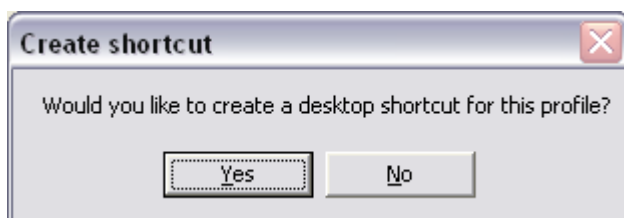
## 2.2.20 Edit : Load profile

You can load a print profile over the top of the current defaults by selecting a pre-defined print profile from this standard open dialog.

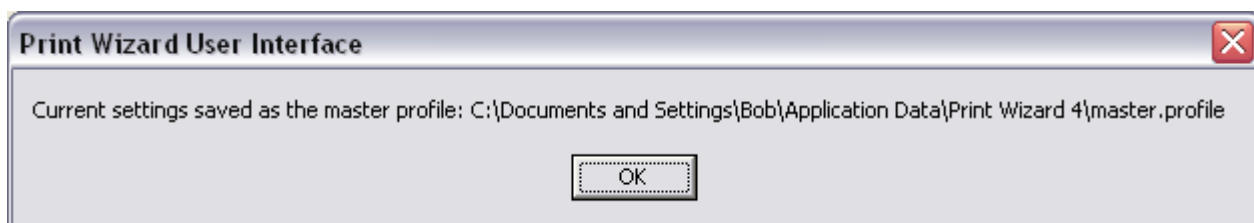
## 2.2.21 Edit : Save settings as user profile



Select this item to save your current settings as a new print profile. When you select “Save”, Print Wizard will ask you if you wish to create a Desktop shortcut for Windows. This shortcut would then allow you to drag and drop files on the shortcut that would print with the new print profile.



## 2.2.22 Edit : Save settings as master profile



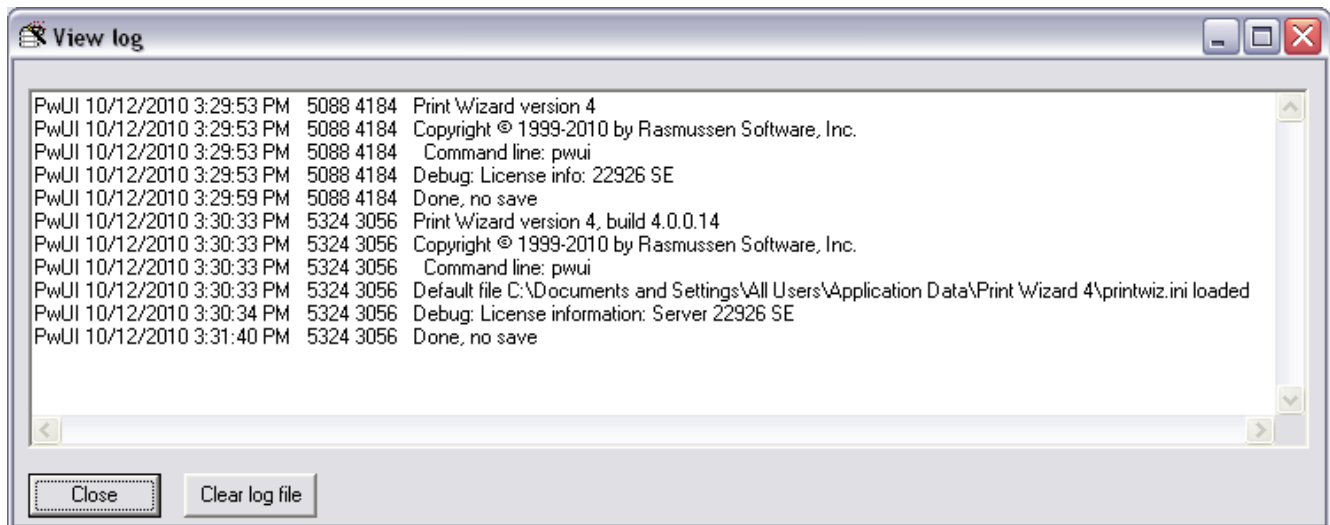
When you select this item, a master profile is selected that contains all the current print settings, making them your defaults from then on. You can remove this master profile with the “Remove master profile” menu item. Otherwise, whenever you print direct, through a drag and drop operation or through a Windows Service or Foreground Service, or whenever you bring up the user interface for Print Wizard, this master profile will be loaded as your defaults.

### 2.2.23 Edit : Remove master profile



By selecting this menu item, you can remove the master profile and reset Print Wizard defaults back to their factory settings.

### 2.2.24 View : View log



If “Log messages” is checked in the View menu, several messages are written for each job run and for each program, into a common log file. Likewise, verbose debug messages are found in the log file if the “Debug” option of the View menu is checked. With this menu item, you can scroll through a view these messages and clear the log file.

The default log file is created in the same directory as the executables and is usually called “printwiz.log”. Each program adding lines to the log file prefaces its messages with an abbreviation and the date and time in order to know the message source.

Likewise, you can set up a separate log file for the services if you wish to keep them separate (see the Appendix on log files).

### **2.2.25 View : Log messages**

When checked, certain messages are written to a log file.

### **2.2.26 View : Debug**

When checked additional debugging messages are displayed where appropriate. If “Log messages” is also checked, additional debugging messages are also written to the log file.

### **2.2.27 View : Start minimized**

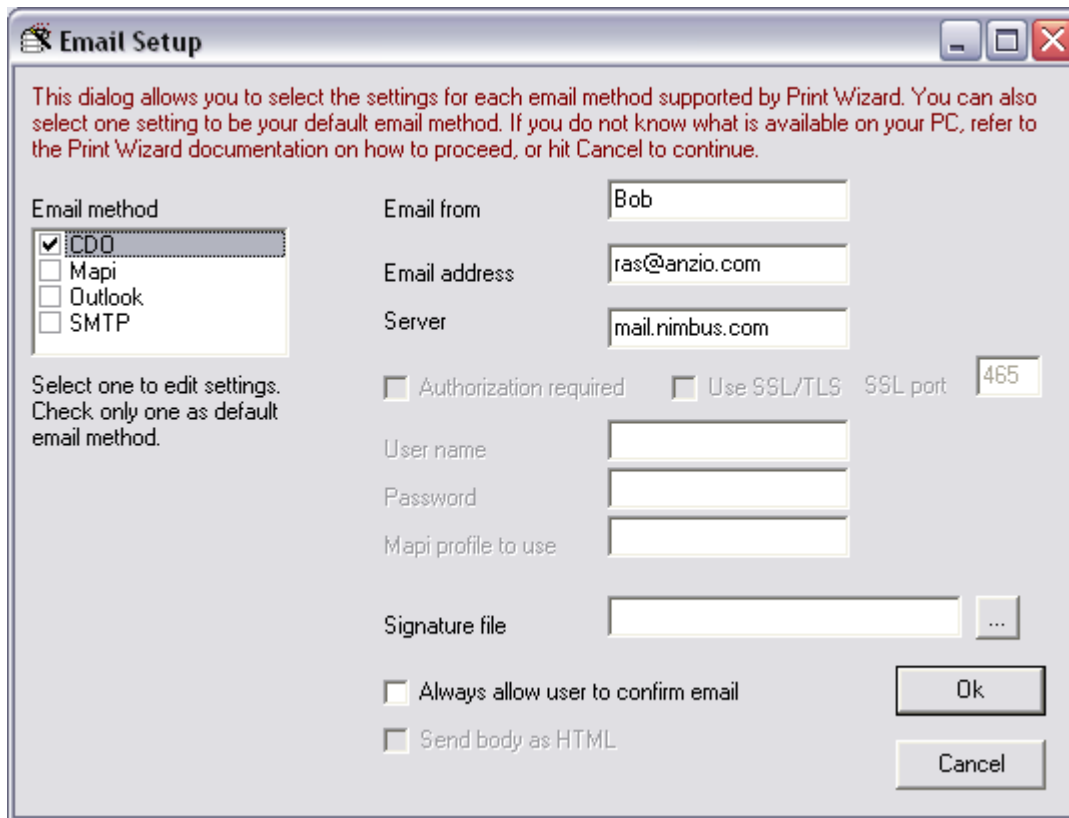
Under certain circumstances you may want to start the user interface minimized. This option allows you to do so.

### **2.2.28 View : Show toolbar**

You can turn off the display of the tool bar icons with this item.

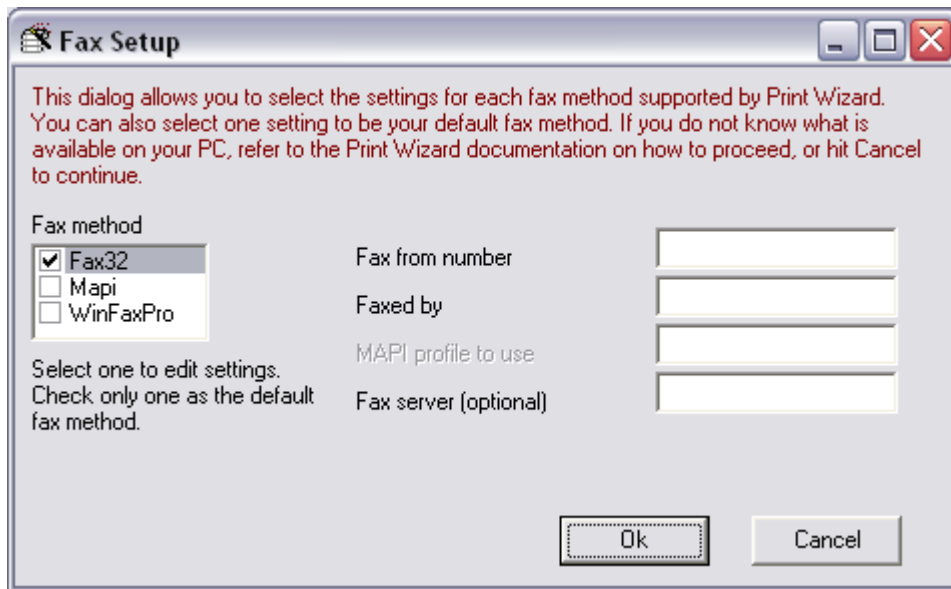


## 2.2.29 Tools : Setup email



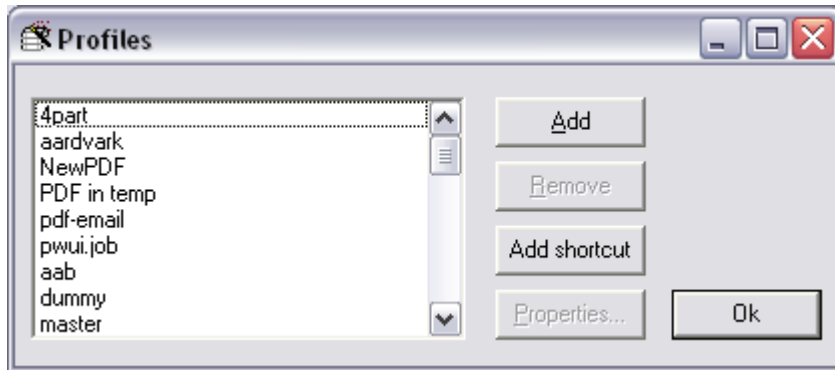
From this dialog you can set up the various emails your PC supports. While you can edit the settings for all email methods, you can only have one item checked as the default email type (see the Appendix on Emailing).

## 2.2.30 Tools : Setup faxing



From this dialog you can set up the various fax support. While you can edit the settings for all fax methods, you can only have one item checked as the default fax type (see the Appendix on faxing).

## 2.2.31 Tools : Setup profiles

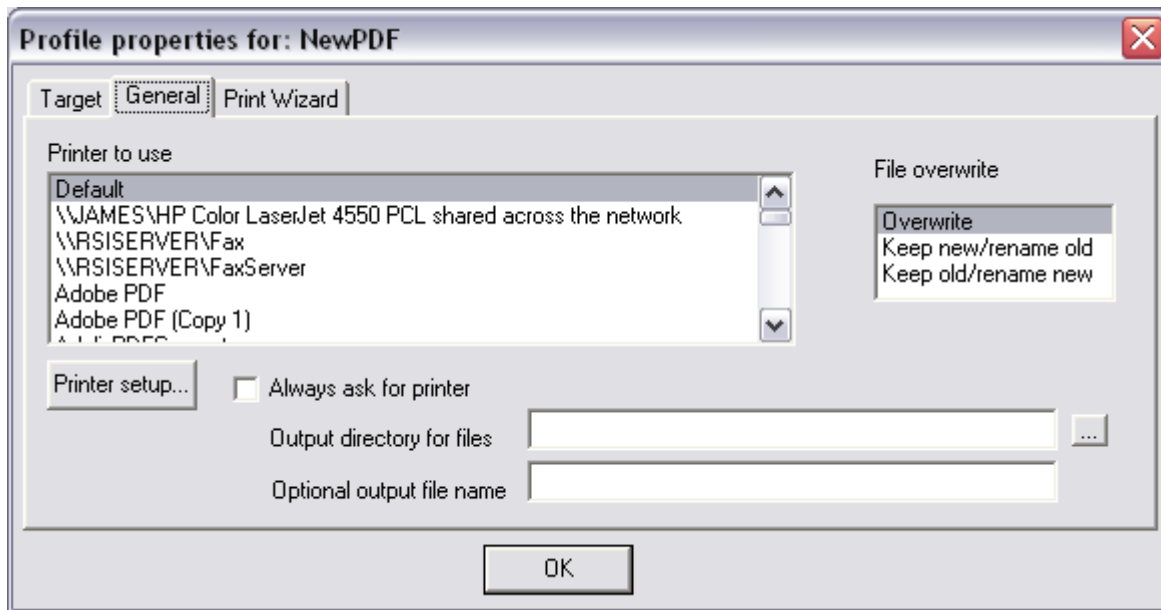


This shows a list of the current print profiles available on your PC. You can select a print profile to edit or remove, or you can add a new one.

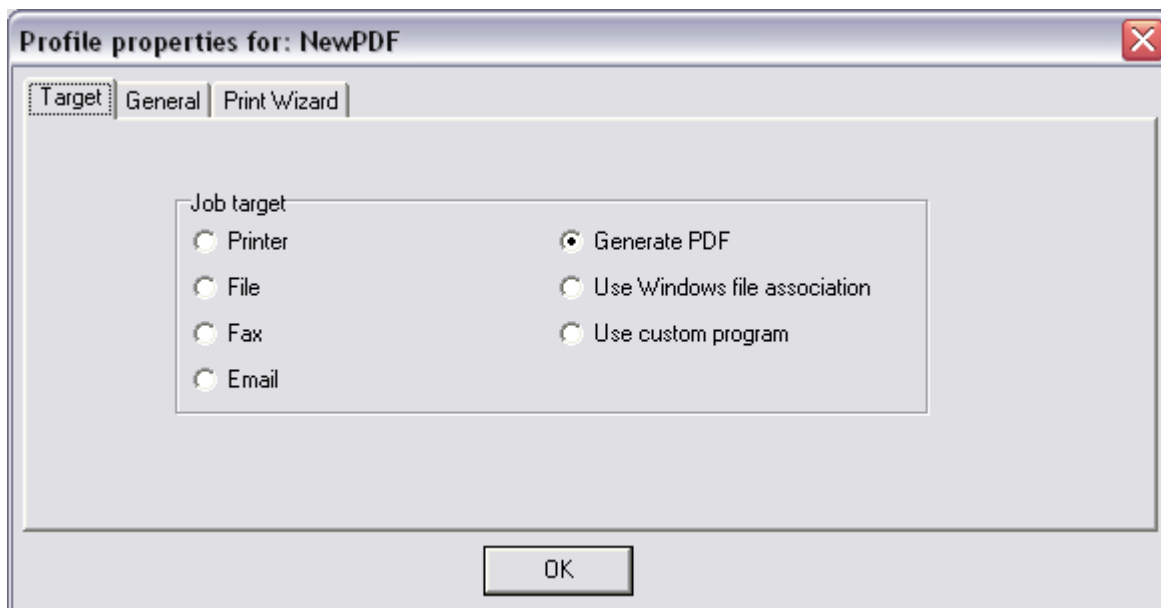
When adding a profile, you will be asked for the profile name.



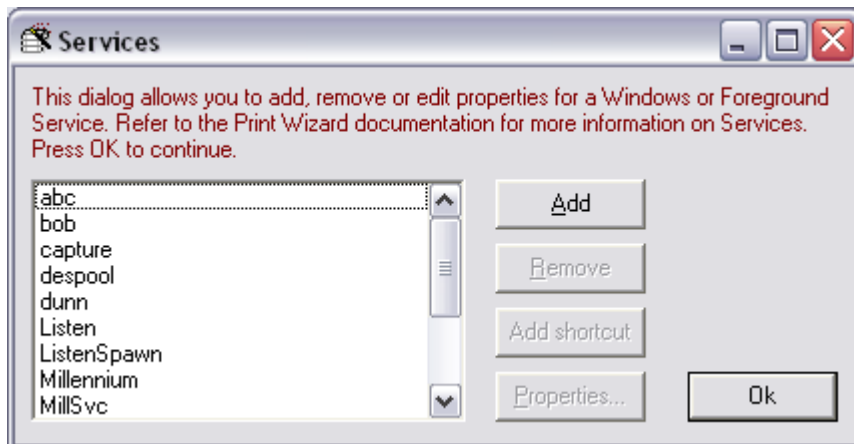
Once you enter a valid name, it will take you to the Edit Profile Properties screen to allow you to set this profile up. If you select Properties instead, it also takes you to this screen for the existing profile you selected.



On the *General* tab, you select the printer and various printer setup settings for your print profile. On the *Target* tab you can select the job target (where is output going). This selection will determine what other properties you may need to set up. There are additional properties for setting a profile for faxing, emailing, using Windows file associations and for using a custom (third-party) program (see the technical documentation on *Inside Profiles* for more information).



## 2.2.32 Tools : Setup services

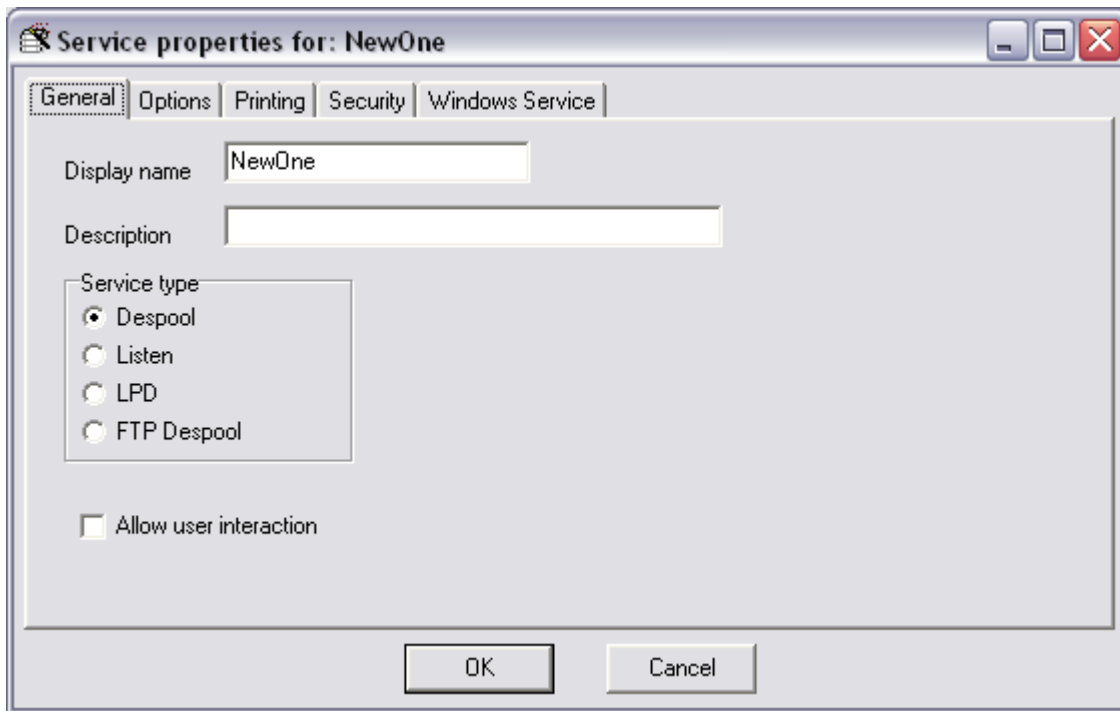


This shows a list of the current services setup for Print Wizard. You can select a service by name to edit or remove, or you can add a new one.

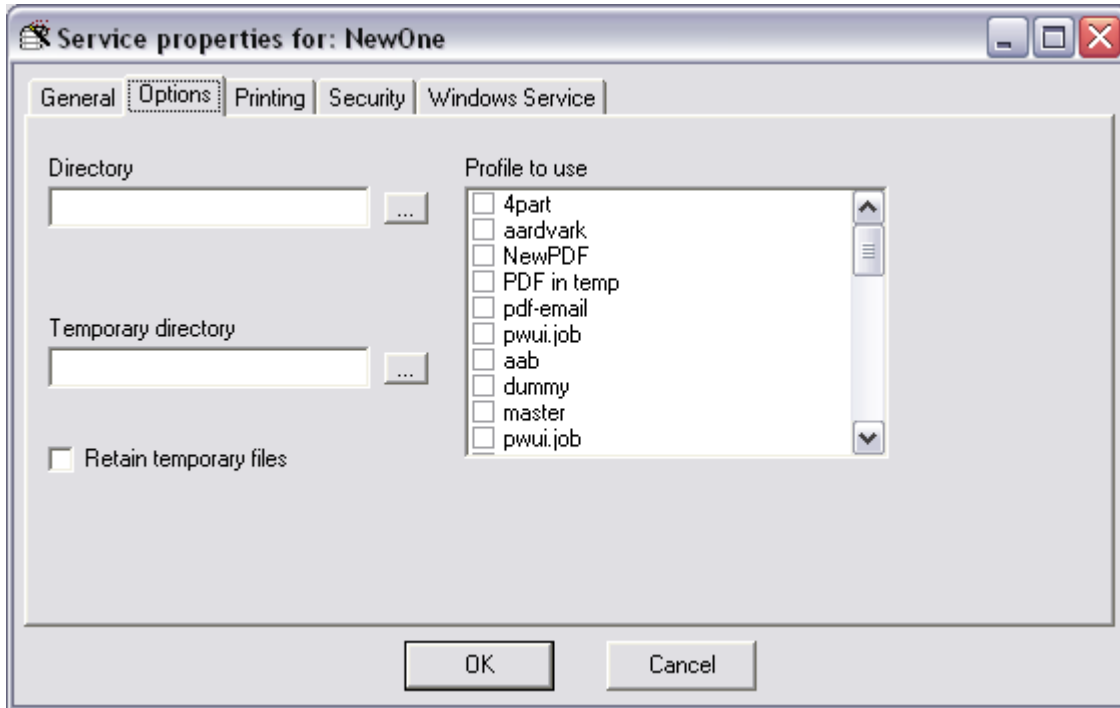
When adding a profile, you will be asked for the service name

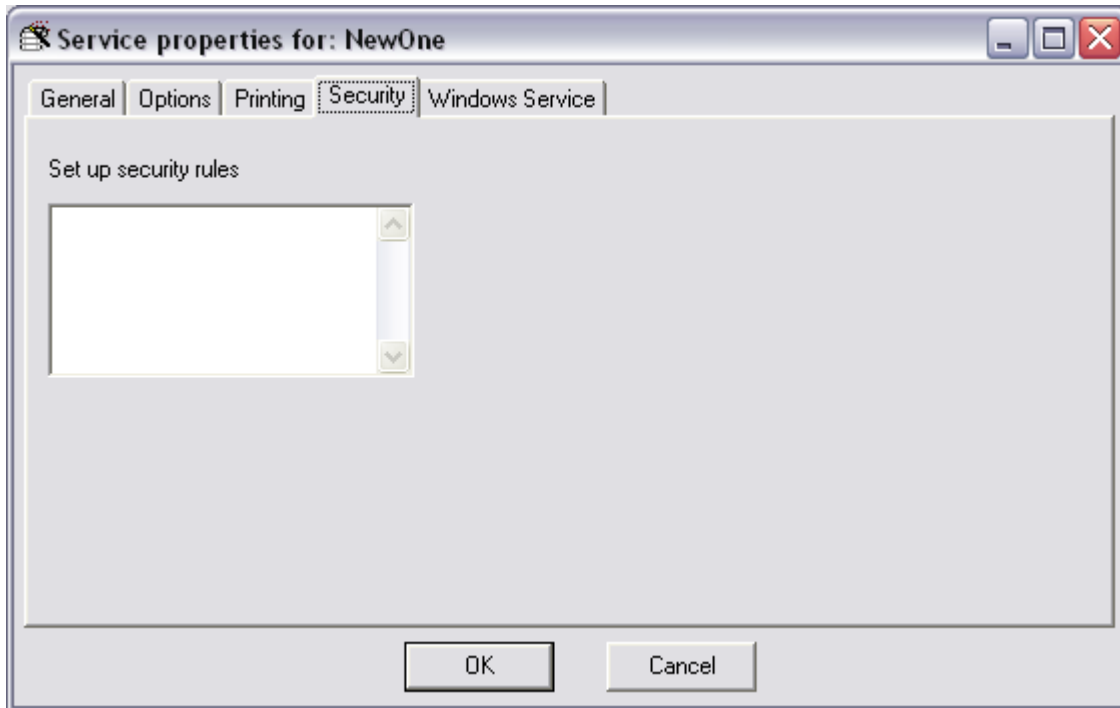
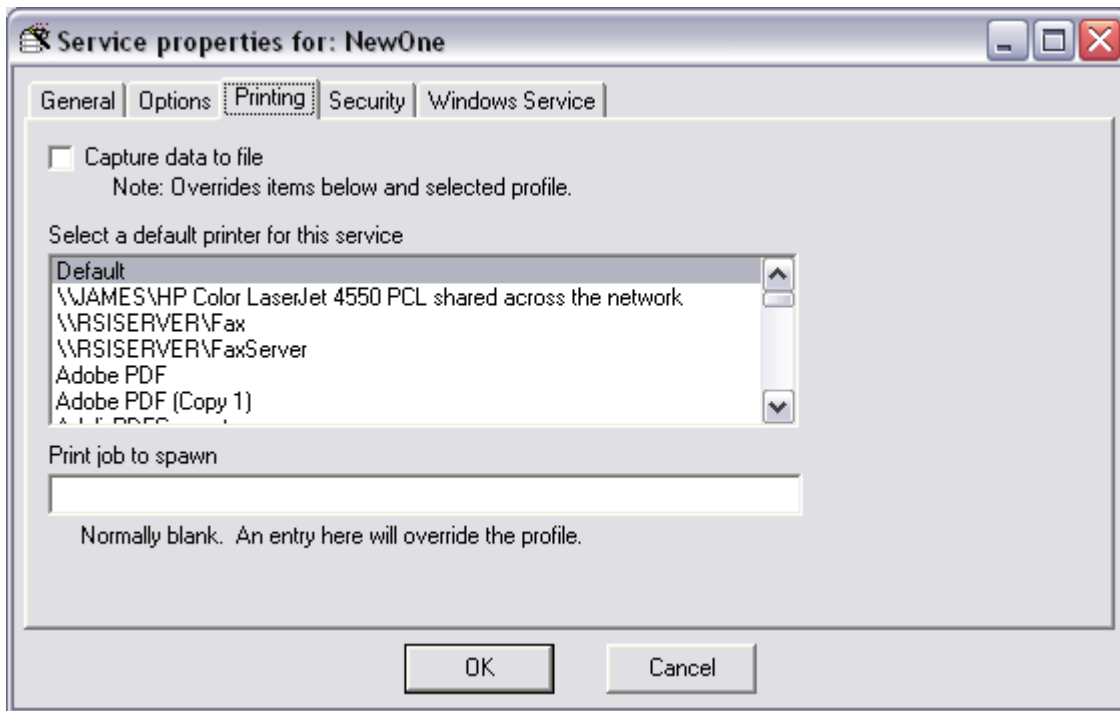


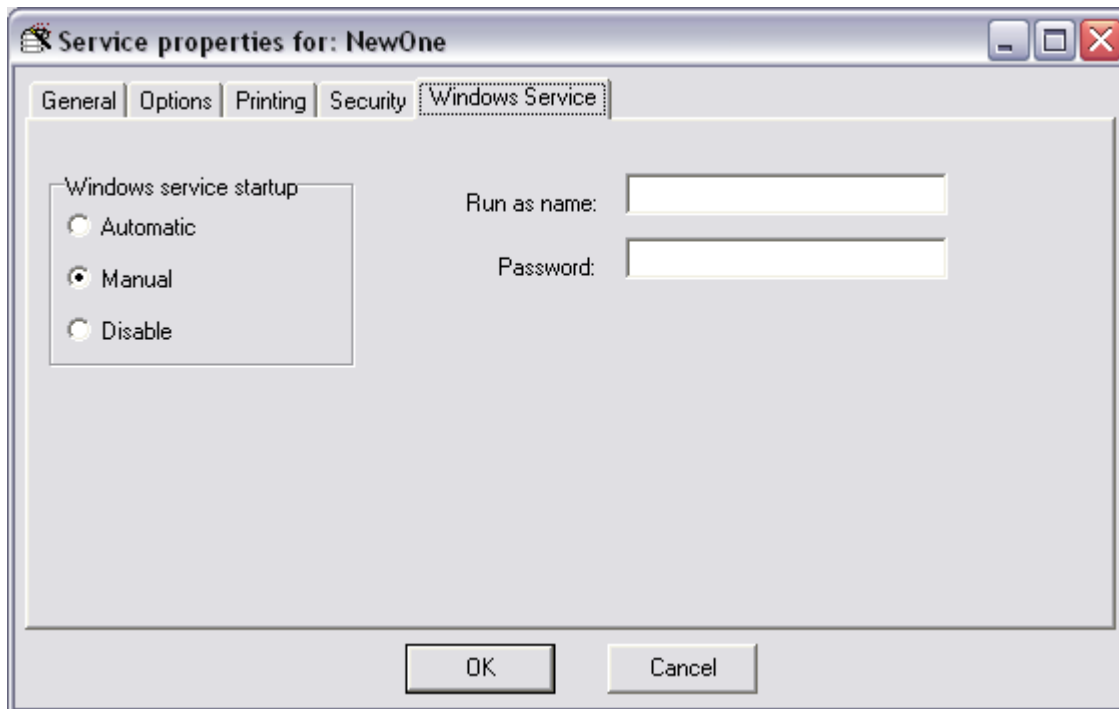
Note that a service name cannot contain any spaces or special characters. Once you select a service or add a new one, you go to the Properties screen to continue to set up or edit the service settings.



Note that the service type will change which options are available for that service (see the documentation on *Inside Services* for more information).





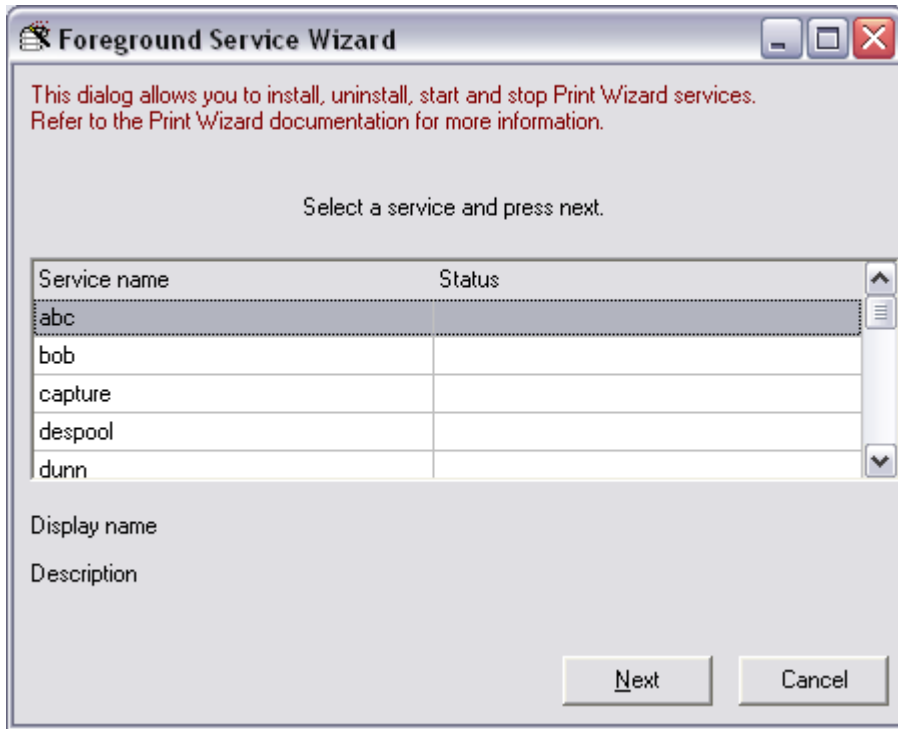


For each service, you need to set up the default printer, the options and any security rules you may need. See the documentation on the individual services and programs that run the service for more information.

These settings apply to both Foreground Services and Windows Services. See additional documentation on services for more information.



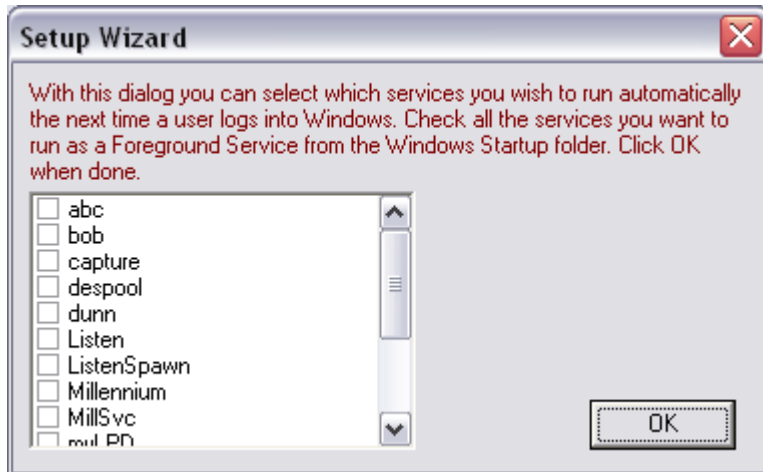
### 2.2.33 Tools : Foreground Service Wizard



Print Wizard supports two types of services, Foreground Services and Windows Services. This dialog walks you through running a Foreground Service. Select your service and follow the instructions.



## 2.2.34 Tools : Add service to Windows Startup



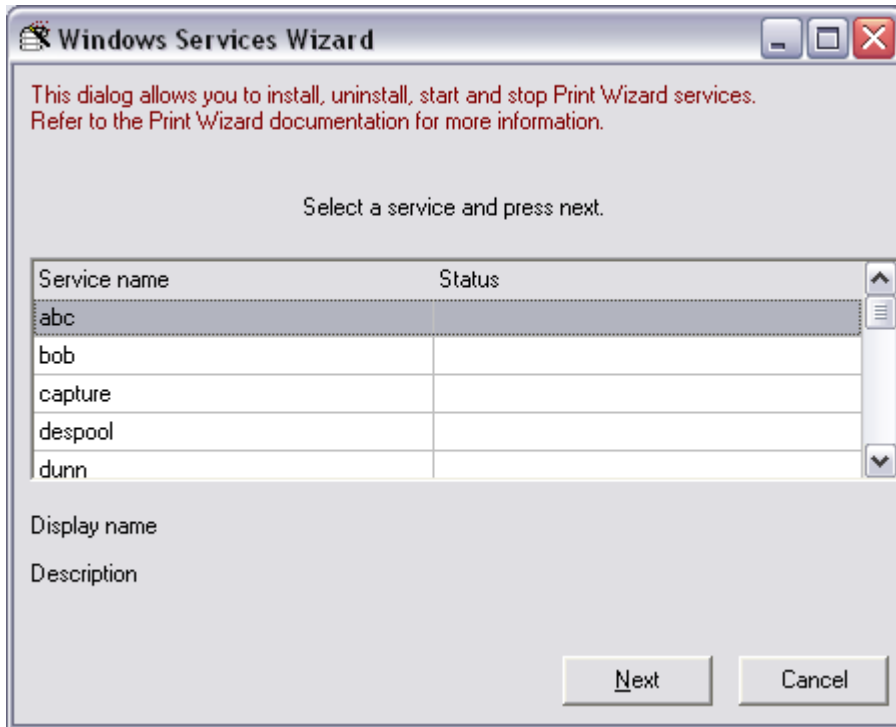
Through the Print Wizard user interface, you can select various services to run as a Foreground Service automatically when a user logs in.

When you click OK and have a service or services selected, a shortcut will be created for each in the Windows Startup folder. When a user then logs into Windows, this service will automatically start up.

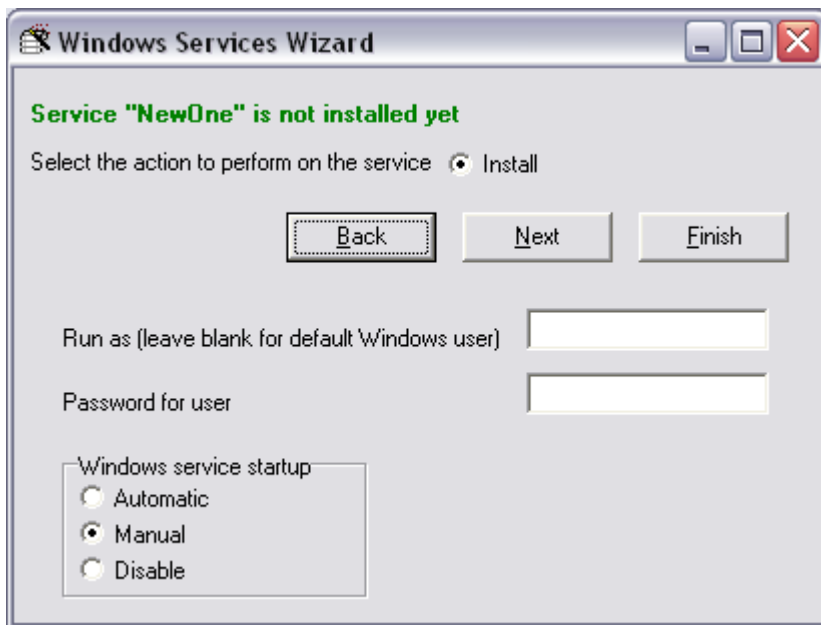
Note that the order of the items in the Windows startup folder usually dictates the sequence in which items will start and may determine who gains rights to what network port if there is any conflict.

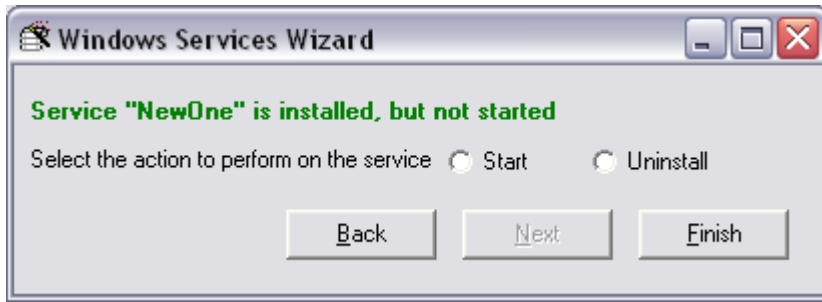
Also note that if the Windows machine does not have any defined users (logs right into Windows without user intervention), there may be issues with whether network drivers are fully up and initialized before services, such as “listen” and “LPD”, start. This is especially a problem with Windows 9X.

### 2.2.35 Tools : Windows Services wizard

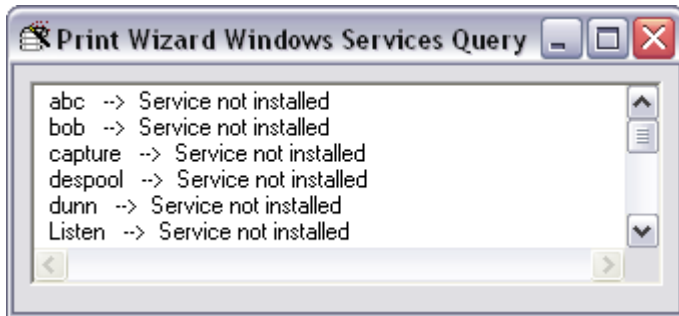


Similar to Foreground Service wizard, this walks you through the handling services. In this case, Windows Services need to first be installed then started in order to set them up for Windows.





## 2.2.36 Tools : Show Windows Services status



This shows the status of each service as it relates to Windows Services (whether it is installed, running, stopped, etc.).

## 2.2.37 Tools : Register file extension



You can provide a file extension that is to be registered as printable with the Print Wizard engine. This would provide Windows a default method to call Print Wizard to print certain types of file.

Care should be taken when using this dialog. You need to understand what you are doing in order not to break something else in Windows.

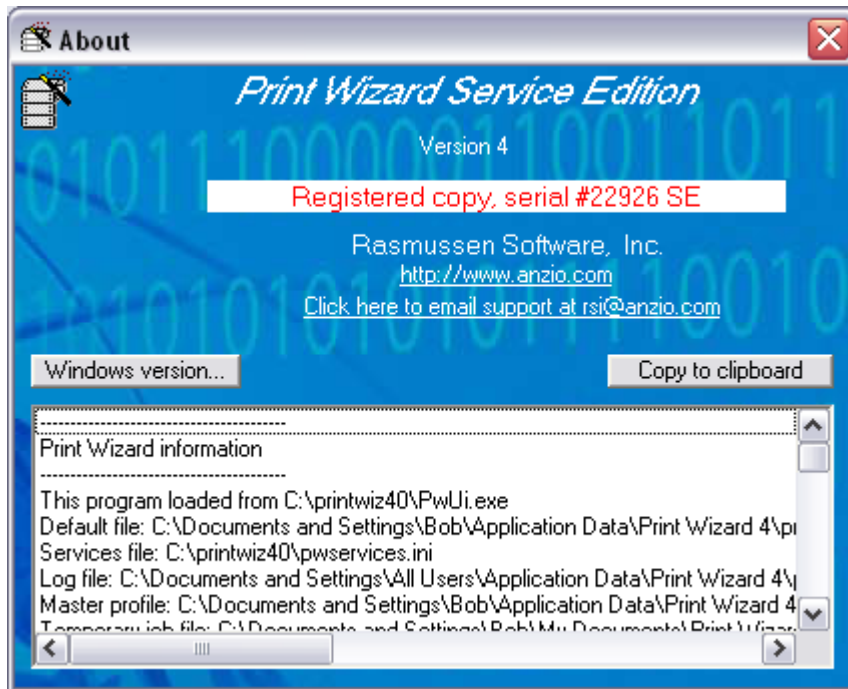
When you enter an extension, such as “pwml”, a file type association for “print” and “printto” will be added to the Windows registry. This will cause all files with the extension of “.pwml” to now print through Print Wizard.

This will create registry entries for the extension and for the program:

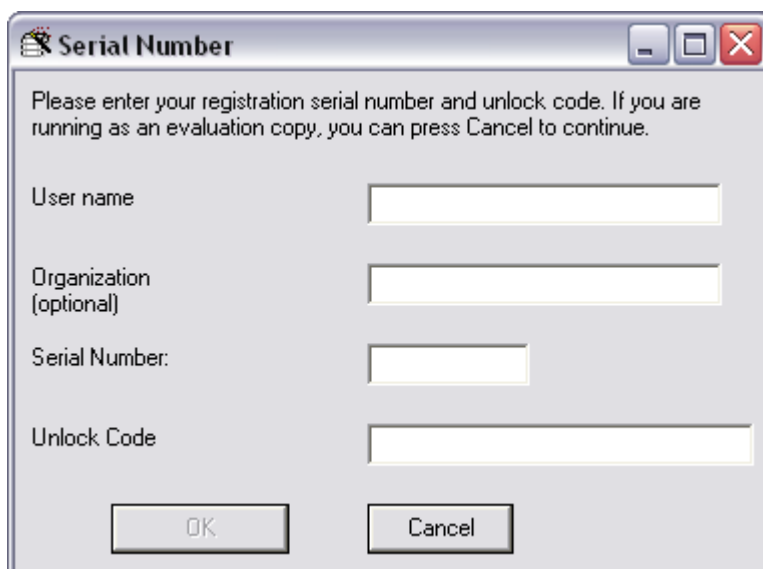
```
My Computer\HKEY_CLASSES_ROOT\.<ext>
  My Computer\HKEY_CLASSES_ROOT\PrintWizard\shell\print\command\
  My Computer\HKEY_CLASSES_ROOT\PrintWizard\shell\printto\command\
```

## 2.2.38 Help : About

This dialog shows the current version and information regarding the installed Print Wizard software. It also lists various version numbers for outside components it may use. This information can be helpful in debugging problems.



## 2.2.39 Help : Enter license



Enter license information here to activate your copy of Print Wizard. User name and organization are free form. Serial numbers and unlock activation codes must be obtained from Rasmussen Software.

### **2.2.40 Help : Print Wizard help**

This menu item takes you to our Print Wizard web page where you will find various documents and pricing information on Print Wizard. You can also get into our support pages from here where your questions will be answered.

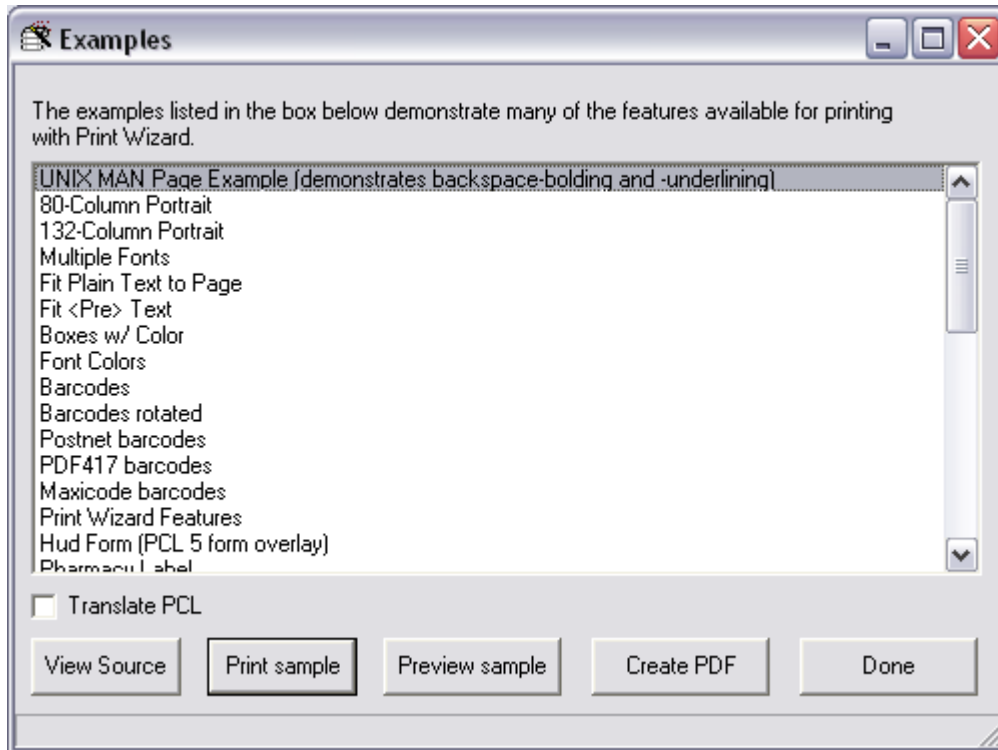
### **2.2.41 Help : Print Wizard manual**

This links to our web site Print Wizard manual for the current release. This document is in PDF format at our web site.

### **2.2.42 Help : Print Wizard updates at Anzio.com**

This link will take you to our web site where you can find pricing and update information as well as the current version of Print Wizard.

## 2.2.43 Help : Examples



Print Wizard ships with several example files demonstrating the use of Print Wizard.



## 3 For Developers

There are several options for developers wanting to use Print Wizard from their application or scripts. The Print Wizard engine comes in three basic forms:

- Print Wizard executable
- Print Wizard DLL
- WēPO (Print Wizard Active X)

Depending on your need and your environment, you can call the Print Wizard engine using one of these modules, and utilize its features for your need.

Also note that the examples mentioned above are a great source of reference as they demonstrate various ways to utilize the Print Wizard engine in your own application.

### 3.1 The Print Wizard Engine

The Print Wizard executable is the main engine used by most Print Wizard programs mentioned in this document. The Print Wizard executable is flexible with a wide range of command-line parameters and with the ability to work with a print profile.

You can run the engine by another application by simply doing a create process, shell execute or its equivalent action from within your program, passing command line parameters as needed.

Whenever you run Print Wizard from another program and in “quiet” mode (command-line switch of “/q”), Print Wizard will issue a return code upon completion stating whether it was successful in processing the file you sent. Below is a list of return codes possible from the Print Wizard engine:

0 Successful	11 License expired
1 No files to process or bad filename	12 Bad command-line parameter
2 Problem with profile	13 Listen failed
3 Bad printer name	14 Despool failed
4 Print failed for one or more files	15 Lpd failed
5 Email failed	16 Function (service) exceeds license
6 Fax failed	17 Socket hand-over failed for Listen
7 PDF generate failed	18 Service file missing
8 By-File-Type failed	19 Service name invalid
9 By-Custom failed	20 No service name found
10 Register extension failed	

A very simple example of running the Print Wizard engine from a Windows script is shown below:

```

\ Run this from Cmd as   wscript xxxxxx.vbs   for dialog messages
\ or                     cscript xxxx.vbs     for console messages

```

```

set shell = CreateObject("WScript.Shell")

for each filename in WScript.arguments
  `Process each command-line filename

  set pw = shell.Exec("c:\printwiz30\printwiz.exe /q " + filename)

  do while pw.status = 0
    WScript.echo "Print Wizard processing, processid is ", pw.processid
    ` current process id

    WScript.sleep 100

  Loop

  WScript.echo "Print Wizard complete for file " + filename
  WScript.echo "  return code", pw.exitcode          ` PW Halt Code
  WScript.echo "  program status code ", pw.status
    ` PW status (0 running, 1-terminated)

next

```

Check out the sections below on *Using Command-Line Parameters* and *Developer Examples* for more information.

### 3.2 Print Wizard DLL

The Print Wizard DLL is a callable Windows DLL for use from third-party programs. It is included with both Print Wizard Personal Edition and Print Wizard Service Edition for use by a calling program.

Note for AcuCobol users: The PWDLL.DLL must be in the same directory as the compiled object. The runtime configuration file must contain the line 'DLL-CONVENTION 1'

Additional information on using the DLL can be found below, along with several examples.

### 3.3 WēPO (Print Wizard ActiveX)

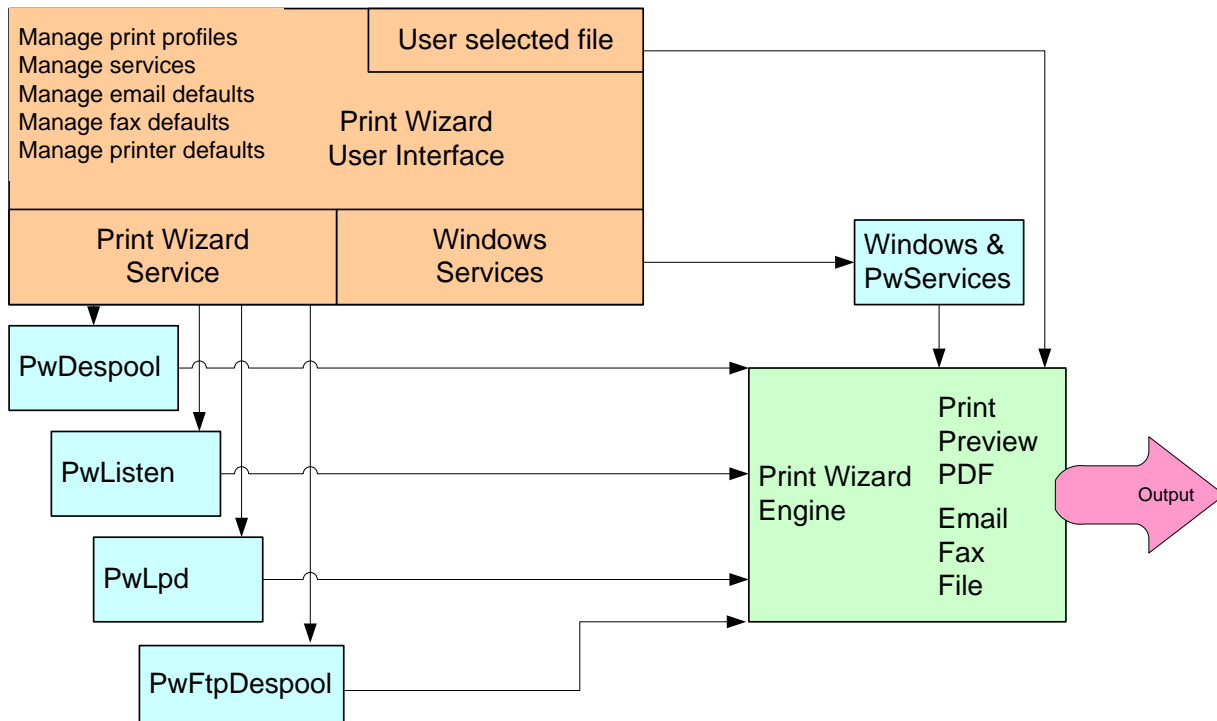
WēPO is an ActiveX, or "ocx", version of the Print Wizard engine, capable of being called from any application supporting ActiveX. This includes web pages and Internet Explorer, along with custom third-party web services programs.

Additional information on WēPO can be found below.

# **Print Wizard Technical Documentation**

## 4 Behind the User Interface

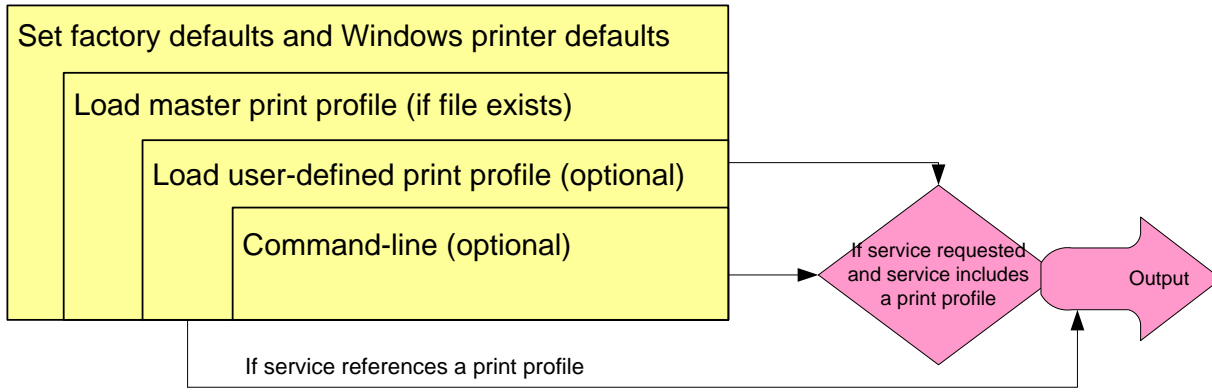
The *Print Wizard User Interface* is a central management point for controlling Print Wizard and its many components.



Print Wizard and its many applications all contain the same startup procedures. These procedures include loading factory and Windows defaults first, then loading a master profile (if present), then any requested user profiles and finally, and top priority, is any command-line parameters.

The only exception to this is if a Windows or Foreground Service Name is requested, and that Service Name record points to another print profile. This print profile then is loaded last, overriding any current settings.

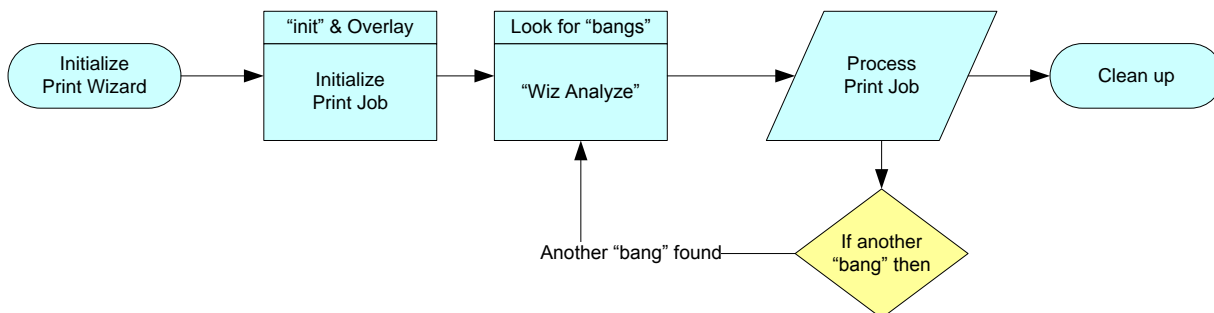
Whenever you are printing from the user interface, the Print Wizard engine is passed current settings in the form of a special profile called “pwui.job”. This profile name is used for this purpose only, but does follow the exact layout of print profiles. In fact, if you have a job that ran just the way you wanted it to, and you want to save it off for later use, you can rename the “pwui.job” profile to some other name with a “.profile” extension and then it will be available for use as a normal profile.



## 5 Behind the Print Wizard Engine

The Print Wizard engine is actually a stand-alone Windows program that takes command line switches as arguments. The Print Wizard program (printwiz.exe) is normally initiated by another program or a Windows script of some type, whereby the proper arguments are included.

When the program runs, it looks for a series of defaults in a specific order in order to initialize the program settings by loading default settings, global settings (master profile), user-requested print profiles and printer settings, and command-line parameters. It then initializes the print job, including the load of any printer “init” file or any overlay file. Once initialization is complete, it starts processing the job, by first retrieving the initial block of data to analyze what to do and how to proceed. If there are any “bang” commands (defined below) in the initial block of the file, these will affect the job flow from then on, processing each “bang” command as if it were a separate file. It then analyzes and prints the data, closing everything down when finished.





## 6 Using Command-Line Parameters

Print Wizard command line parameters are numerous. Below is a list containing a brief description of the command line, what the switch setting is, and a brief description of the command line parameters. These can be passed from another program or script, can be set by the user through the Start : Run dialog or from a Windows command (CMD) prompt, or by including them in a Windows shortcut that the user initiates.

When a particular parameter can be set to one of several possible values, those values are shown here separated by slashes. For instance, “/LOG = ON / OFF” means you can enter either “/LOG=ON” or “/LOG=OFF”.

Command line parameters are handled in the order in which they are included on the command line with a few exceptions noted below. You can precede a command line parameter with a dash, ‘-’, or a forward slash, ‘/’.

---

The first set of command line parameters set logging and quiet options

---

/L<logfile>	The name of the log file, sets logging to true
/L	Sets logging to true
/LOG=ON	Sets logging to true
/LOG=OFF	Sets logging off
/QUIET	Sets “quiet” on
/Q	Sets “quiet” on
/QUIET=OFF	Sets “quiet” off

---

Then we look for a profile

---

/PROFILE=<profilename>	Load a specific profile parameter file
------------------------	--

---

Then we look for a service type

---

/SERVICE=<servicename>	Load s specific service by "servicename"
------------------------	--

---

Then we look for type of email or fax

---

/EMAILCDO	Sets email type to CDO
/EMAILMAPI	Sets email type to MAPI
/EMAILOUTLOOK	Sets email type to OUTLOOK
/EMAILSMTP	Sets email type to SMTP
/FAXMAPI	Sets fax type to MAPI
/FAX32	Sets fax type to FAX32
/FAXWINFAXPRO	Sets fax type to WinFax Pro
/X	Sets fax type to WinFax Pro

---

Finally, we process any remaining command line parameters

---

Following is a list of all command line parameters:

/A<comma-list>	Attach files to printout, PDF, etc
/ASKFORFILES	Prompt the user for which files to print
/ASKFORFILES=<directory>	Prompt the user for files to print from a specified



	directory
/AUTOTAB[=ON   OFF]	Turns on tab-separated-values processing
/BIN=<pattern>	Bin(s) on printer to pull paper from
/BH	Print 2-sided, with binding horizontal
/BV	Print 2-sided, with binding vertical
/BL	Print 2-sided, with binding on left
/BLANKSAFTER=<n>	Print <n> blank pages after job
/BLANKSBEFORE=<n>	Print <n> blank pages before job
/BN	Print 1-sided
/BT	Print 2-sided, with binding on top
/C<baud>	Baud rate for raw-level serial printing
/COLLATE[=ON   OFF]	Should multiple copies be collated?
/CONFIRM[=ON   OFF]	Require user to confirm email send
/COPIES=<n>	Number of copies to print
/D	Delete input file after completion
/DEBUG[=ON   OFF]	Control output of debugging information
/DESPOOL	Set despool mode
/E	Eject page at end
/EMAIL=<commalist>	The email address(es) to send file to
/EMAILBCC=<commalist>	Sets the blind copy list for email
/EMAILCC=<commalist>	Sets the CC list for email
/EMAILCDO	Sets email type to CDO
/EMAILCDO=<commalist>	Sets email type to CDO, specifies whom to email to
/EMAILFROM=<fromname>	Sets who (the name) the email is from
/EMAILFROMADDRESS=<fromaddress>	Sets the address the email is from
/EMAILHTML[=ON   OFF]	Email in HTML, otherwise plain text
/EMAILMAPI	Sets the email type to MAPI
/EMAILMAPI=<commalist>	Sets the email type to MAPI, specifies whom to email to
/EMAILOUTLOOK	Sets the email type to Outlook
/EMAILOUTLOOK=<commalist>	Sets the email type to Outlook, specifies whom to email to
/EMAILPASS-<password>	Sets the password for emailing
/EMAILPORT=<port number>	Sets the port number for emailing
/EMAILSERVER=<servername>	Sets the email server name
/EMAILSMTP	Sets the email type to SMTP
/EMAILSMTP=<commalist>	Sets the email type to SMTP, specifies whom to email to
/EMAILUSER=<username>	Sets the email username (login)
/F=<filename>	Sets the file or superfile to output to
/F<filename>	Sets the file or superfile to output to
/FAX	Sets output type to fax
/FAX=<commalist>	Sets output type to fax, sets whom to fax to
/FAX32	Sets output type to Fax32

/FAX32=<commalist>	Sets output type to Fax32, sets whom to fax to
/FAXCOVERPG=<filename>	Sets the file name to use for a fax cover page
/FAXFROM=<from name>	Sets who the fax is from
/FAXFROMNUM=<from number>	Sets the fax number the fax is from
/FAXPROFILE=<name>	Sets a profile for faxing from Outlook
/FAXMAPI	Sets the fax type to MAPI
/FAXMAPI=<commalist>	Sets the fax type to MAPI, sets whom to fax to
/FAXWINFAXPRO	Sets the fax type to WinFaxPro
/FAXWINFAXPRO=<commalist>	Sets the fax type to WinFaxPro, sets whom to fax to
/FIRSTPAGE=<n>	Sets which page to start printing from
/H	Displays help information
/HELP	Displays help information
/I	Sets file input character set as ISO
/INIT=<initfile>	Specifies a print initialization file
/INPUTFILTER=<sequence>	Sets one or more processes to filter the text input
/INTERACTIVE[=ON   OFF]	Runs in interactive mode
/INTERACTIVE=<n>	Runs in interactive mode, with certain features available
/J<subject>	Sets the subject line for email or fax
/JOB-COPIES=	Sets the number of times to repeat an entire job
/K	Sets the keepalive flag
/KIOSK	Runs in kiosk mode
/L	Turns logging on
/L<logfilename>	Turns logging on, sets the log file name
/LASTPAGE=<n>	Sets the last page to print
/LISTEN	Runs in “listen” mode
/LOG[=ON   OFF]	Turns logging on
/M	Sets the “from” name for faxing or emailing
/MINIMIZE=[ON   OFF]	Starts the program minimized on the screen
/MINIPAGE=[ON   OFF]	Turns on mini-page logic
/N<notes>	Sets text for the coverpage of a fax, or for the body of an email. This can be a file name or actual text.
/NICEMARGINS	Sets margin handling to “ON”
/NICEMARGINS=OFF   ON   SAFE   EDGE	Sets margin handling
/OA	Sets orientation to automatic
/OL	Sets orientation to landscape
/OP	Sets orientation to portrait
/OVERLAY=<pattern>	Sets which overlay(s) to use
/OVERWRITE=[ON   OFF]	When printing to a file, overwrite an existing file
/P<prntername>	Sets printer name (or wildcard) to use
/PAGE-COPIES=<n>	Sets the number of times to repeat each page
/PDF	Generate a PDF file
/PDF://<parameters>	Generate a PDF file with superfile parameters

/PORT=<n>	Sets the port to use for Listen or LPD
/PREVIEW[=ON   OFF]	Sets Print Preview on
/PREVIEW=PAGE	Sets page mode Print Preview on
/PREVIEWOVERLAY=<filename>	Sets a filename to use as an overlay, but only in Print Preview; overlay file will not be printed
/PRINTFILTER=<name>	Sets the printer emulation filter
/Q	Runs in quiet mode
/QUIET[=ON   OFF]	Runs in quiet mode
/R	Renames the input file immediately
/RENAME	When writing to a file, rename an existing file of that name
/S	Show user the Printer Setup dialog, to prompt for which printer to use, etc.
/SERVICE=<servicename>	Sets which service to run
/SMARTPRINT[=ON   OFF]	Print file with unknown extension using Windows file association
/SPLITJOB=<n>	Split a job every <n> pages into separate output jobs
/SPLITSETS[=ON   OFF]	Each copy of a job (“job-copies”) becomes a different job.
/SUBJECT=<text>	Sets the subject line for email or fax
/T<to-name>	Sets the “to” line for email or fax
/TRANSLATEPCL=[ON   OFF]	Print Wizard will read and react to PCL
/TRANSLATEPCL=MAIN	Print Wizard will read and react to PCL for the main file, but not the overlay
/U	Sets the input character set to Unicode UTF-8
/VDEBUG[=ON   OFF]	Sets debugging on
/VDOCNAME=<docname>	Sets the document name
/VEXT=<extension>	Tells Print Wizard to register itself as the file handler for this extension
/VFONT=<commalist>	Sets the primary (and alternate) fonts
/VIEW	Turns on Print Preview
/VIEW=PAGE	Turns on page mode Print Preview
/VIEWONLY	Use Print Preview to view a file, but don’t allow printing
/VINIT=<initfile>	Sets the print initialization file
/VLEVEL=<print level>	Sets the print level (HIGH, LOW, SPOOLER, RAW)
/VOVERLAY=<pattern>	Sets the overlay(s) to use
/VPORT=<listen port>	Sets the port to listen on
/VRTL	Text is right-to-left, such as Hebrew or Arabic
/VTAGS[=ON   OFF]	If OFF, all tags will be printed
/VUSER=<username>	Sets the username
/W	Sets the wait-for-completion flag
/WAITFORFILE[=ON   OFF]	Wait for a file to appear, then print it

/X<faxnumber>	Fax with WinFaxPro to the indicated number
/ZOOM[=ON   OFF]	Print Preview window will be maximized



## 7 “Bang” Commands

Bang commands are one way that you can control how Print Wizard treats a particular print job. That is, should Print Wizard print it, make a PDF out of it, fax it, etc.? Other mechanisms for control include command line parameters and profiles; both of those control mechanisms take place *outside* the print file. Bang commands allow you to control the processing from *inside* the print file. That is, if you are programming the printout, in whatever programming language you use, you can also program Print Wizard’s processing of that file.

When Print Wizard is told to print a certain file, it looks first at the very beginning of that file. If the first printable data it sees (not counting spaces, returns, or linefeeds) is a “bang”, that is, an exclamation point (“!”) followed by one of the command words described below, it recognizes that file as a file of bang commands. It reads the entire *first line* as the command. That command applies to the following data, until it reaches either a) the end of the file, or b) another line starting with a bang command.

So in the “a)” case, the bang command dictates how the entire file is processed.

In the “b)” case, the file (the print job) is made up of multiple “sub-jobs” or “sections”, each one handled differently.

In most cases, the bang command word will be followed by multiple parameters, similar to command line parameters, that specify details and options for how to process the job.

### 7.1 Summary of Available Bang Commands

Following is a list of all the possible bang commands, what they do, and some important parameters:

<u>Command</u>	<u>Action</u>	<u>Key parameters</u>
!PRINT	Print the following section	What printer to use
!PREVIEW	Print the following section, but preview it first	What printer to use
!PDF	Make the following section into a PDF	Output filename, email recipient
!EMAIL	Email the following section <b>as is</b>	Email recipient
!FAX	Fax the following section, using the configured fax mechanism	Fax number
!FAX32	Fax the following section, using the fax32 mechanism	Fax number
!MAPISEND	Send the following section by email or fax	Recipient(s)
!FILE	Write the following section to a file	File name
!RUN	Run a specified program	Program to run
!RUNWAIT	Run a specified program, wait for completion	Program to run
!PARAMS	Save current settings	
!SET	Save current settings	
!RESET	Reload saved settings	
!ACROBAT	Deprecated	
!WINFAX	Deprecated	

By default, the !RUN command is disabled. You can enable it by manually editing the Print Wizard default file (“printwiz.ini”). Caution should be taken when editing this file. Call if you have any questions.

## 7.2 Parameters for “Bang” Commands

Bang command line will usually include some parameters to further instruct Print Wizard on how to process the next section of the print file. All parameters start with a forward slash “/”, are case INsensitive, and must be separated by at least one space.

Not every parameter can be used with every command.

Following is a summary of parameters:

/A<commalist>	Attach
/BIN=<pattern>	Input bins on printer
/COPIES=<n>	Number of copies to print
/COLLATE= OFF   FALSE   ON   TRUE	Whether to collate multiple copies
/C<coverpage>	Filename of coverpage file
/DELAY=<n>	Delay in seconds after processing this sub-job.
/EMAILBCC=<commalist>	Addresses for the blind copies list of an email
/EMAILCC=<commalist>	Addresses for the CC list of an email
/EMAILFROMADDRESS=<name>	The address an email will be from
/EMAILFROM=<name>	The person an email will be from
/EMAILHTML=OFF   FALSE   ON   TRUE	Whether email will be in HTML (or plain text)
/EMAIL=<commalist>	Recipients of email
/E <commalist>	Recipients of email
/FAXCOVERPG=<coverpage>	Filename of coverpage file
/FAXFROMNUM=<number>	The fax number a fax will be from
/FAXFROM=<name>	The person a fax will be from
/F <commalist>	Recipients of a fax
/J<subject>	Subject line of a fax or email
/JOB-COPIES=<n>	Number of times the sub-job will be repeated
/N<notes>	The file name or actual text of a the note on the coverpage of a fax, or the body of an email
/OVERLAY=<pattern>	Name(s) of overlay file(s)
/PAGE-COPIES=<n>	Number of times each page will be printed
/PREVIEW	Sub-job will be previewed before printing
/POSTVIEW	For a PDF output, the created file will be viewed after creation, by the Windows-associated program.
/P<printer>	Specifies printer to use
/T<to-name>	Sets the “to” name of a fax or email

/VIEW	Same as “/PREVIEW”
/X <commalist>	Sets fax recipients

“Bang” commands are designed to provide a means of separating print jobs within one print file. With the inclusion of “bang” commands, multiple print jobs can be processed by the Print Wizard engine even though they come from the same source file. There are some rules to follow however, in order for “bang” commands to work properly:

- In order to utilize “bang” commands, the first line of the print file must start with an exclamation mark (a “bang”), followed by one of the commands listed above. This list is constantly being added to and modified according to user requests, so it will change over time.
- Depending on the “bang” command, certain switches may be necessary for the data to get processed correctly. Check the documentation for each “bang” command.
- Most “bang” commands require that some data follow the command, even if it is just a carriage return / line feed, in order for the print job to be processed correctly.
- When the next “bang” command is found (after a carriage return / line feed), or end of file is encountered, the current process assumes an end of data and closes off that print job, starting another or ending the program.

### 7.3 !ACROBAT

```
!ACROBAT /x<fax#> /p<MAPI profile> /e<email address> /t<to> /j<subject>
/n<note> /a<attachment> /f<filename> /preview
```

!Acrobat is a special “bang” command for creating a PDF file through the Adobe Acrobat printer driver (provided with Acrobat Professional or Acrobat Standard), generating a PDF from the following data. If you include email information, this command can email it automatically using MAPISEND and the MAPI protocol (included for backward compatibility with prior releases – you can do this with Print Wizard by doing a !PDF /EMAIL=<> ...).

### 7.4 !FAX32

```
!FAX32 /x<fax #> /t<to> /j<subject> /n<note> /a<attachment> /c<cover pg file>
```

!Fax32 utilizes the Windows Fax 32 protocol to send a fax of the following data. A Microsoft Fax printer driver must be available for this to work properly.

Multiple fax attachments can be listed provided there are no spaces included. If a file name includes spaces, be sure to quote the whole attachment string with no spaces around the commas:

```
/a"c:\temp\test.doc,c:\testfile with blanks.xls"
```



When the “bang” line contains a `/n<notes>` command, it is assumed we need to fax a cover page. The defaults for “fax32” in the “printwiz.ini” file may specify a default cover page file name to be used. If none is specified, no cover page is sent. If you wish to override this behavior, you can include a `/c<coverpage>` where you specify the cover page (with full path) to use.

You can also send a cover page only by including the note data in the “bang” command line with no following data to process (see Developer Examples in the appendix).

## 7.5 !MAPISEND

`!MAPISEND` (passes any mapisend switch)

`!MapiSend` saves all the following data off to a file that then processes that temporary file via the MapiSend program. Any included switches are passed on to the stand-alone MapiSend program (see the MapiSend section later in this document). In this manner the data can be faxed or emailed according to the MAPI protocol and the Outlook profile.

`!MapiSend` does take advantage of the “Expand Variables” procedure described in the Appendix for placing Print Wizard global variables as command line parameters.

One additional note, if you end the list of switches with a “/n” followed immediately by carriage return and line feed, the following data will become the body of either the email or fax as is passed through to the MapiSend program.

## 7.6 !PREVIEW

`!PREVIEW` /p<printer> /f<print to filename>

`!Preview` allows you to specify that the following data should be processed as if for a normal Print Wizard “print” operation, but should be previewed first.

Note that in order for this to work smoothly, the data should not contain information that Print Wizard cannot process – “smart print” is not supported through “bang” commands.

## 7.7 !PRINT

`!PRINT` /p<printer> /f<print to filename>

!Print allows you to specify that the following data should be processed as if for a normal Print Wizard “print” operation.

Note that in order for this to work smoothly, the data should not contain information that Print Wizard cannot process – “smart print” is not supported through “bang” commands.

## 7.8 !PDF

```
!PDF      /f<filename> /email=<email address>
```

!PDF utilizes the Print Wizard engine to generate a PDF of the following information. If a “/email” is included, the PDF file is then emailed to the specified email address through your default email settings.

## 7.9 !RUN

```
!RUN      <program> (pass any switches you wish to include) %1
!RUN      start %1
```

!Run will save any following data into a file and pass that file off to an external program. This external program can be an executable or can be the Windows shell “start” command. You would include a “%1” to represent the file name location in the string you are running.

In the case of “start”, the file will be given to Windows to start according to the Windows file type (through the association of the file extension to the application).

Note that the !RUN command is not enabled by default. You must enable it by manually changing the Print Wizard default file (“printwiz.ini”). This command can be dangerous if not properly used.

The !Run command uses the “Expand Variables” described in the appendix to determine the position of certain global Print Wizard information including where to place the file name in the string of switches (command-line parameters to be passed to the external program).

Also note that this “bang” command does not require any data following it. However the %1 and the %filename that you can use as a variable name, will refer to the temporary file created from data that follows, not from already existing data (see example below).

## 7.10 !WINFAX

```
!WINFAX   /x<fax #> /t<to> /j<subject> /n<note> /a<attachment>
```

!WinFax uses the WinFax PRO object calls to fax the following data and any included attachments to the indicated fax number. Symantec’s WinFax PRO version 10.03 is recommended for this to work properly.

## 7.11 An Example

Note that there are more examples in the appendix under Developer Examples. This example will print data to a specific printer, preview data for the default printer and then print to the default printer:

```
!print /p"HP Color LaserJet 4550 PCL"  
Aaaaaaaaaaaaaaaaaaaaaaaaaa bbbbbbbbbbbbbbbbbbbbbb cccccccccccccccccccc  
!preview  
Dddddddddddddddddddddd eeeeeeeeeeeeeee ffffffffffffff ggggggggggggggggggggg  
!print  
Hhhhhhhhhhhhhhhhhhhhhhh hhhhhhhhhhhhhhhhhhhhhhh hhhhhhhhhhhhhhhhh  
Iiiiiiiiiiiiiiiiiiiiiiiiiii iiiiiiiiiiiiiijjjjjjjjjjjjjjjkkkkkkkkkkkkkkk lllllllllllllllllll
```



## 8 Inside Profiles

### 8.1 Default Settings

The default settings file is called “printwiz.ini”. While it is not a print profile, one thing worth mentioning here is that some of the global settings for certain items that make it into your default file, may originate here. This would be items such as default command strings, and strings used for spawning the Print Wizard engine. Also any object calls, such as to the WinFax PRO printer driver, will have a default global entry in the default file.

### 8.2 Master Profile

The master profile is a print profile that is loaded immediately after any default settings are loaded and is used to override the majority of your Print Wizard and printer settings from their factory or Windows defaults. This is a normal print profile, so refer to the documentation below.

### 8.3 User Print Profiles

Print profiles are used by Print Wizard to carry forward a current set of printer and Print Wizard settings, so they may be remembered and re-used from job to job. These files exist with an extension name of “.profile” and are located in the same directory as the Print Wizard engine and the executables.

Print profiles are nothing more than your typical Windows “ini” file, defining a value for a specific key. Each key relates to a printer, Print Wizard, fax or email setting and are called by each program when specified on the command line when running that program. The one exception is that a “service” setup may point to a specific print profile and force load it whenever it runs a job.

There are several optional settings found in the print profile that the user cannot access through the user interface, but are not documented in detail. Email us if you have specific needs and cannot figure out how to make it work.

### 8.4 User Interface Profile

Whenever you are printing from the user interface, the Print Wizard engine uses a special profile called “pwui.job”. This profile name is used for this purpose only, but does follow the exact layout of print profiles.

As an option, if you have a job set up just the way you wanted it to, and you want to save it off for later use, you can rename the “pwui.job” profile to some other name with a “.profile” extension and it will be available for use as a normal profile.

## 8.5 Extension Profiles

One other type of print profile should be mentioned here. Again, this is another rare use for a profile. Though it can appear as a normal profile to the user, this profile is automatically used under certain circumstances.

An extension profile has a name based on a file extension it should apply to. There is a flag in the “printwiz.ini” default file that states whether to use automatic extension profiles. “UseExtProfiles” is set off by default (set to ‘0’). To enable this, you can set its value to true, “1”.

When the “UseExtProfiles” is true and there is a profile that starts with an extension for the file you are printing, those profile settings are automatically loaded after all other options and just prior to printing the file.

As an example, you may have a profile called “.afx.profile” that forces a fax of any file with that extension when printed through Print Wizard.

## 8.6 Partial Profiles

A normal profile file contains many settings when created through the user interface. However, it is possible to create any profile as a partial profile, containing only those lines (keys) that you want it contain.

This is particularly handy when you want to override something with the master profile. You can create a “master.profile” in the current executables directory that contains only those lines you want to override certain settings with. As an example, suppose we only want to override the orientation and the default debug settings, you could set a master profile with just these lines in it:

```
[General]
IniVersion=3.0

[Profile]
psorientation=1

[Print]
Debug-print=1
```





## 9 Inside Services

Services are a special way of retrieving files from remote sources. As discussed earlier, there are two types of services supported by the Print Wizard Service Edition package: Foreground Services and Windows Services.

At present, there are four types of services:

### Despool

The “despool” service watches a local or network directory for any files fitting its criteria and then prints those files, deleting them when they are complete.

### FTPDespool

Similar to “despool”, this service watches a specific directory for files to print. However, in this case the directory resides on an FTP server somewhere remote.

### Listen

The “listen” service waits for some other client to “push” a print job to it. It then collects the file, and prints it. This is similar to the HP JetDirect protocol and works with most host systems print spoolers and the “netcat” or “netprint” utility programs.

### LPD

The “lpd” protocol is one whereby the host client, running “lpr”, send the print job to a specific print queue name on the “lpd” server, where it will get processed according to the settings for that queue (“queue” here should not be confused with the Windows printer queue or the UNIX / Linux print queue).

Foreground Services for “despool” and “listen” can run from the command line without the need to set up a service record. The FTP Despool and LPD services require a service record as you need to define information pertaining to these services before they can run successfully.

Windows services **MUST** have a service record set prior to the installation of a service. This is because Windows registers services by a name in advance of running that service. Also there is no user interface or command line options possible upon install of a Windows service.

When you create a service through the user interface, the service information is stored in the “pwservices.ini” file. This file, like print profiles, is a Windows “ini” file. Unlike profiles, where there is a file for each print profile, there is a single services file which has to be located in the same directory as the executable for the service, pwservices.exe.

## 9.1 Despool

### 9.1.1 How does it work?

Print Wizard's "despool" service is designed to watch a directory or a number of directories, for the presence of specified files or wildcard file matches, and then print those files.

You can set up the "despool" service to watch one directory  
c:\myprintjobs\despool\\*.\*

or to watch multiple directories with wildcards, such as  
c:\myprintjobs\ap\\*.txt, c:\myprintjobs\hr\\*.prn, c:\myprintjobs\ar\invoice\*.\*

or to even watch for just one specific file  
c:\myprintjobs\acc\checkreg.txt

The "despool" service, running either as a Windows Service or as a Foreground Service, will watch these locations for any files dropped into there by remote or local application.

Once a file is found, it is checked against the "despool" service "rules" for validity and if ok, copied off to a temporary location and removed (so as not to be reprinted again). The temporary file is then handed over to the Print Wizard engine to process.

### 9.1.2 Rules

For a service, you can define a set of rules that govern who and what can print through that service. The rule set is very basic. You can allow or deny everything and then add exceptions to allow/deny printing certain files smaller or larger than a threshold, or allow/deny printing certain file extensions.

This example allows all files unless larger than 30k or smaller than 50 bytes:

```
*=allow  
>30000=deny  
<50=deny
```

The second example limits files based on extension:

```
.txt=allow  
.spl=allow  
.gif=allow
```

The third example allows only text files:

```
.txt=allow
```

If no rules exist, such as would be the case when you run without a service name, or when the service name rules is empty, the default is to allow everything. As soon as a single rule is added, the default becomes to deny everything.

## 9.2 FTPDespool

### 9.2.1 How does it work?

Print Wizard's "FTP despool" service is designed to watch a directory on a remote host through the FTP protocol. This service obeys all the rules of the FTP server which may or may not require a log in, which may or may not allow files to be deleted from the host by the client and which may or may not be accessible through firewalls (depending on your network configuration).

You can set up the "FTP despool" service to watch a user's home directory (by leaving the directory blank) or to watch an initial, specific, directory, such as  
despool

But unlike the "despool" service, wildcard and file names are not matched. Instead, if you need to exclude or include only certain file extensions, you can do this through "FTP despool" rules.

The "FTP despool" service, running either as a Windows Service or as a Foreground Service, will watch the location specified based on some time interval for any files placed there by a remote host's application., and process them through the Print Wizard engine.

Once a file is found, it is checked against the "FTP despool" service "rules" and if okay, the file is fetched to a temporary location. The service will then attempt to remove it from the server (if it has proper permissions) and index the file as having been printed (so as not to print it again). The temporary file is then handed over to Print Wizard to process.

### 9.2.2 Rules

For a service, you can define a set of rules that govern who and what can print through that service. The rule set is real basic. You can allow or deny everything and then add exceptions to allow/deny printing certain files smaller or larger than a threshold, or allow/deny printing certain file extensions.

This example allows all files unless larger than 30k or smaller than 50 bytes:

```
*=allow  
>30000=deny  
<50=deny
```

The second example limits files based on extension:

```
.txt=allow  
.spl=allow  
.gif=allow
```

The third example allows only text files:

```
.txt=allow
```

If no rules exist, such as would be the case when you run without a service name, or when the service name rules is empty, the default is to allow everything. As soon as a single rule is added, the default becomes to deny everything.

## 9.3 Listen

### 9.3.1 How does it work?

Print Wizard's "listen" service is different than the prior two, which despool files dropped into local or remote directories. The "listen" service actually monitors a network port for incoming print jobs, very similar to what the HP JetDirect devices do.

By default, the Print Wizard "listen" service listens on port 9100 for incoming data, assuming it to be a print job.

You can run multiple "listen" services to monitor a number of network ports where host applications may be sending data. Each port can have its own set of parameters as to what to do with the data.

If you need to exclude or include only certain hosts, you can do this through "listen" rules.

The "listen" service also requires you have the specified port open in your firewall if you are printing to or from outside your local network.

The "listen" service, running either as a Windows Service or as a Foreground Service, will monitor the port for any data coming in. This data may be sent by a remote application, a local application or through a remote systems printing subsystem. Also many hosts include utility programs that can also print to these network IP ports, such as "netprint" and "netcat".

Print Wizard contains very basic "netprint" and "netcat" example code for those systems that require it. If you have specific questions about your host operating system, give us a call.

### 9.3.2 Rules

For a service, there is a set of rules that govern who and what can print through that service. The rule set is very basic. You can allow or deny everything and then add exceptions to allow/deny printing from certain hosts.

If no rules exist, such as would be the case when you run without a service name, or when the service name rules is empty, the default is to allow everything. As soon as a single rule is added, the default becomes to deny everything.

There is a check item for listen services that determines whether you are checking IP addresses or host names and IP address when you set up listen rules. If the "NoHostCheck" setting is true, then only IP addresses are verified.

This example limits jobs to a specific IP range (where host name is reported as an IP address):

```
h192.168.*=allow
```

If host names return as names (host lookup), you can limit based on name:

```
h*.site.com=allow
```

If you want to limit access to a single host, do so by including the host name or IP:

```
hmyunixserver.site.com=allow  
h192.168.1.232=allow
```

If you want to allow everything except for a specific range, use wildcards (remember that as soon as a rule is seen, the default is "deny"):

```
*=allow  
h66.126.*=deny
```

The “NoHostCheck” flag is important if you are having any type of DNS or reverse DNS look up slow down. This procedure can be slow under certain circumstances. You can only turn this off or on through services in the user interface or by editing the services file directly. Try changing the state of this flag and see if your performance improves.

## 9.4 LPD

### 9.4.1 How does it work?

Print Wizard's "lpd" service is somewhat similar to the "listen" service, except much more robust. The "lpd" service follows the rules of the LPR/LPD protocol which most all host operating systems implement for remote printing. The "lpd" service monitors a network port for incoming print jobs, port 515 by default. The difference is that the "lpr" protocol (the host end client) designates a specific print queue on the remote print server (your Windows machine running Print Wizard's "lpd" service) for which a job is destined.

When you set up an "lpd" service for Print Wizard, you can specify the print queue and apply various parameters for any job that may be received for that print queue.

You could set up a print queue for faxing, for emailing, for creating PDFs, etc. Your host system can then be set up with different printer names which print to different print queues, which would get directed to the remote print queue for Print Wizard. Your host application simply needs to print to the right printer.

The "lpd" service, running either as a Windows Service or as a Foreground Service, will monitor the port for any data coming in. This data is placed into a temporary file and the proper file is processed according to the print queue settings.

### 9.4.2 Owners

One item that despool and listen services does not have to worry about is ownership. In the case of LPD, owners have the rights to kill and query the status of certain jobs. While a normal user may (depending on host permissions) be able to post a print job, only root and the owner of the queue will have permissions to kill jobs from the PC end of the queue.

The "owners" in the service record for LPD is a comma separated list of users that have the right to manage the queue.

If you define rules of any type, your default becomes to deny all access. Be sure to at least add the necessary users, the owners and "root" if you want any of them to have access to that print queue (both for printing and for print management).

### 9.4.3 Rules



As with the other services, rules are available to control access and prevent misuse. These rules once again are pretty basic, covering just enough to prevent unauthorized access. You can prevent both host (host name and IP) and users from access.

In the case of LPD services, once you add rules, there should be at least one entry for "user" and for "host" or no one will have access. If you have set an "owner" of this list, you will need to make sure that owner does have access as well. Check your service record set up.

This example limits jobs to a specific IP range (where host names return as IP addresses):

```
h192.168.*=allow
u*=allow
```

This example limits jobs to a specific site for any user with the first name of 'john':

```
h*.site.com=allow
ulibrary*=allow
```

This example limits access to a single host:

```
hmyunixserver.site.com=allow
u*=allow
```

This example allows everything except for any users with login names ending in 'smith' (remember that as soon as a rule is seen, the default is "deny"):

```
*=allow
h*=allow
u*smith=deny
```

To deny everyone except "root" from printing through the LPD process, you need set up only this one rule:

```
h*=allow
uroot=allow
```



## 10 The Print Wizard Engine

This section describes the logic of the “engine” of Print Wizard; that is, the code that actually prints the printfile.

Any reference to “printing” here can also refer to generating PDFs, TIFs, EMFs or faxes.

### 10.1 Beginning Assumptions

Print Wizard starts off assuming several things about how you wish to print. Any of these assumptions can be overridden, in several ways, as described later.

Print Wizard’s beginning assumptions are:

- 1) that the file contains printable text, and Print Wizard should handle the printing.
- 2) that you wish to print on the printer identified by Windows as your default printer.
- 3) that it will use the Windows printer driver; that is, do “high level” printing.
- 4) that paper size, orientation, bin, duplexing (2-sided printing), etc. are as selected in the “properties” of the chosen printer, at the Windows level.
- 5) that printing should be done in the “Courier New” font. This font is found on every Windows system, is scalable in each direction, has good character set coverage, and is mono-spaced. Furthermore, auto-font switching is enabled; see below.
- 6) that printing should begin as high and as far to the left on the page as the printer is capable of printing.
- 7) that the printfile is in the “DOS” or “OEM” character set.

Again, any and all of these assumptions can be overridden in a number of ways.

### 10.2 Detecting Special File Types

If Print Wizard knows a file name for the print data, it will examine the extension of that file name to determine the file type. Certain file types will be diverted, and printed in other ways. Specifically:

- Image files, such as .TIF, .JPG, etc., will be sent to a special routine that prints them as full-page images.
- Metafiles, including .WMF, .EMF, and Windows spool files (.SPL) will be sent to a special routine that prints them as full pages.
- Files with a .PDF extension will be read by Print Wizard’s built-in PDF reader, then treated as full pages.
- Files with an extension .manifest or .manilink will be treated as manifest files. These are text files containing the names of other files to be printed.
- If the “Smart Print” feature is turned on, Print Wizard will check the extension and determine if a different Windows program should do the printing. If so, Print Wizard will initiate that process. So a .DOC file, for instance, will be sent to Microsoft Word to be printed.
- If the file starts with an exclamation point (a “bang”) followed by a command word that Print Wizard recognizes, Print Wizard will treat it as a bang command file. When processing the bang command file, **parts** of the file may be sent through the processing described below.
- If the file appears to contain PCL-5 data and the “pclbang” switch is turned on, Print Wizard will examine the beginning of the file to see if the initial printable data is a bang command. If so, it will be treated as a bang command file, and parts of it will be sent back through the processing described below.
- Any other file is assumed to be a text file, and is processed as explained in the following sections.

Each of these special cases is described elsewhere in this manual.

Bang file processing and Smart Print are not part of AnzioWin, the DLL, or WePO.

## 10.3 Print Level Switching

Print Wizard starts off by assuming that it will print at a high level; that is, it will use the Windows printer interface to draw characters (and other things) on each page. This high-level print is Windows' preferred (and assumed) way of printing, and provides access to the Windows spooler functions, as well as support for printers connected via serial, parallel, network, USB, and other means.

This approach is a problem if the data stream contains escape sequences, such as a sequence that puts an Epson printer into compressed mode. This could be the case if the data stream is being generated by an application program that knows (or thinks it knows) what kind of printer you have, and is trying to control that printer. If printed at high level, these escape sequences would be "drawn" on the page, and they would not be obeyed; that is, they would not affect the printer's printing. So print jobs of this type must be printed at "spooler" level. However, if Print Wizard is configured to translate PCL, and it sees escape codes, it will go into its PCL translation mode.

An "Escape" is a specific control character, having a decimal value 27, or hex 1B. It is sometimes represented as "\E" or as "^[".

Likewise, if the print job contains PostScript code, printing it at high level would result in a listing of the PostScript instructions, instead of the intended printout. Print Wizard detects PostScript jobs by seeing if they start with "%!". These print jobs must also be output at spooler level.

So, if Print Wizard detects either PostScript or escape sequences (and it is not translating PCL), it will switch automatically to spooler level. In spooler level, Print Wizard writes directly to Windows' spooler, bypassing the printer driver. The printer can still be connected via parallel, serial, network, JetDirect, etc.

(Earlier versions of Print Wizard used "raw" level; this has been superseded by the spooler level).

## 10.4 Detecting Special File Types By Initial Content

Print Wizard examines the first data in the file (after initial white space – returns, linefeeds, and spaces) to determine if special handling should be applied. Specifically:

- If the file appears to be PostScript (starting with "%!"), the data is sent directly to the printer at spooler level.
- If the file starts with "<!DOCTYPE" or "<HTML" or "<PWML", Print Wizard identifies it as a PWML/HTML document; that is, as a document containing markup. See the section "PWML Syntax" below. Note that text within the document will still be treated as described here, by default.

## 10.5 Basic Text Handling

This section assumes Print Wizard is writing text at a high level and not to the spooler.

A font has been chosen by some means, and character size, spacing, and margins in each direction have been determined (more on that later). What follows is a description of Print Wizard's basic text handling.

Print Wizard assumes line-oriented text, with a linefeed (hex 0A) and optionally a carriage return (hex 0D) at the end of each line. Each line of input is assumed to be intended as one line of output. If lines are too long to fit within the margins, the line may be broken at the character (the default) or the word; or characters may be discarded, according to the WRAP mode, described later.

If a tab (hex 09) is encountered, Print Wizard will treat it in one of two ways. In the *classic* approach, Print Wizard assumes that tab stops are every 8 columns, and will advance to the next tab stop. However, if *auto tab* is on, Print Wizard will treat the job as a tab-separated values file – see below.

If a backspace (hex 08) is found, Print Wizard checks to determine whether it is being used to cause overstriking of the same character (backspace bolding) or underlining. It will convert these cases to actual bolding and underlining, respectively. For other cases of backspacing, Print Wizard will overprint the characters as indicated.

If a Field Separator (hex 1C) is found, Print Wizard will adjust the horizontal position to align fields; more on that later.

If a carriage return (hex 0D) without a linefeed (hex 0A) is encountered, Print Wizard will print the next line over the top of the current line. Some legacy reports use this technique for bolding and/or underlining. Print Wizard will print the characters correctly, but will currently not convert the data to a bold or underlined font.

When a form feed (hex 0C) is found, or when no more lines will fit on the page, Print Wizard will advance to the next page.

Characters above hex 7F are printed according to the current *character set*; more on that later.

Print Wizard will also attempt to avoid printing blank pages at the beginning of the print job.

## 10.6 Auto-fit Logic

Print Wizard attempts to determine the best way to fit the text to the page. It does this by analyzing the first several hundred lines of data.

In a plain text print job, this begins at the beginning of the file. However, if the file contains markup (to be explained later), the text to be auto-fit starts immediately after a `<pre>`, `<listing>`, `<xmp>`, `<legacy>`, or `<plaintext>` tag.

Print Wizard makes its decisions based on convention and readability. For instance, it is conventional to have 10 characters per inch horizontal spacing, and it is conventional (at least in the United States) to have 66 lines per page. But it would not be readable to have more than 30 characters per inch.

Print Wizard analyzes lines to determine the maximum number of characters per line. In doing this, it knows that certain Asian characters are double wide, and that certain characters are combining marks. It then checks to see the page size selected, or more precisely, the printable area. This is based on information returned by the Windows printer driver. Based on these factors, it determines character width, pitch, and left and right margins.

It also checks to see whether the data is paginated; that is, if it contains form feeds. If so, it determines maximum lines per page; that is, the lineage. If the data does **not** contain form feeds, Print Wizard attempts to do "lineage guessing". It will look through the print data, looking for patterns of blank or similar lines, as well as for occurrences of the word "Page", and do a best guess as to what is the implied page length. If it can not make a good guess, it will assume that 66 lines equal a page.

Once lineage is determined, Print Wizard will test whether that number of lines can be printed reasonably on the page. If not, it assumes 6 lines per inch. With a lines-per-page measure determined, it calculates character height, line spacing, and top and bottom margin.

By default, Print Wizard will assign the minimum possible margins for the top and left; printing will begin as high and as far to the left as the printer is capable of printing. This is generally the same point at which the printer will print if no printer driver is involved, so text positioning should be correct for preprinted forms. However, there is also an option to have Print Wizard assign "nice margins"; that is, margins up to ½ inch at the top and left. This may be preferable if the printer can print to the extreme edges of the page. Setting options is described elsewhere.

Any of these calculated values can be overridden with markup tags. Some of them can also be adjusted in the *Print Preview* screen.

You can get an inside look at Print Wizard's calculations by turning on the debug printing option. This will cause Print Wizard to display information about its calculations as it goes along.

## 10.7 Mini-page Logic

Print Wizard has an additional feature that makes it easy to print a small number of lines, such as might occur when printing one label at a time, or when printing variable-length receipts.

If a) the selected paper bin is 8 (tractor), 7 (auto), or 15 (automatically select), and b) you haven't specified paper length, etc., and c) the printer can print to the very top of the page, and d) the number of lines in the print job would constitute less than 80% of a page, if printed at 6 lines per inch (or specified line spacing), then Print Wizard will automatically adjust the page size to exactly fit the text, at 6 lines per inch (or specified line spacing).

So suppose Print Wizard gets a print job containing exactly 12 lines, and it is configured to use a tractor-feed printer with letter-size paper. After applying the tests above, it configures the print job for a custom paper length of 2 inches, and then prints the 12 lines.

This can be repeated indefinitely, and each print job will use 2 inches of paper. Thus, if continuous-feed label stock is used, with 2-inch vertical spacing, printing will be correct and no labels will be wasted.

If the printer driver doesn't identify the paper source as Windows standard "tractor", you may need to specify your BIN as "8", "7", or "15" using the markup language as described below.

If you specify line spacing, that line spacing will be used in the calculation above. You can specify line spacing directly, or you can do so implicitly by specifying a *font size*.

If you specify a *top margin*, that amount will be added into the calculation of page length.

The *mini page* logic can be turned off if you do not want this feature.

## 10.8 Auto-rotation Logic

One of the most basic formatting questions is orientation: which way on the paper should the job print? The usual options are portrait and landscape.

Print Wizard has an additional option called AUTO. If ORIENTATION has been set to AUTO (as explained later), Print Wizard will start off assuming an orientation of portrait. However, if one of two thresholds is crossed, it will automatically switch to landscape.

The first threshold is characters per line. The default is 100; that is, if Print Wizard detects at least 100 characters of data in the widest line, it will switch to landscape. You can override this default with a parameter "ROTATEWIDTH=n", where n is a number of columns.

The second threshold is the width of the character. After Print Wizard analyzes the print job, and determines the length of the longest line, it calculates how wide each character needs to be (the pitch). If the needed width would be less than or equal to a certain value, it will switch to landscape. The

default is 1/16 of an inch. You can override this with the “ROTATEPITCH=n” parameter, where n is a value in Dots or specified units.



## 10.9 Character Set Issues

When Print Wizard receives characters beyond the ASCII range (i.e., greater than hex 7F), it has to know what character set they are in, in order to print them correctly. The character set is specified in different ways, depending on the form of Print Wizard that is being used. Print Wizard is able to handle characters in the following sets:

- ISO: The "Windows" character set (varies by national Windows version)
- OEM: The "DOS" character set (varies by national Windows version), usually containing line-drawing characters
- UTF-8: Unicode characters, encoded in the UTF-8 scheme

In AnzioWin (which contains Print Wizard logic), there is also the capability to translate passthrough print data according to several additional character sets.

Print Wizard looks at the beginning of a file for a special sequence of characters that identify it as containing UTF-8. This sequence in hex is EF BB BF, which is the UTF-8 representation of the Zero-Width Non-Breaking Space (ZWNBSpace) as defined in the Unicode standard. If you save a file as UTF-8 in Windows Notepad, for instance, this sequence is placed at the beginning of the file. If Print Wizard finds this sequence, it will set its *character set* to UTF-8 (possibly overriding other settings).

Print Wizard Markup Language (PWML) or HTML can also set the character set to UTF-8 using the META tag.

When Print Wizard is analyzing a print job to determine maximum line width, it takes into account the fact that most Far East characters are rendered twice as wide as Latin characters.

When Print Wizard prints the line-drawing characters found in the OEM (DOS) character set (also in UTF-8), it will use internal bitmap characters, to ensure that lines print as connected properly.

Print Wizard has some support for bidirectional languages (Arabic, Hebrew), on Windows platforms that include such support. This includes all versions of 2000, and later (if that feature has been installed and enabled).

## 10.10 Automatic Font Switching and Diacritics

Print Wizard has sophisticated handling of "font coverage". That is, it will detect when the font it is using does not contain the character it needs to print. When that happens, it will search through other fonts available on the Windows system to find one that contains the needed character. If the data being printed is mono-spaced, it will try first to find a mono-spaced font that will work; failing that, it will check variable-spaced fonts also.

Print Wizard also takes special care with combining diacritics. If it needs to print a character/diacritic combination that does not exist in any font, it will print the base character, and then try a variety of techniques to find the needed diacritic: a non-combining form, an alternate form, a replacement character, or an internal bitmap.

All this happens automatically, so that Print Wizard can render successfully an extremely wide range of characters.

You can control which alternate fonts Print Wizard will use, and their order of preference, as described below.

## 10.11 Auto Tab and Tab Handling

The data file may contain tab characters (hex 09). Print Wizard has two approaches to interpreting tab characters: *classic* and *auto tab*. The behavior Print Wizard uses is determined by the AUTOTAB configuration setting; if it is ON then auto tab handling will be done.

### 10.11.1 Classic tab handling

In the classic approach, it is assumed that tabs are set at every 8th column, starting at column 1 (so 1, 9, 17, 25, etc.). Print Wizard keeps track of a “column counter” as it prints (even with a variable-pitch font). When it encounters a tab character (control-I, hex-09), it advances to the next tab stop.

### 10.11.2 Auto tab handling

When AUTOTAB is active, however, an entirely different approach is used. This approach is provided so that you can easily print tab-separated-values (TSV) files. These files can be exported by various applications, including Microsoft Excel, for instance. The intent of Print Wizard is to print these files in a logical, readable manner, without user intervention.

The TSV file must follow this format:

- 1) The file is a line-oriented text file, with linefeeds or return/linefeeds between lines.
- 2) Each “record” in the file is in one line.
- 3) Fields (columns) within a line are separated by tab characters. Fields may or may not be padded with spaces.
- 4) Each line that contains tabs should have the same number of tabs, and therefore the same number of fields.
- 5) The **first** line that has tabs may or may not be a header line, with text describing each column.
- 6) There may be lines in the file that have no tabs. These are treated as normal text.

When Print Wizard analyzes the file before printing, it will determine the maximum width needed for each column, adding one space for separation between columns. In total, this determines the maximum line size needed for the report; this will affect the auto-fit mechanism in Print Wizard. If the maximum line size is too big to be legible (would print at more than 30 characters per inch), Print Wizard will adjust the tabs. This results in a multi-line output for each input line, with a separator line between records.

Print Wizard will also analyze the data in each column to see if the data appears to be numeric in all fields within the column (but the first tabbed line is ignored, because it might be a header line). This determines whether the column is a numeric column.

As it prints the data, Print Wizard will follow these rules:

- 1) When a tab character is encountered, it will advance to the next column location.
- 2) If the data hits the right margin, it will wrap to the next line (on a character boundary, not a word). This will not disrupt the tab handling.
- 3) If a column is numeric, it will be printed right-justified. The header text for a numeric column will also print right-justified.
- 4) If the file is generating multi-line output, Print Wizard will draw a thin line below each record (input line), and will not do a page break within a record.
- 5) Lines with NO tabs will be printed as normal text.

This logic will work in either a fixed-space or a variable-space font.

## 10.12 Print Wizard's Preview

Print Wizard has the option to open a Print Preview window, before printing each print job. In the freestanding Print Wizard (Printwiz.exe), this option is turned on with the “/view” or “/preview” command line parameter. In AnzioWin, it is controlled in Edit:Advanced Options:Print Preview or in File:Printer Setup.

The Print Preview window will show the first page (only) of the job to be printed. It will also show many of the calculated (or specified) printing parameters, such as “*lines per inch*”, “*characters per inch*”, “*lines per page*”, margins, etc. As is usual in a print preview, you can choose to print or to cancel.

You can also change many of the printing parameters. You can do this either by entering a new numeric value, or by dragging the corresponding line on the screen.

Many values are interrelated, such as left and right margin, columns, and columns per inch. If you increase the *left margin*, for instance, Print Wizard will compress the text horizontally, changing the *characters per inch*. If the print job contains no form feeds, and you change the *lines per page*, Print Wizard will break the page in different places.

Changes are made immediately to the sample shown on the screen. If you click *print*, the document will be printed with the changes you have indicated.

Changes are temporary; they will **not** be stored in any way.

The Print Preview window can be resized or zoomed to give you a better look at the page. Simply drag an edge of the window, or click the Zoom box in the upper right, or double-click the title bar to zoom or unzoom.

If you are running on a Tablet PC or on a version of Windows with Tablet features installed, you can annotate the preview image by drawing on it with the Tablet's stylus or the mouse. Right-click on the page image, and from the popup menu select “Allow ink markup”. This drawing will be printed on the first page (only) of your printout. You can control the pen width, color, etc., by using the popup menu, activated by the right mouse button or equivalent.

It is also possible to use the Preview window's popup menu to specify that label printing should start at a particular point. More on that later.

## 10.13 Programming with Print Wizard

This section describes how you can “speak Print Wizard's language”. If you have access that lets you change the content of the file that you feed to Print Wizard, you can control Print Wizard's printing, a little or a lot. Even if you can't alter the original file, you can tell Print Wizard to use another file as an initialization file.

### 10.13.1 Column Alignment with FS Characters

Print Wizard provides a fairly simple way to create reports with aligned columns, even when printing with variable-spaced fonts.

Reports generated by many legacy programs achieve alignment by simply using spacing. For instance, the "customer name" field in every line begins at column 30. This works fine, as long as it is printed in a mono-spaced font – one in which every character has the same width (such as the Courier New font). However, if you instead print in a variable-spaced (proportional-spaced) font, such as Arial, the alignment is destroyed. Customers have asked for an easy way to have both – column alignment and variable-spaced fonts. You can achieve this using the "Field Separator" character, or FS. This control character has a value of hex 1C. Here is how it works:

When Print Wizard first analyzes the print job, it determines an intrinsic *character width*, based on the number of characters per line. This value is retained for the entire report.

When Print Wizard prints each line, it keeps a *column count* – the number of characters that have been printed.

Whenever it encounters an FS character, Print Wizard does realignment. It sets the horizontal printing position to the product of the current column count times the intrinsic character width. So variations that have happened because of differing character widths are discarded.

Furthermore, if there are one or more space characters immediately after the FS, Print Wizard advances the width of the zero character, which is usually also the width of each numeral. That way, columns of numbers will align properly in most cases.

### 10.13.2 Print Wizard's markup language, briefly

Print Wizard has a markup language to allow you to control most aspects of the printing. This markup language is contained within the file being printed. So to change the printout, you may need to be able to alter the program that generates the print job. However, it is also possible to put some initial markup into a *print initialization* file, which Print Wizard will process before the primary print job. This way you can alter the printout *without* changing the original program.

Print Wizard's markup language (PWML) is based on HTML. This has several advantages. It is well-known by most programmers and web designers. It is based entirely on printable ASCII characters. It can be created with any programming language. And it adapts easily to dealing with text-only data.

But HTML is lacking in two key areas: precision and pagination. It does not allow precise placement of print elements. And because it is designed primarily for the screen, not the printer, it has no concept of a page break. (Some of these issues are being addressed with CSS). So Print Wizard's markup language extends HTML in these and some other areas.

Conversely, Print Wizard is targeted at printing “line printer” type legacy printouts. However, it can print some kinds of HTML documents. In this document, we will use “PWML” and “HTML” as virtually synonymous. In actual use, the initial tag (see examples below) can be either <PWML> or <HTML>.

Print Wizard allows you to move incrementally into the markup language. Suppose you have a print file that contains 132 columns of data. If you feed this to Print Wizard when it is set to portrait orientation, Print Wizard will compress the font horizontally in order to fit 132 columns onto the page. But now suppose you require that report to print out in landscape orientation. You can simply add to the beginning of the file:

```
<HTML><pagesize orientation="landscape"><BODY><PLAINTEXT>
```

and you will accomplish your objective. Similarly, you could specify other “job-level” parameters such as paper size, bin selection, font, line spacing, and lines per page. You could even specify a form overlay (or watermark). But the original text, after this inserted line, would not need to be changed at all.

In fact, you can place these initial sequences in a separate file, and instruct Print Wizard to prepend (insert at the beginning) that file. This is called a “print-init file”.

At the next level, to use bolding, italics, or underlines, you could insert simple tags such as “<B>” to start bolding and “</B>” to end bolding, in the body of your report. You could also indicate special characters, such as “&euro;” for the Euro character. You would need to change the <PLAINTEXT> above to <PRE>, so that tags would be recognized.

But in <PRE> mode, your existing data might contain things that looked like tags (“<”) or like character entities (“&”), so you’d have to change “<” to “&lt;” and change “&” to “&amp;”. To avoid that need, there is <LEGACY> mode. In <LEGACY> mode, tags (starting with “<”) and character entities (starting with “&”) must be preceded by a “trigger” control character, which is normally control-Z (hex 1A); otherwise, they are not recognized, but are printed as plain text.

Finally, you could advance to including bitmaps, barcodes, rectangles, etc.

A later section explains the details of Print Wizard’s markup language.

## 10.14 Photo and Image Printing

The freestanding Print Wizard program can also assist with what *should* be the simple task of printing a photo; that is, a file in an image format with a file extension .BMP, .GIF, TIF, .JPG, or .JPEG.

When Print Wizard is told to print a file with one of these extensions, it will print it as large as possible, given the printer’s printable area. That is, the bitmap image will be stretched to fit the page. If the picture is wider than it is high, Print Wizard will print it in landscape orientation. Other options, such as paper type and graphics resolution, can be set in printer setup.

If the file is a multi-page TIF file, all pages will print using this logic.

It is possible, as explained below, to tell Windows to use Print Wizard as the designated printing program for various file types (extensions). When you do this for one of the image types, such as “.JPG”, it provides a very easy way to print your photos. While browsing a particular directory (in Windows Explorer), you can right-click on a JPG file. The popup menu that appears will include an option “print”. Just select that option, and Windows will use Print Wizard to print the file.

Also, you can create a shortcut icon for a particular printer on your desktop. Then you can drag and drop a file of the appropriate type onto that printer, and Print Wizard will print it on the indicated printer.



## 10.15 Printing Metafiles

Windows has long supported metafiles. A metafile is basically a recording of print operations, which can be “played back” at a later time. The first implementation was of Windows Meta Files, which carried a “.WMF” file extension. Later, the structure was enhanced to carry more information, and these became Enhance Meta Files (EMFs). It is important to note that a WMF and an EMF can contain only one page of output.

Many graphics programs can produce WMF or EMF files. If you have a choice between WMF and EMF, use EMF.

Later versions of Windows (NT and later) introduced a variation of EMF that is a “spool file”, and carries an extension “.SPL”. Windows uses these as a temporary storage format for print jobs on their way to the printer. An SPL file can contain more than one page, where each page is embedded as an EMF. It is possible to capture the SPL file that is generated by virtually any Windows program printing to any Windows printer, as described below.

Print Wizard can read WMF, EMF, and SPL files, and use them in various ways, including:

- As an overlay. An SPL file can be a multi-page overlay.
- As an included picture, similar to a bitmap. A metafile may be a smaller and more efficient file than the corresponding bitmap file.
- To be printed directly.
- To be sent as a fax.
- To be converted to a PDF file.

Print Wizard can also **produce** EMF files from any of its usual input sources.

## 10.16 Capturing a Spool (SPL) File

As mentioned, recent versions of Windows use the SPL format to store a print job on its way to the printer; that is, while it is “spooled”. With a few easy steps, it is possible to capture the SPL file. For instance, you might have a form that you have defined in Microsoft Word. If you print this from Word, and capture the SPL file, Print Wizard can use this as a form overlay.

### 10.16.1 Locating the SPL Directory

To capture an SPL file, you have to know where it will be created. The directory where Windows holds the SPL files is usually “\windows\system32\spool\printers”. Under Windows NT, it may instead be “\winnt\system32\spool\printers” or “\winnt\SYSTEM\CurrentControlSet\Control\Print\Printers”.

It is possible, however, to configure Windows to use a different directory for all printers, or for a specific printer. To find your PC’s general spool directory:

- 1) Go to the Control Panel.
- 2) Click on “Printers” or “Printers and Faxes”.
- 3) In the File menu, click on “Server Properties”.
- 4) Click on “Advanced”.
- 5) Note what is in “Spool folder”.

If in the steps below you fail to find a particular spool file, you may want to check the Windows Registry, under one of the following areas:

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Print\Printers\DefaultSpoolDirectory
- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Printers\*specific printer name*\SpoolDirectory

Once you have identified the spooler directory, proceed as follows.

### 10.16.2 The capture process

The trick to capturing an SPL file is to get it to stay in place long enough for you to copy it. You can do this by either pausing the printer (at the Windows level), or telling Windows to retain the print job after it is printed. The following steps detail the first method:

- 1) Open the Control Panel’s “Printers” or “Printers and Faxes” applet.
- 2) Double-click the icon for the printer in question. Ideally, there will be no print jobs showing.
- 3) From the Printer menu, choose “Pause printing”.
- 4) Print the document you want to capture, on the chosen printer, using whatever Windows software is appropriate. You should see the print job show up in the printer window on your screen.
- 5) Look in the spooler directory as identified previously. You should find a file with a “.SPL” extension (hopefully only one). Copy this file to some other location on your computer.
- 6) In the printer’s window, cancel the print job.
- 7) In the printer’s window, remove the “Pause” checkmark.

You now have your SPL file as copied in step 5.

### 10.17 Scanner as Input File

In any place Print Wizard might otherwise use a bitmap file, it can instead be told to pull an image from a scanner attached to your PC; in fact, from any TWAIN source. To specify this, you refer to the scanner by a special file name: “SCAN://?”. The name is NOT case sensitive. Whenever Print Wizard goes to open a file, it looks for this special name and, if found, it uses the TWAIN interface to read an image. Then the image is treated the same as another bitmap file.

It is also possible to specify certain options for the scanning, such as the scanner name, density, and number of pages. These are entered as options on the end of the filename, such as “scan://?pages=3”. For more details, see ###.

Print Wizard will present a dialog with a “Select” combo box and an “Acquire” button. In “Select”, choose which TWAIN source (scanner) to use (if you have more than one). Click “Acquire” to

activate the scanner's user interface. What you see from this point on will vary, because this is provided by the scanner's driver. When you actually scan an item, control will return to Print Wizard.

In the scanner's user interface, you can generally select the color depth and scan density (dots per inch) to be scanned. It is best to match these to your printer (or other output source) for maximum clarity and minimum processing.

Following are some suggested uses and notes:

- SCAN:// as the main file: The scanned image will be printed as large as possible on the page.
- As an overlay: The scanned image is used as a form overlay for the print job. For best results, scan at the same page size as you are printing (edge-to-edge).
- As a PREVIEWOVERLAY: The scanned image will be shown in the Print Preview window, but will not be printed. You might want to scan the preprinted form or label stock that the job will be printed on, and use the Print Preview dialog to adjust the spacing of the text to properly fit.
- As an included graphic: A PWML print job could call for an <IMG> tag where "SRC" was identified as "SCAN://". As it prints, it will prompt you to scan something. You could include a picture or a signature, for instance. If multiple <IMG> tags are specified in a document, you will be prompted for a scan for each one.
- Scan to Fax: When the output is one of Print Wizard's fax mechanisms, you can achieve programmatic faxing from a scan.
- Scan to PDF: You can scan an image and convert it to a PDF file, using either Print Wizard's native PDF generation or Adobe PDF Writer. Note that this process does not do Optical Character Recognition. So the PDF will contain one or more bitmaps, not text. In other words, it is not "searchable" PDF.



## 11 Using Print Wizard in AnzioWin

AnzioWin is our telnet/SSH (terminal emulation) program for Windows. More information can be found at:

<http://www.anzio.com>

We also have Anzio Lite, but it does not include the Print Wizard logic.

When Print Wizard is turned on in AnzioWin, it affects a) print screens, b) the PRINTFILE command, c) CAPTURE to printer, d) certain COPY commands, e) certain file transfers, and f) passthrough print, sometimes called aux print, transparent print, or attached print. If you are unfamiliar with passthrough print, see the reference documents on our website, listed above..

If the program generating the printout on the host system has some capability for specifying printer types, it may insert escape sequences, especially at the beginning of the print job, to format the printing. If Print Wizard sees these (and it is not translating PCL), it will switch to “spooler” level, and will do no further processing of the job. In this mode, the host program is “in charge”. That is, its escape sequences are responsible for configuring the printer. If they are wrong, there is nothing Print Wizard can do about that.

However, if you can tell the host system that you have a “generic” printer, it should send out plain text. This will allow Print Wizard’s logic to control the format of the printout.

With AnzioWin’s menu system, the user can select several things that influence Print Wizard’s operation, including:

In the File:Printer Setup dialog:

- Primary printer – setup: Choice of printer, orientation, bin, paper size, duplexing (2-sided printing)
- Print Wizard: Be sure it is checkmarked
- Destination: Select whether you want to generate a printout, a PDF file, a fax, or an EMF file, and set appropriate options.
- Preview: Check if you want Print Preview to show.
- Nice Margins: Explained at ###
- Alternate fonts: The list of fonts that Print Wizard should use (or not use) for non-ASCII characters.
- Primary font: the main font that Print Wizard should use.
- Debug printing: Turn this on to get information on Print Wizard’s decision making.
- Printing initialization file: Specify a file to be inserted at the beginning of the print job.
- Print level:Spooler, on printer: Which printer should be used if in Spooler mode.

In other places:

- File:Printer Font: The initial font *name* that will be used. Duplicates the “Primary font” selection above, but more complete.
- File:Flush Timer: Determines end of print job.
- Communicate:Character Set: In what character set is the data coming from the host?

- Edit:Advanced Options:Print Wizard: Options described below.

The font size chosen (in Printer Font) does NOT affect Print Wizard output, since font size is automatic.

Following are some specific notes.

In AnzioWin you can specify a different printer to be used for jobs printed at a spooler level, using “Spooler, on printer” as above. If you do so, then when Print Wizard detects escape codes (or PostScript code) and switches to spooler level, it will also switch to a using a different printer.

AnzioWin also lets you control (turn on or off) certain features of Print Wizard. This is done by going to the Edit menu, then to “Advanced options”, and finally to the “Print Wizard” tab. There you will see a checkmark for each of the following features:

- **Print Preview**      Should the Print Preview window be displayed before each print job? (Duplicates the setting in Printer Setup).
- **Auto Orientation**      If on, orientation is assumed to be *Portrait*, unless there are more than 100 characters across, or text would be narrower than 16 characters per inch.
- **Create Mini-pages**      Controls Print Wizard’s mini-page feature, as described in this document.
- **Guess Linage**      Controls the *linage guessing* feature, as described in this document.
- **Nice Margins**      If on, Print Wizard will create balanced margins, up to ½” top and left margins. If off, printing will start as far left and as high as possible.
- **Translate PCL**      If off, Print Wizard assumes the printer can handle PCL. If “Main and overlay”, Print Wizard will translate PCL print jobs and overlay files for use on any Windows printer. If “Main only”, Print Wizard will translate the main print job’s PCL, but will rely on the printer’s PCL support for PCL overlays, for efficiency and speed.
- **Auto tab**      If off, tab characters in the print data are assumed to use tab stops at every 8 columns. If on, files with tabs are assumed to be in Tab Separated Values (TSV) format.
- **Compress PDFs**      If on, PDF files generated by Print Wizard’s built-in generator will be compressed.

## 12 PDF Generation

Adobe Systems Incorporated has defined and published a specification called “Adobe Portable Document Format”. Files that are created in this format are commonly called “PDF files”, and are a popular way of storing, transferring, viewing, and printing data in a device-independent (portable) way.

Print Wizard is able to generate PDF files in the following ways:

- 1) Via Adobe PDF (Distiller), as a pseudo printer driver.
- 2) Via other pseudo printer drivers.
- 3) With its own built-in PDF generator.

For the first two methods, Print Wizard prints to a printer driver just as though it were printing to a printer. However, Print Wizard can specify the output filename to be created.

For the third option, the built-in PDF generator, Print Wizard has complete control, and no other software is needed.

### 12.1 The Built-in PDF Generator

Print Wizard’s built-in PDF generator allows it to produce industry-compatible PDF files with no additional software. This generator is developed entirely by Rasmussen Software. Following are some notes on how PDF is generated:

- Files are generated as PDF-1.4.
- Files are generated with a resolution of 600 dots per inch.
- Where reasonable, fonts are mapped to PDF’s built-in fonts. So Courier New and Lineprinter become Courier, Arial becomes Helvetica, and Times New Roman becomes Times (with variations for bold and italics). Also, System becomes Helvetica.
- Characters outside of the ANSI range will be output as bitmaps.
- There is support for Zapf Dingbats and Symbol.
- Characters in other fonts will be output as bitmaps. There is no font embedding or use of non-standard fonts.
- Line-drawing characters are output as bitmaps.
- Output files are not in “linearized PDF” format.
- Graphics are not individually compressed.
- The file as a whole is compressed, if “Compress PDFs” is set.
- Overlays are stored in the file only once, and invoked for each appropriate page. This yields considerable savings in file size.
- PDF files can be passworded, encrypted, and restricted. See ###
- Print Wizard can immediately send the generated PDF file to a PDF viewer.
- Print Wizard can email the generated PDF.
- There are numerous other things that PDF defines that Print Wizard does not yet support.

### 12.2 PDF Encryption

Print Wizard’s built-in PDF generator is able to encrypt the output file. This may mean that a password is required in order to view the file; that is, there is a USER password. It may also mean that the user is able to view but not print the password; that is, there are PERMISSIONS.

By default, the file is generated with NO encryption.

Encryption is specified by including three string variables in the PDF output specification: OWNER, USER, and PERMISSIONS.

The OWNER is the password of the person who owns the PDF. This is the main encryption key of the file. If you enter the owner password when you view the PDF file, the file will be decrypted and you will get complete access to the file, regardless of PERMISSIONS.

The USER is the password needed to view the PDF. If you open the PDF file with the user password, or there is NO user password (but there is an OWNER), you can view the file and do other operations only as allowed by the PERMISSIONS.

The PERMISSIONS variable can consist of the following letters, in any order:

- P Allows the user to PRINT the file.
- C Allows the user to COPY text and graphics out of the file.
- A Allows the user to ADD or modify text annotations, or fill in forms.
- M Allows the user to MODIFY the document.

These variables are inserted as part of the “superfile” designation of the output. For instance, in the command line to Printwiz.exe, you could specify an encrypted PDF file as:

```
/fPDF://outfile.pdf?OWNER=mypassword?USER=readerpassword?PERMISSIONS=P
```

## **12.3 Things You Can Do with PDF**

Nearly everything that Print Wizard can do when printing can also be done when generating a PDF. Here are some things you might not have thought of:

- Print with form overlays.
- Include barcodes.
- Convert a bitmap file to PDF.
- Translate a PCL file to PDF.
- Convert a TSV file to PDF.
- Scan a document and create a PDF.
- Print purchase order data, with a form overlay, adding a dynamically-scanned signature, and email it to a supplier.





## 13 PCL Translation

PCL is the printer command language developed by Hewlett Packard, and supported by most, if not all, HP laser printers, some HP inkjet printers, and many printers from other manufacturers. It is a language, based on escape codes, that controls such things as paper size, font selection, character size, rectangle drawing and more. PCL sometimes includes HPGL, a vector-based language originally developed for plotters. HPGL is able to draw straight lines, arcs, polygons, etc. Many printers that support PCL also support HPGL as an alternate language.

Finally, PCL is sometimes wrapped in PJI (Printer Job Language), a language that initiates and controls whole print jobs.

Print Wizard has built-in support for PCL, HPGL, and PJI, collectively referred to as “PCL translation”. This means that Print Wizard can read, understand, and process these languages, independent of a printer.

### 13.1 The Built-in PCL Translation

Print Wizard understands PCL 5e. This feature can be turned on and off as “TranslatePCL”. The mechanism to do this varies by which implementation of Print Wizard you are using. The same switch controls PJI and HPGL translation.

We will not claim that support for these languages is totally complete. However, it should cover the majority of situations. This translation is constantly being enhanced. If you encounter a PCL file that is not translated correctly by Print Wizard, please let us know and, if possible, email us the file.

Following are some notes on Print Wizard’s PCL translation:

- Supports emulation of the barcode fonts provided in HP’s barcode font cartridge.
- Supports PCL macros.
- Supports color in both PCL and HPGL.
- Supports soft fonts, including embedded TrueType fonts.
- Supports most PCL character sets.
- Supports a custom character set for Unicode UTF-8, initiated by <Esc> ( 8 Q
- Uses Print Wizard’s automatic font logic for printing non-ASCII characters.

### 13.2 Where Do PCL Files Come From?

Many application systems, particularly on Unix/Linux, can insert PCL sequences into print jobs in order to control printing on laser printers. This can range from the minimal, such as a code to switch into compressed print, to control of the entire job.

It is also possible to create PCL files from most Windows applications. To do so, install (if necessary) a PCL5 printer driver on your Windows PC.

Be sure you use a PCL 5 driver. PCL 6 is something entirely different. We recommend an HP LaserJet 3, 4, or 5 driver. Or use an HP LaserJet 4550 PCL-5 driver for color.

From your Windows application, select Printer Setup, and choose this driver. Checkmark the “Print to file” box. Now print your document. You will be prompted for a file name. Enter a file name or, if allowed, an entire pathname so you’ll know where the file ends up. Once you have located the generated file, we recommend you change its file extension to “.PCL”.

In this way you can make a PCL file from a Word document, an Excel spreadsheet, or a forms design program. This file can then be used as a form overlay, or as the primary document to be printed, faxed, or converted to PDF, as described later.

### **13.3 How You Can Use PCL Translation**

With TranslatePCL turned on (set to “Main and overlay”), you can do a variety of things:

- Use a PCL file as an overlay, even on a non-PCL printer (even a Windows-only printer).
- Use a PCL file as an overlay for a fax.
- Use a PCL file as an overlay in a PDF.
- Render a PCL file on any printer.
- Convert a PCL file to a fax.
- Convert a PCL file to a PDF file.
- Print a PCL main file on a PCL overlay, on printer, fax, or PDF.

When a PCL file is specified as an OVERLAY or PREVIEWOVERLAY, and PREVIEW is on, Print Wizard will translate the PCL file to use as an overlay in the Preview window, regardless of the setting of TranslatePCL.

### **13.4 When TranslatePCL is OFF**

Then TranslatePCL is turned OFF, Print Wizard assumes that you are using a PCL5-compatible printer. Specifically:

- If the main print file contains any escape codes, Print Wizard will switch to Spooler mode, passing the print job unchanged to the printer. Of course, if the escape codes don’t correspond to what the printer can do, you won’t get the results you want.
- If a file with a “.PCL” extension is referenced as an OVERLAY, Print Wizard will do some processing on it, to convert it into one or more overlay macros. These will then be loaded into the printer, and dynamically

combined with the contents of the main file. It is also possible to have Print Wizard save these macros in the printer for later use.

If you try this with a non-PCL printer, it won't work.

For overlays, the printer driver must also be a PCL 5 driver. Some laser printers support PostScript, PCL-5, and PCL-6. Install and use the PCL-5 driver.

### **13.5 When TranslatePCL is MAIN**

When TranslatePCL is set to "MAIN", only the main file will be translated. If a file with a ".PCL" extension is referenced as an OVERLAY, Print Wizard will convert it to one or more printer macros, assuming the printer can support this feature. This can be much more efficient, and faster, than Print Wizard outputting the translated PCL for every page.

Print Wizard does not support a PCL main file on (a) PCL overlay(s) if TranslatePCL is off.

## 14 PWML Syntax

As stated earlier, Print Wizard's markup language (PWML) is based on HTML. A book about HTML may be useful to those unfamiliar with this coding concept. This document assumes some basic familiarity with HTML.

Because of Print Wizard's focus on legacy data, it is not yet fully compatible with generic HTML. Many capabilities of HTML have not yet been fully implemented. Cascading Style Sheets (CSS) are not recognized.

### 14.1 Recognizing a Marked-up Document

Print Wizard detects marked-up data as a file starting with any one of (case insensitive):

```
<!DOCTYPE  
<HTML  
<PWML
```

Any white space (tabs, spaces, returns, or linefeeds) at the beginning of the file will be ignored when looking for these indicators.

### 14.2 Tag Basics

Print Wizard's tag structure, like HTML, consists of tags between less-than and greater-than, such as:

```
<PRE>
```

If you are in LEGACY mode (after a <LEGACY> tag), the less-than must be preceded with a "trigger" character, normally control-Z.

Most tags have additional parameters, such as:

```
<FONT face="Courier New">
```

If the parameter value contains embedded spaces, the parameter must be in quotes as shown. If there are NO embedded spaces, the quotes are optional.

The parameter value for some parameters must be numeric, implying there is a need for units of measure. These are described later.

Tags are case insensitive, and can be continued from one line to another.

## 14.3 End Tags

As with HTML, Print Wizard recognizes certain *end-tags*. So for instance, `<B>` will start **bold** printing, and `</B>` will end it. Similar tags that have an on/off behavior are `<I>` *italics* `</I>` and `<U>` underlined `</U>`, as well as others.

Other end-tags cause a return to settings that were in effect before the start-tag. For instance, if you are in a certain font such as Arial, then do a `<FONT>` tag specifying a FACE of “Courier New”, then do `</FONT>`, you will return to printing in Arial. Tags that behave in this way include `<FONT>`, `<P>`, `<DIV>`, `<SUP>`, `<SUB>`, and `<SPAN>`.

Every start-tag should have a corresponding end-tag, in reverse order.

## 14.4 The Bare Minimum

Print Wizard will not actually start printing anything until it gets into the body text part of the file. So the minimal set of tags is:

```
<HTML><BODY>
```

This will leave Print Wizard in a state of free-form text, where line endings are ignored.

Much of the design of Print Wizard, however, is focused on processing line-oriented text. That is, each line of input should be printed on a separate line. HTML standards refer to this as *preformatted* text, which is initiated with the tag `<PRE>`. For historical reasons in HTML, and special needs in Print Wizard, there are four other very similar tags: `<XMP>`, `<PLAINTEXT>`, `<LEGACY>`, and `<LISTING>`. Since any of these should be inside the `<BODY>` section, a minimal set of tags (that does not alter Print Wizard’s default behavior) is:

```
<HTML><BODY><PLAINTEXT>
```

The differences between the five tags above involve a) what kind of tags are allowed after each, b) assumed width, c) whether ampersand character entities (explained below) are allowed, and d) whether a trigger character is needed.

- After the `PLAINTEXT` tag, no other tag or character entity processing or is done.
- `LISTING` implies 132 characters per line, and `XMP` implies 80 characters per line. No character entities are processed. The only tag that is processed is the ending tag, `“/LISTING”` and `“/XMP”` respectively.
- After `PRE`, any tag is recognized, and character entities are processed. There is no assumed number of characters per line, but that can be specified with the `WIDTH` attribute.
- After `LEGACY`, any tag or character entity is allowed, but only if preceded by the trigger character, normally control-Z.

For new code, we recommend either `PRE` or `LEGACY`. Use `LEGACY` if the printable data following might include “&” or “<”.

## 14.5 Units of Measure

Standard HTML is ambiguous about placement of text, images, etc. For instance, font size is indicated from “1” to “7”, and bitmap widths use units of pixels. Print Wizard, however, is designed to allow precision, and pixel size varies from printer to printer.

For every numeric value entered in a tag, Print Wizard will assume a certain unit of measure. For instance, the tag

```
<FONT pointsize=12>
```

(which doesn’t exist in standard HTML) assumes a unit of “point” (1/72 inch). Units can also be relative to the current setting. To make the point size 2 points larger, do

```
<FONT pointsize=+2>
```

A minus sign (“-”) also works. Changes can also be specified in percentages:

```
<FONT pointsize=-25%>
```

If neither “+” nor “-” is included, the measurement is *absolute*. To enter an absolute amount that is less than zero, use the syntax “0-<amount>” such as

```
<IMG x=0-20>
```

Print Wizard does not process Cascading Style Sheets (CSS).

In Print Wizard, many measurements allow you to specify units exactly, with a two-character code immediately following the number, such as:

```
<PAGESIZE x=8.5in>
```

Allowable units are:

- pt = points (72 points = 1 inch)
- pc = pica (1 pc = 12 pt, 1/6 inch))
- in = inches
- mm = millimeters
- cm = centimeters
- li = lines (based on current line spacing)
- px = pixel

For many measurements, we have tried to make the assumed units compatible with standard HTML, which deals in dots or pixels of the screen. When translating this to the printer, we had to make some

assumptions. We have tried to maintain a certain amount of compatibility with existing browsers, as well as with earlier versions of Print Wizard.

Print Wizard takes the following approach. Each document starts off with an assumption of a virtual pixel density of 120 dots per inch (DPI). This is similar to the density used by various browsers when they print. Thus a horizontal rule (HR) that has a WIDTH of 300, will be printed as 300 DOTS wide, or 2.5 inches. Similarly an image (IMG) with a WIDTH of 480, will be printed as 480 IMG-DOTS wide, or 4 inches. Metric equivalent is 304.8 dots per centimeter.

If Print Wizard encounters a <PAGESIZE>; tag, which is non-standard HTML, it then assumes a DPI of 720 (or 1828.8 dots per cm), and a density for images (IMG-DPI) of 300 (or 762 dots per cm). Thus bitmaps that are scanned at 300 DPI will print in true size, regardless of the printer's capabilities. Finally, within either the <PAGESIZE> tag or the <BODY> tag, DPI can be set with a DOTSPERINCH attribute, and IMG-DPI can be set with an IMGDOTSPERINCH attribute.

Thus if DPI is 720, then a DOT unit is 1/720 of an inch. Likewise, IMG-DPI determines the size of an IMG-DOT. If a tag places a rectangle at X=360, and DPI is 720, then the rectangle is placed 1/2 inch in from the edge of the page.

You will find it easier to always specify units (such as “in” for inches), and avoid this DOT confusion.

## 14.6 Positioning

Items that Print Wizard will print, including text, bitmaps, rectangles, etc., can all be placed precisely where you want them. The horizontal position is always relative to the left edge of the page. The vertical position is always relative to the top edge of the page. Positions can be specified in various units (such as inches) as described above.

Many printers have an unprintable area (or “offset”) on one or more edges of the page. This means that you will not be able to print something to the very top edge, for instance. However, Print Wizard takes into account the unprintable areas to ensure that all positioning is done relative to the edges, so that printout will be consistent between printers.

## 14.7 Positioning Relative to Margins

A positioning parameter can also specify a location relative to the margin. For instance, to have an image positioned one inch from the left margin, do:

```
<IMG X=left+1in ...
```

Each dimension of this kind starts with a margin name (LEFT, RIGHT, TOP, or BOTTOM) followed by a plus or a minus, then an amount, and optionally a unit of measure.



If ROWOFFSET and/or COLUMNOFFSET have been used to establish full-sheet labels, for instance, the margins used will be for the individual label, column, or row. If a TABLE is in effect, the margins used will be for the individual cell within the table.

## 14.8 Global Filtering with Regular Expressions

It is possible to specify one or more search-and-replace operations that will be performed on every line of the data. These operations are specified using *regular expressions*, an approach that is widely used in the computer world.

To use this feature, place one or more REPLACE tags at the beginning of the document, *before* the BODY tag. Replacement operations are applied to all lines *following* the BODY tag, including acting on any tags contained there. It is also possible for a REPLACE action to replace plain text with a tag (with the normal requirement that in LEGACY mode a TRIGGER character must be used).

The syntax is:

```
<REPLACE IN="search string" OUT="replace string">
```

This will search *every* line for the string listed after IN, and replace it with the string listed after OUT.

A more restrictive format is:

```
<REPLACE IN="search string" OUT="replace string" LINE=n>
```

When LINE is specified, then the replacement will be applied only to that line number, where the first line in the page (or label) is 1.

If multiple REPLACE operations apply to the same line, all will be processed in the order they were specified.

The general format of the *search string* and *replace string* follows conventions used elsewhere. A book on regular expressions may be useful. But while the concepts are widely used, details and syntax vary, so they are listed here.

### 14.8.1 The IN string

The characters that can be searched for in the IN string are as follows:

Entries are case sensitive!

<u>Entry</u>	<u>Meaning</u>
.	(period) Any character except Newline
*	Matches previous character 0 or more times
+	Matches previous character 1 or more times
?	Matches previous character 0 or 1 times

<code>^</code>	Beginning of line
<code>\$</code>	End of line
<code>[abc]</code>	Matches “a” or “b” or “c”
<code>[a-c]</code>	Match “a” to “c”
<code>[^a-c]</code>	Matches any character NOT “a” to “c”
<code>\s</code>	space
<code>\t</code>	tab
<code>\b</code>	backspace
<code>\r</code>	return
<code>\l</code>	linefeed
<code>\n</code>	newline (return + linefeed)
<code>\p</code>	(pipe symbol)
<code>\w</code>	word delimiter; matches <code>\t\s!\"&amp;()*+,-./:;&lt;=&gt;?@[\\^{}~</code>
<code>\xHH</code>	character represented by hex value <i>HH</i>
<code>( ... )</code>	Change sequence of operation, or concatenation
<code>{ ... }</code>	Include contents as one replacement field
<code> </code>	Logical OR
<code>\.</code>	.
<code>\*</code>	*
<code>\+</code>	+
<code>\?</code>	?
<code>\^</code>	^
<code>\\$</code>	\$
<code>\x</code>	x

Special codes, including hex codes (`\xHH`) can be used inside brackets `[]` also.

### 14.8.2 The OUT string

The characters that can be in the OUT string are:

<u>Entry</u>	<u>Meaning</u>
<code>\z</code>	Replace with nothing (delete)
<code>\s</code>	space
<code>\t</code>	tab
<code>\b</code>	backspace
<code>\r</code>	return
<code>\l</code>	linefeed
<code>\n</code>	newline (return + linefeed)
<code>\1 to \9</code>	Tagged fields found by search and enclosed in <code>{ }</code>
<code>&amp;</code>	Entire found text

### 14.9 Default Margins and Alignment

Print Wizard by default will begin printing text as far up and to the left as possible (if *nice margins* are **not** turned on). Stated another way, the initial margins are the same as the edges of the printable area

(described above). So when you print with Print Wizard, text should align with the same place it would if you sent that text directly to the printer, with no printer driver involved.

Also, Print Wizard's auto-fit logic will, if possible, print the data at 10 characters per inch on the horizontal and 6 lines per inch on the vertical. Again, this corresponds to common printing standards.

Taken together, this means that if you move from printing certain text directly from DOS or Unix, to printing it with Print Wizard, the alignment should generally be the same. If the printout fit nicely onto a preprinted form before, on the same printer, it should continue to do so.

However, if you move the process from one printer to another printer, you may have to make adjustments. In particular, if you move from printing on a dot matrix printer, which typically can print to the edges of the page, to a laser printer, which typically can NOT, you will need to adjust the margins and spacing.

If *nice margins* is ON, Print Wizard will attempt to provide balanced margins, up to ½ inch.

## 14.10 Setting Page Dimensions and Margins

The <PAGESIZE> and <BODY> tags, taken together, allow you to specify many aspects of the paper selection and page layout, including paper size, orientation, input bin, orientation, duplexing, and margins. Some of these settings will be dependent on your printer's capabilities.

Paper size can be specified as one of certain standard sizes (identified by name or number), by printer-specific size names or numbers, or by actual dimensions ("X = 8.5in Y=11in"). Note that in Windows NT only, there must be a "form" defined to fit arbitrary dimensions. However, on all other versions of Windows, arbitrary sizes can be specified freely.

Orientation can be specified as "portrait", "landscape", or "auto". In "auto" orientation, Print Wizard will print in portrait, unless the width (in columns) or computed character pitch crosses a certain threshold, as described in "Auto-rotation Logic", page 87.

If the printer can print both sides of the paper, that can be controlled with the "DUPLEX" parameter. Note that "VERTICAL" and "HORIZONTAL" refer to binding along the long and the short edge of the paper, respectively, without regard to the printing orientation. However, "LEFT" and "TOP" do take into account the orientation.

The printing margins can be specified in the <BODY> tag. Note that margins are specified relative to the edge of the paper.

Note also that the "BOTTOMMARGIN" is specified relative to the TOP of the paper, and the "RIGHTMARGIN" is specified relative to the LEFT edge of the paper.

The PAGESIZE tag can also specify what BIN the printer will pull paper from (by name or number), the number of COPIES to be printed, whether multiple copies should be COLLATED.

Windows does not provide a device-independent way to specify which **output** bin or tray should be used, so neither does Print Wizard. Nor is there a way to control stapling.

The PAGESIZE tag can also specify the MEDIA; that is, the kind of paper (by name or number). On printers that support this feature, the printer will prompt on its front panel for an operator to insert the specified kind of paper, before the printer will print the job. Or, the printer may decide which bin to draw paper from, based on what it has been told about what kind of paper is in what bin.

### 14.10.1 Paper names, bin names, etc.

When you specify things such as PAPER and BIN, you have several options. You can:

- 1) Use a Windows-standard number, as listed in the reference section.
- 2) Use a Windows-standard name, as listed in the reference section.
- 3) Use a printer-specific number.
- 4) Use a printer-specific name.

Printer-specific names and numbers are defined by the printer drivers. Once you find out what names/numbers are available, you can use those for a particular printer.

Using printer-specific settings makes your PWML files no longer device independent.

To find out what options are available for a particular printer, run Print Wizard with DEBUG and Printer Setup (“/s”). When you are presented with the Printer Setup box, go to the appropriate area, such as “Input bin”. Note the names available – these are what you can use in PWML. Now select the one you want, and print the job. In the DEBUG output, you will find a line that says, for instance, “Paper bin code = 257”. This is the numeric equivalent.

### 14.10.2 Changing pagesize parameters in the middle of a job

It is possible to issue a new PAGESIZE tag after printing has started. In this case, the parameters specified will take place at the beginning of the next page, however that occurs. You might, for instance, specify an initial PAGESIZE with a BIN of “1”, start into the first page, and then issue another PAGESIZE tag specifying a BIN of “2”. This would cause the second page (when it occurs) and all subsequent pages to print from bin 2 of the printer.

Parameters that can be changed in this manner include PAPER, X, Y, COPIES, COLLATE, BIN, MEDIA, ORIENTATION, OVERLAY, and USEOVERLAY.

### 14.10.3 Usage patterns in pagesize tags

For certain parameters in a PAGESIZE tag, you can specify a *pattern* instead of a single value. For instance, if you specify

```
<PAGESIZE BIN=1,2>
```

then Print Wizard will print the first page from bin 1, the second page from bin 2, then go back to bin 1. That is, by default, the pattern will *cycle*. Or, you could enter

```
<PAGESIZE BIN=1,2,continue>
```

In this case, the reserved word “continue” means to stay on the last value, and NOT to cycle. This would print the first page from bin 1 and all later pages from bin 2.

Parameters that can be set in this way include PAPER, BIN, MEDIA, COPIES, COLLATE, OVERLAY, and USEOVERLAY.

### 14.11 Pagination and Linespacing

“Pagination” refers to breaking print jobs into pages. Something has to decide when to advance printing to the next page. Print Wizard helps with this.

When Print Wizard analyzes a print job, as described elsewhere, it will do its best to determine a “linage” value; that is, how many lines constitute a page. This is sometimes referred to as *length* or as *lines per page (LPP)*. Alternatively, you can set this with the LENGTH parameter in the PAGESIZE tag.

Each time Print Wizard encounters a linefeed character (hex 0A), or otherwise needs to wrap to the next line, it will advance its printing position by an amount called LINESPACING. You can specify an initial LINESPACING amount in the PAGESIZE tag. Note that this can be overridden by various other tags.

Obviously, there is an interrelationship between linespacing, length, and printable space. For instance, if you’re printing on letter-size paper on a laser printer with ¼ inch unprintable space at the top and bottom of the page, you won’t get 66 lines per page at 6 lines per inch. You’ll have to change one number or the other.

While it is processing text, laying it out on the page, Print Wizard will cause a page advance in any one of the following circumstances:

- A formfeed character (hex 0C) is encountered.
- Any vertical movement crosses over the BOTTOMMARGIN.
- A <FORMFEED> tag is encountered and processed.
- A <LABELFEED> tag is encountered and processed, and there are no more labels available on the page.
- A line advance happens, and a LENGTH (LPP) is in effect, and that number of lines have already been printed on the page. You can disable this method by setting LENGTH to less than zero.

In multi-column (label) printing, described below, the LENGTH refers to lines per label, not lines per page.

### 14.12 Multi-column (Label) Printing

Print Wizard has the ability to convert plain text input into multiple columns and/or multiple rows, such as for printing on label stock. That means you can adapt a print job that was designed to print on one-across continuous-feed labels on a dot matrix printer into multi-column labels on a laser printer. This is done by specifying all the relevant dimensions of the label.

Specify the `TOPMARGIN` and `BOTTOMMARGIN` so as to avoid printing outside any labels. If necessary, specify the `LEFTMARGIN` so as to position the beginning horizontal print position of the left-most label. You may want to specify the `RIGHTMARGIN`, just to make sure Print Wizard doesn't try to print too many columns.

The measurement that switches on the multi-column feature, is `COLUMNOFFSET`. Specify the `COLUMNOFFSET` as the horizontal distance from one label to the next (left edge to left edge). To avoid printing in the gap between labels (that is, between the columns), you can specify `COLUMNGUTTER`; this is subtracted from the `COLUMNOFFSET` to establish the right edge of the printable space on a label.

You can set the `WRAP` parameter in various tags to control how Print Wizard will behave if the text won't fit horizontally within the label. Especially useful is `WRAP=SQUEEZE`.

To initiate the multi-row feature, specify `ROWOFFSET` as the distance from the top of one label to the top of the next label. To avoid printing in the gap between rows of labels, specify a `ROWGUTTER`; this is subtracted from the `ROWOFFSET` to establish the effective bottom edge of the printable space on the label.

By default, if both `COLUMNOFFSET` and `ROWOFFSET` are specified Print Wizard prints in column order. To alter this set

```
LABELORDER=ROW
```

When `ROWOFFSET` has been specified, Print Wizard treats each row as a small page, with logic similar to that above (see *Pagination*). In this case, `LENGTH` refers to the number of lines per label. You will also need to ensure that the print data coming to Print Wizard is properly formatted in the vertical dimension. The number of lines per label, combined with Print Wizard's `LINESPACING` (computed or specified), must cause printing to fit within the printable space of the label. Most often, you will want a `LINESPACING` of 1/6 inch (which is the default), so if your label stock is 1 inch high (top-of-label to top-of-label) your print data must contain 6 lines per label.

A `<LABELFEED>` tag can be used to advance to the top of the next label.

Following is the beginning of a multicolumn label job:

```
<pwml><pagesize paper=letter orientation=portrait linespacing=12pt>
<body topmargin=.5in bottommargin=10.5in
leftmargin=.5in rightmargin=8in columnoffset=4in><legacy>
Name 1
Address 1a
Address 1b
```

City, state, zip  
USA

Name 2  
...

Often, the dimensions of the job will be placed in a print initialization file. Included with Print Wizard are many sample initialization files. The file “labels.init” is a generic sample. The files whose names start with “Gaylord” are designed for stock labels from Gaylord, a library supply company.

You can experiment with label layout in the Print Preview screen.

## 14.13 Colors

Print Wizard recognizes the following colors for fonts, font backgrounds, and horizontal rules:

#rrggbb	hex RGB value
GRAYxx	gray scale from GRAY00 (black) to GRAY99 (white)
GREYxx	likewise
white	as a text background, Print Wizard treats this as transparent
whiteout	as a text background, Print Wizard treats this as opaque

aliceblue	darkgreen	gray
antiquewhite	darkkhaki	green
aqua	darkmagenta	greenyellow
aquamarine	darkolivegreen	honeydew
azure	darkorange	hotpink
beige	darkorchid	indianred
bisque	darkred	indigo
black	darksalmon	ivory
blanchedalmond	darkseagreen	khaki
blue	darkslateblue	lavender
blueviolet	darkslategray	lavenderblush
brown	darkturquoise	lawngreen
burlywood	darkviolet	lemonchiffon
cadetblue	deeppink	lightblue
chartreuse	deepskyblue	lightcoral
chocolate	dimgray	lightcyan
coral	dodgerblue	lightgoldenrodyellow
cornflowerblue	firebrick	lightgreen
cornsilk	floralwhite	lightgrey
crimson	forestgreen	lightpink
cyan	fuchsia	lightsalmon
darkblue	gainsboro	lightseagreen
darkcyan	ghostwhite	lightskyblue
darkgoldenrod	gold	lightslategray
darkgray	goldenrod	lightsteelblue

lightyellow	olive	seagreen
lime	olivedrab	seashell
limegreen	orange	sienna
linen	orangered	silver
magenta	orchid	skyblue
maroon	palegoldenrod	slateblue
mediumaquamarine	palegreen	slategray
mediumblue	paleturquoise	snow
mediumorchid	palevioletred	springgreen
mediumpurple	papayawhip	steelblue
mediumseagreen	peachpuff	tan
mediumslateblue	peru	teal
mediumspringgreen	pink	thistle
mediumturquoise	plum	tomato
mediumvioletred	powderblue	turquoise
midnightblue	purple	violet
mintcream	red	wheat
mistyrose	rosybrown	whitesmoke
moccasin	royalblue	yellow
navajowhite	saddlebrown	yellowgreen
navy	salmon	
oldlace	sandybrown	

## 14.14 Fonts

Print Wizard was originally designed to work with mono-spaced fonts, such as Courier New. Mono-spaced fonts are consistent with the focus on legacy print data, which generally relies on character spacing to align columns. Also, Courier New can contain many international characters, if those options have been installed on your Windows system.

Print Wizard can also print using variable-pitch fonts, such as Arial or Times New Roman. However, if you print a job that was designed to use fixed-space fonts to achieve column layout, and you cause Print Wizard to use a variable-pitch font, your column alignment will be lost.

Print Wizard works best with scalable fonts, i.e., TrueType or OpenType fonts. These fonts can be scaled precisely to get the necessary height. If fixed-pitch, they can be sized independently in the horizontal dimension to get the correct printing width.

It is possible, with the FONT tag, to specify ANY available font, but your results may not be acceptable. If you are using a dot matrix printer, you may want to specify a font that is built in to the printer, for faster printing. These fonts can be identified by doing a font selection in a Windows program, and looking for a printer icon beside the font. Again, though, if they're not scalable then that limits Print Wizard's flexibility.



If you specify a font that is built into the printer, such as “Courier 10cpi” in an Okidata printer, and then Print Wizard determines that it needs to print at a tighter pitch (more characters per inch), Print Wizard will attempt to find a similarly named font that will work, such as “Courier 17cpi”. The *debug* output will show you that this has happened.

Text can also be printed in any rotation, with <FONT ... ROTATION=*n*> where “*n*” is in degrees, counterclockwise from horizontal. When ROTATION is not zero, Print Wizard’s cursor advancement is hard to predict, so you will want to specify location afterward with a <GOTO> tag.

Some printer drivers may not handle international characters well. If you have trouble, open the printer’s properties in Windows’ control panel, and try different settings for font substitution and type of download. You might also try a different but similar printer driver, such as a LaserJet 4 driver for a LaserJet 5 printer.

### **14.15 Automatic Font Selection**

Print Wizard has sophisticated logic to ensure that the characters you want to print will in fact get printed. Part of this logic includes testing the font to make sure that it contains the character. If it does not, Print Wizard will attempt to find and use one that does.

Print Wizard can be told to print non-ASCII text by a) using an HTML ampersand character entity (see below), or b) feeding it characters whose numeric value is above hex 7F (decimal 127). In the latter case, with the freestanding Print Wizard program, those characters are assumed to be in the OEM (DOS) character set, unless a command-line parameter specifies them to be in the ISO (Windows) set (“/i”) or in Unicode UTF-8 (“/u”). Within AnzioWin, the character set of passthrough print data can be specified in the Communicate:Character set menu item.

Print Wizard works with a “font list”, a list of font names separated by commas. The first font in the list is the primary font; it is used if possible. The second and following fonts are alternate fonts. If a needed character is not in the primary font, Print Wizard will search through the second and following fonts until it finds one that contains the character. If the last font in the font list is “auto”, Print Wizard will check Windows for all installed fonts (generally with a preference for mono-spaced fonts) to find one with the necessary character.

On startup, Print Wizard has a font list of “Courier New, auto”. That is, Courier New will be used wherever possible; when it doesn’t contain a character Print Wizard will automatically search for an appropriate font.

You can specify a different font list in two ways: a) through the command line that invokes Print Wizard, or b) using Print Wizard Markup Language (PWML).

For the command line, use a parameter:

```
/vFont="name1, name2, name3"
```

You can list as many names as you like. Names can contain embedded spaces. Names must be separated by commas. Use a name “auto” as the LAST item in the list. For instance:

```
printwiz /vFont="Courier new,ms song,auto" somefile.txt
```

Within PWML (or HTML), use the syntax:

```
<font face="name1, name2, name3">
```

You can also specify one or more fonts that should NOT be used during “auto” font selection, by including those font names with a minus sign in front. For instance:

```
<font face="Courier new, Mingliu, -badfont, auto">
```

Note that if you specify the font list by either means, it will override the default startup values, including secondary font(s), including "auto".

## **14.16 Font Size, Pitch, and Line Spacing**

LINESPACING is the vertical distance from one line of text to another.

FONT SIZE is the height of a capital letter.

CELL SIZE is the maximum height of the cell in which any letter could fit. This includes descenders (such as the tail on a “y”) and diacritics above a character (such as on “Ä”). The cell size is larger than the font size.

PITCH is the width of a character.

Your PWML code can specify LINESPACING directly. If you do not specify LINESPACING, but you specify a FONT size (or POINTSIZE or CELLSIZE), LINESPACING will be the same as the FONT size. If you specify neither, Print Wizard calculates the LINESPACING necessary to fit the required lines into the page, and then bases the font size on that.

The standard line spacing, at least in the US, is 6 lines per inch. Because there are 72 points in an inch, a font size of “12pt” (12 point) would yield 6 lines per inch.

You can change the dimensions of the text with a <FONT> tag.

If the font size is changed within the body of a print job, with a <FONT> tag, the line spacing will be changed accordingly, for all printing from then on, until a font end-tag (</FONT>).

LINESPACING affects every linefeed; that is the end of every line in your input file. We recommend changing FONT (and therefore line spacing) **before** an end-of-line.

In a similar manner, the spacing in the horizontal direction is related to the width of the characters. Both are controlled by the PITCH parameter within the FONT tag. It is quite common to print at 10 characters per inch; that is, one character is 1/10 inch wide. This would be specified:

```
<font pitch=.1in>
```

For 132-column reports in 8 inches, it is common to print at 17 CPI. Because  $1/17 = .0588$ , this would be specified:

```
<font pitch=.0588in>
```

Again, these settings will affect both the spacing between characters and the printed widths of the characters themselves (at least for TrueType fonts).

For variable-pitch fonts, such as Arial or Times New Roman, specify a PITCH of 0 (zero).

### 14.17 Horizontal Alignment

When Print Wizard places text on the page, it is always working with certain margins (right and left), which may have been set by a <BODY> tag. Other tags, such as <DIV>, can change the left and right margin on a temporary basis, ending when the corresponding end-tag is encountered (</DIV>). Inside each cell of a <TABLE>, there are implicitly set left and right margins. And in multi-column output (labels), there are implicitly set left and right margins.

Text is normally aligned to the left; that is, it starts printing at the LEFTMARGIN. However, it is possible to specify different ALIGN values.

Possible values are LEFT, RIGHT, CENTER, and FILL. Meanings should be obvious.

A special case is the <SPAN> tag. This tag specifies a span of text *within a line* that can be printed with a certain WIDTH and a certain ALIGN.

See the reference section for places where ALIGN can be specified.

### 14.18 Word Wrap

When Print Wizard can't fit its text between the working LEFTMARGIN and RIGHTMARGIN (see above), what should it do? By default, it will put as many characters as possible on the line, and then

advance to the next line. That is, it will break on a character boundary. However, it is possible to specify different WRAP values.

Possible values are:

NONE	Text is allowed to slop over the margins
CHAR	Line is broken at the character (default)
WORD	Line is broken on a word boundary (a space)
CHOP	Characters that don't fit are discarded
SQUEEZE	Font size is reduced until the text fits

The SQUEEZE option is especially useful in fitting variable amounts of text with a specified space on one line.

## 14.19 Character Entities

In <PRE> mode, Print Wizard recognizes HTML standard named values for character entities, such as "&lt;" for less-than ("`<`"), "&amp;" for ampersand ("`&`"), and also "&euro;" for the Euro character ("`€`" -- might not show on your PC).

It also recognizes characters coded numerically, such as "&#242;" and "&#D242;" (both decimal), and "&#H20AC;" and "&#X20AC;" (both hex). Numbers below 128 (decimal) resolve to ASCII values. Numbers between 128 and 255 are in the ISO 8859-1 character set, which is essentially the same as Windows Latin-1. As in true HTML, numbers above 255 resolve to 16-bit Unicode characters.

Characters specified in this way are not affected by the character-set setting (/I, /O, /U) of Print Wizard.

To look up Unicode values, you can browse through the tables at <http://www.unicode.org>. Or, on later versions of Windows, you can use the Charmap utility. Do Start:Run:Charmap. Set the font to Courier New, for instance. Now browse through the available characters. As you move your mouse over a character, a hint will appear showing something like "U+221E: Infinity" for the infinity symbol. The "221E" is the hex Unicode value.

Following is a complete list of the named character entities that Print Wizard recognizes. Note that names are case sensitive. Some characters may not show below, due to font limitations

quote	"	nbsp	non- break space	pound	£	uml	¨
amp	&			curren	¤	copy	©
lt	<			yen	¥	ordf	ª
gt	>	iexcl	¡	brvbar	¸	laquo	«
euro, Euro	€	cent	¢	sect	§	not	¬

shy	-	Ucirc	Û	Iota	I	hellip	...
reg	®	Uuml	Ü	Kappa	K	prime	'
macr	-	Yacute	Ý	Lambda	Λ	Prime	"
deg	°	THORN	Þ	Mu	M	oline	-
plusmn	±	szlig	ß	Nu	N	frasl	/
sup2	<sup>2</sup>	agrave	à	Xi	Ξ	weierp	∅
sup3	<sup>3</sup>	aacute	á	Omicron	O	image	ℑ
acute	´	acirc	â	Pi	Π	real	℔
micro	μ	atilde	ã	Rho	Ρ	trade	™
para	¶	auml	ä	Sigma	Σ	alefsym	ℵ
middot	·	aring	å	Tau	T	larr	←
cedil	¸	aelig	æ	Upsilon	Υ	uarr	↑
sup1	<sup>1</sup>	ccedil	ç	Phi	Φ	rarr	→
ordm	°	egrave	è	Chi	Χ	darr	↓
raquo	»	eacute	é	Psi	Ψ	harr	↔
frac14	¼	ecirc	ê	Omega	Ω	crarr	↵
frac12	½	euml	ë	alpha	α	lArr	⇐
frac34	¾	igrave	ì	beta	β	uArr	⇑
iquest	¿	iacute	í	gamma	γ	rArr	⇒
Agrave	À	acirc	î	delta	δ	dArr	⇓
Aacute	Á	iuml	ï	epsilon	ε	hArr	⇔
Acirc	Â	eth	ð	zeta	ζ	forall	∀
Atilde	Ã	ntilde	ñ	eta	η	part	∂
Auml	Ä	ograve	ò	theta	θ	exist	∃
Aring	Å	oacute	ó	iota	ι	empty	∅
Aelig	Æ	ocirc	ô	kappa	κ	nabla	∇
Ccedil	Ç	otilde	õ	lambda	λ	isin	∈
Egrave	È	ouml	ö	mu	μ	notin	∉
Eacute	É	divide	÷	nu	ν	ni	∋
Ecirc	Ê	oslash	ø	xi	ξ	prod	∏
Euml	Ë	ugrave	ù	omicron	ο	sum	∑
Igrave	Ì	uacute	ú	pi	π	minus	-
Iacute	Í	ucirc	û	rho	ρ	lowast	*
Icirc	Î	uuml	ü	sigmaf	ς	radic	√
Iuml	Ï	yacute	ý	sigma	σ	prop	∝
ETH	Ð	thorn	þ	tau	τ	infin	∞
Ntilde	Ñ	yuml	ÿ	upsilon	υ	ang	∠
Ograve	Ò	fnof	ƒ	phi	φ	and	∧
Oacute	Ó	Alpha	Α	chi	χ	or	∨
Ocirc	Ô	Beta	Β	psi	ψ		
Otilde	Õ	Gamma	Γ	omega	ω		
Ouml	Ö	Delta	Δ	thetasym	ϑ		
times	×	Epsilon	Ε	upsih	Υ		
Oslash	Ø	Zeta	Ζ	piv	ϖ		
Ugrave	Ù	Eta	Η	bull	•		
Uacute	Ú	Theta	Θ				

cap	∩	Yuml	ÿ
cup	∪	circ	ˆ
int	∫	tilde	˜
there4	∴	ensp	en space
sim	≈	emsp	em
cong	≅	thinsp	thin
asymp	≈		space
ne	≠	zwnj	zero-
equiv	≡		width
le	≤		non-
ge	≥	zwj	joiner
sub	⊂		zero-
sup	⊃		width
nsub	⊄	lrm	joiner
sube	⊆		left-to-
supe	⊇	rlm	right
oplus	⊕		mark
otimes	⊗	ndash	–
perp	⊥	mdash	—
sdot	⋅	lsquo	‘
lceil	⌈	rsquo	’
rceil	⌋	sbquo	‚
lfloor	⌊	ldquo	“
rfloor	⌋	rdquo	”
lang	⟨	bdquo	„
rang	⟩	dagger	†
loz	◇	Dagger	‡
spades	♠	permil	‰
clubs	♣	lsaquo	‹
hearts	♥	rsaquo	›
diams	♦		
quot	"		
amp	&		
lt	<		
gt	>		
OElig	Œ		
oelig	œ		
Scaron	Š		
scaron	š		

## 14.20 Variable Width Spaces

In certain situations, especially when printing in a variable-spaced font, it is convenient to be able to use spaces of different widths. Print Wizard recognizes several space characters as standardized in Unicode. These can be specified by their numeric values, as described above, or as encoded in UTF-8 in the print file. Note that many sizes are based on an "em" unit, which is the same width as the current point size (height).

<u>Character value (hex)</u>	<u>Width</u>
0020	standard
2000	½ em height
2001	em height
2002	½ em height
2003	em height
2004	1/3 em
2005	1/4 em
2006	1/6 em
2007	same width as zero
2008	same width as decimal point
2009	1/5 em
200A	1/12 em
200B	zero width space

## 14.21 Line-drawing Characters

Line-drawing characters are commonly used in DOS and UNIX programs to create boxes. Print Wizard can be told to print these in three ways: a) as 8-bit characters in an OEM encoding, b) as character sequences in a UTF-8 encoding, or c) by their Unicode values (in the hex 2500 range).

Print Wizard does NOT rely on fonts to print these characters, because fonts are inconsistent in placing and connecting the lines. Instead, Print Wizard will draw the lines using internal bitmaps. Thus they should always print properly.

Print Wizard uses the following rules to determine the dimensions at which it prints line-drawing characters:

The height is the height of the font being used or the LINESPACING value, whichever is greater.

If a fixed-pitch font is being used, the width is the pitch width; otherwise, the width is the same as the current font height.

## 14.22 <Arctext> for text on an arc

You may need to align text along the arc of a circle, such as to create a label for a CD or DVD. The <ARCTEXT> tag allows you to do this.

Two points and a direction define the circle along which the text will be aligned. The first point, identified by the X and Y parameters, is on the circle. The default is the current printing position. An ALIGN parameter, with a value of LEFT, RIGHT, or CENTER, tells which part of the text goes at this point.

The text to be printed is given in the SRC parameter.

A second point, given by X2 and Y2, is the center of the circle. Default is one inch below the (X, Y) point.

Finally, the DIR parameter sets the circle direction as CW (clockwise) or (CCW) (counterclockwise).

### **14.23 Rectangles**

HTML has a feature called “horizontal rule”, as <HR>. By specifying SIZE (height), WIDTH, COLOR, and OUTLINE, you can create a solid, hollow, or color-filler rectangle of any size. The OUTLINE is always black.

Print Wizard extends this with absolute positioning (X and Y), so you can place the HR any place on the page. With clever use of these, you can create grids and other forms.

Placing an HR will change the current printing position, so you may want to use a GOTO tag afterward to begin printing text in a particular spot.

It is possible to print rectangles over each other, text over rectangles, etc.

### **14.24 Lines**

You can also tell Print Wizard to draw arbitrary lines, from one x-y location to another (including lines at any angle). The WIDTH and COLOR parameters will dictate how the line is drawn. Defaults are one pixel wide, in black.

The endpoints of the line are identified as X and Y for one end, and X2 and Y2 for the other end. The default for both ends is the current printing position; if you don't specify *any* you will not get a visible line.

### **14.25 Images**

Print Wizard is able to print images (bitmaps, pictures) from files that are in BMP, GIF, TIF, or JPG/JPEG format, as well as Windows metafiles (WMF) or enhanced metafiles (EMF). Files can be on the PC, a local area network, or a web or FTP server. Images are coded using the <IMG> tag as in standard HTML. These can be sized with the HEIGHT and WIDTH attributes. A BORDER can be added (always in black).



If the filename (in the SRC parameter) does not contain a complete path, Print Wizard will look for it in a particular directory, generally the directory containing the data file. You can override this by using the BASE tag with the HREF parameter. You can also specify an assumed file extension, with the EXT parameter of the BASE tag. So, if early in the print job you specify:

```
<base href=c:\mypix ext=".bmp">
```

then you can generate a picture with the tag

```
<img src=wally>
```

and Print Wizard will read the file "c:\mypix\wally.bmp".

With Print Wizard's additional attributes (X and Y), graphics can be placed anywhere on the page.

Printing a bitmap will change the current printing position, so you may want to use a <GOTO> tag afterward to begin printing text in a particular spot.

Print Wizard places items on the page in the order they are in the file. It is possible to print text over graphics. However, graphics over text may hide the text.

PNG and GIF files can have a transparent background. Print Wizard handles the transparency when printing them.

## 14.26 Barcodes

Print Wizard can internally generate a variety of barcodes, which can then be printed on nearly any printer, with a tag as simple as:

```
<barcode src=1234>
```

Of course there are many kinds of barcode, and you'll want to specify style, size, position, etc. The Reference section below lists all the parameters. But here are some special notes.

The SRC is the actual data to be barcoded.

The kind of barcode is set with the STYLE parameter; default is CODE128.

The SIZE of the barcode is its height. If not specified, it will default to the current font height.

Print Wizard starts off assuming that BARWIDTH, the width of the narrowest bar or space, is .012 inches. However, this width will be rounded up, based on your printer's graphics resolution, so the actual width of the printed barcode will vary some between printers. It is advisable to set your printer's properties (through Windows) at its maximum resolution. On a low-resolution printer, the default BARWIDTH may produce unreadable barcodes. Therefore, you may need to specify a larger value.

Some barcode styles have particular requirements, such as numeric data only or a particular length. If you don't meet these requirements, Print Wizard may detect that and print a message on the page, in place of the barcode.

Some barcode styles can have a check character, which is used by the barcode reader to mathematically validate the barcode. You can tell Print Wizard to calculate and print this character with the ADDCHECKCHAR parameter. Without this parameter, you must supply the check character as part of the SRC.

Barcodes usually have one line of human-readable text also. This can be turned on or off with the CAPTION parameter. The font used for the caption can be specified with the FACE parameter; default is the same face being used for text.

Certain styles of barcode have optional features, such as bearer bars or tall guard bars. These can be turned on or off.

Bar codes can be rotated to 90, 180, or 270 degrees, with the ROTATION parameter.

Barcodes are printed in line with the text, unless placement is specified with X and/or Y. The horizontal (if not rotated) size of the barcode is calculated automatically from the BARWIDTH, the STYLE, and all the necessary elements of the barcode. The current printing position will advance based on the total width. However, if the barcode is rotated, final printing position is undetermined.

Print Wizard does not enforce the need for white space around a barcode. If you have other printing too close, the barcode will not be scanned reliably. Be sure to test your results.

Once the format of the barcode (STYLE, SIZE, etc.) has been established, it will remain in effect for the duration of the print job. It is sometimes easier to establish this format once, at the beginning of the print job, by including a BARCODE tag that has no SRC parameter (and will therefore not print anything).

Print Wizard still retains an earlier approach to printing barcodes, using special barcode font files. If you need to use these, please contact Rasmussen Software.

### **14.27 Tables**

Print Wizard will print many kinds of HTML tables.

### **14.28 Passthru**

Print Wizard allows low-level text to be passed through to the printer.

### **14.29 Tail – Data to be added at end of job**

The <TAIL> tag allows you to specify data to be appended to the end of a job.

### **14.30 Special tags for PDF generation**

#### **14.30.1 <Attach>**

#### **14.30.2 <Index>**

#### **14.30.3 <A> for Anchor**

#### **14.30.4 <Signature>**

## 15 Init. Files

### 15.1 Using initialization files

Regardless of how Print Wizard is invoked, it is possible to specify a print initialization file. This file should be a plain text file, created with an editor such as NOTEPAD or EDIT. Whatever is in the initialization file will be inserted onto the beginning of the data file. The initialization file can contain escape sequences, or, more commonly, PWML initialization tags. It can also contain actual text and/or graphics to be printed at the beginning of the print job.

The file can be located on a local or networked disk drive, or it can be on an HTTP, HTTPS, or FTP server.

If the file contains escape sequences, that will force Print Wizard to switch to spooler mode, so that the initialization file and then the data file will be sent at a very low level to the printer, bypassing the Windows printer driver. The escape code sequences must be appropriate for the kind of printer involved. This is a less-common use of printer initialization files. (It would also be possible to do PCL translation of these codes.)

More often, the file will contain PWML tags. These can initialize the Print Wizard process, establishing job-level parameters such as paper selection, orientation, font, overlay, etc. The initialization file will generally leave the print process in a mode such as PRE or LEGACY.

For instance, the following initialization file will cause reports to print in landscape mode, with an assumed length of 60 lines per page (if there are no form feeds), with a top margin of  $\frac{3}{4}$  inches:

```
<PWML><Pagesize orientation=landscape length=60>  
<BODY topmargin=.75in><legacy>
```

The following file would cause reports to print with a bitmap form overlay, and text margins set to 1 inch at top and left:

```
<PWML><pagesize overlay=myform.gif><body topmargin=1in leftmargin=1in><pre>
```

Note that line breaks before <legacy>, <pre>, <listing>, or <xmp> are ignored (both in the initialization file and in general usage). Furthermore, a single line break at the end of the initialization file is ignored.

The SAMPLES directory contains some sample print-init files. The file "LETTER.INIT" initializes Print Wizard to letter-size paper (8.5" by 11"), auto orientation, with convenient margins. The file "LABELS.INIT" initializes Print Wizard for 2-column labels. Files starting with "Gaylord" are for laying out labels on standard labels as sold by Gaylord, a library supply company. You can use these files as starting points for your own experimentation.

## 16 Overlays

A powerful and useful feature of Print Wizard is its ability to print forms as overlays. An overlay is any printing that is placed on every page of output, in combination with the text (or other content) that is coming from the main file.

The term “overlay” is actually a misnomer. It should be “underlay”, because it is generally drawn first on the page, with the variable text over it.

When overlays are printed as part of the output, it creates a way to print on plain paper what would otherwise require a preprinted form. This can be a significant cost savings.

The classic illustration of an overlay is in printing invoices. You have a file of text that in the past has been printed on preprinted invoice forms. If you can create an electronic image of that invoice form (as described below), Print Wizard can print the invoice *plus* the text on each page. Another example would be a *watermark*; that is, an image or large text that prints lightly in the background of each page.

Historically, an overlay has consisted of one image, which was printed on every page of output. However, Print Wizard also supports *multi-page overlays*, in which the overlay images are cycled as text is printed.

Note also that Print Wizard can apply overlays to non-printed output, namely faxes and PDFs.

Finally, note that an overlay is always considered to be a full-page image. There is no mechanism for specifying the *placement* of text on the page.

An OVERLAY is always contained in a separate file. This file, like the main print file, can be on a local disk, a network drive, or an HTTP, HTTPS, or FTP server. You can specify that a particular overlay file should be used a) externally, such as in the command line for PRINTWIZ.EXE; b) in a print-initialization file; or c) in the PWML header of the main print file.

A special case of overlays is the PREVIEWOVERLAY. This is an overlay that is shown only in the Print Preview window. It is not actually printed. This can be useful as an image of the preprinted form (or label stock) that the data will be printed on. You can use the adjustment capabilities of the Print Preview window to make the data line up correctly on the overlay.

The way that Print Wizard handles an overlay file depends on its file extension. Make sure that's right.

Following are descriptions of the various kinds of overlays Print Wizard can support.

### 16.1 Bitmap Overlays

A bitmap overlay is simply a file that contains a picture that you want in the background. Some notes:

- Works on any non-generic printer.
- Supports BMP, GIF, TIF, JPG, JPEG.
- Image is stretched to the full size of the paper, including covering unprintable areas. If you are prescanning a form, scan it edge-to-edge so it will print at the same size.
- One-page overlay only.

## 16.2 Scanned Overlays

In this case, you scan something to use as an overlay at the start of the print job. You specify this by using the reserved filename “SCAN://”. Some notes:

- Works on any non-generic printer, fax, PDF.
- Supports any TWAIN-compliant scanner.
- Image is stretched to the full size of the paper, including covering unprintable areas. Scan it edge-to-edge so it will print at the same size.
- For best results, scan at the same dot density as the printer.
- One-page overlay only.

## 16.3 EMF and WMF Overlays

Any EMF (Enhanced Meta File) or WMF (Windows Meta File) can be used as an overlay. These files can be created by many graphics programs.

- Works on any non-generic printer, fax, PDF.
- Image is stretched to the full size of the paper, including covering unprintable areas.
- File extension “EMF” or “WMF”.
- One-page overlay only.

## 16.4 SPL Overlays

An SPL (spool) file is even more enhanced than an Enhanced Meta File (EMF), because it can contain multiple pages. SPL files are created by capturing print output from other programs, as described elsewhere. Notes:

- Works on any non-generic printer, fax, PDF.
- Image is stretched to the full size of the paper, including covering unprintable areas.
- File extension “SPL”.
- Can be multi-page. Pages are cycled as data is printed.

## 16.5 PCL Overlays

As previously described, a PCL file can be used as an overlay. Described elsewhere is how to create a PCL file using a Windows printer driver. Notes:

- With TranslatePCL on, works on any non-generic printer, fax, PDF.
- With TranslatePCL off, requires a PCL-5 printer and driver. This is more efficient.
- File extension “PCL”.
- Can be multi-page. Pages are cycled as data is printed.

## 16.6 PWML Overlays

You can use a file containing PWML markup as an overlay. This would be appropriate if you wanted to *describe* the overlay using PWML tags, such as <LINE> and <HR>. The entire overlay file is processed first, and saved as one or more pages in memory (as EMFs). These are then added to the data from the main file. There is no interaction between the tags in the overlay and the tags in the main file. Notes:

- Works on any non-generic printer, fax, PDF.
- File extension “PWML”.
- Can be multi-page. Pages are cycled as data is printed.

## 16.7 PDF Overlays

Print Wizard can read standard PDF files and use them as overlays.

## 16.8 Comparison of Methods

The primary tradeoffs between methods are file size, speed, and portability.

A full-page bitmap is a large file, and it must be sent to the printer for each page of output. It is inherently inefficient.

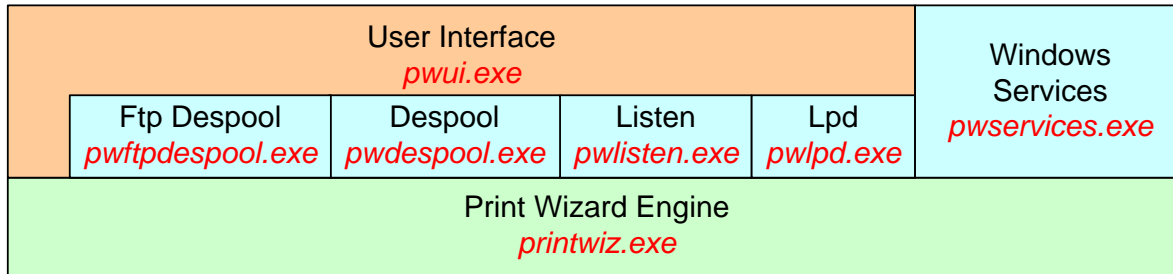
A WMF, EMF, or SPL file *can* be small, depending on how it is produced. It must be sent to the printer for each page. Also, although these metafiles are designed to be portable between versions of Windows and types of printers, there could be problems. Fonts specified in a metafile might not be present on a different PC.

PCL files can be quite small and efficient. Later drivers (LaserJet 5 as compared to LaserJet 3, for instance), tend to be smaller. They are very portable.

PWML files are small, efficient, and extremely portable. However, they must be created by hand.

# Individual Program Documentation

Print Wizard Personal Edition contains several programs for processing your print job. These same programs are included in the Print Wizard Service Edition with the addition of several programs for handling “services”.



Print Wizard  
 Embedded

Print Wizard  
 DLL

WePO

MapiSend &  
 Others



## **17 The UI - pwui.exe**

Refer to the Print Wizard User Interface in the first section of this manual for more information on running this program and setting up the Print Wizard options.



## 18 The Print Wizard Engine – printwiz.exe

Print Wizard can be started from a number of sources:

- From the User Interface.
- From the various services (PwDespool, PwListen, PwLpd, PwServices).
- From a Windows script or batch file.
- From a third-party program or script.
- From a desktop shortcut.
- From a Windows Command (Cmd) prompt or from the Start : Run dialog.

In all these cases, the command-line parameters are used to describe any changes from the ordinary when printing a file. Please refer to the section above on *Command-line Parameters* for more information.

Note that not all programs handle all command-line parameters, only those that are logical for the operation the program is to perform.



## 19 Despool Service – pwdespool.exe

PwDespool is a program that starts a Print Wizard Despool service. This program can be started with command-line parameters that include the directory to watch. However, in order to take advantage of higher level procedures, the command-line will need to specify an already defined service.

To start a standard Print Wizard Despool, do:

```
Pwdespool.exe <directory name or wildcard>
```

To start despool with a specific service, do:

```
Pwdespool.exe /service=<servicename>
```

Despool services can watch for and process any file that shows up in a directory, or it can watch for files by a wildcard, or by a specific file name.

```
Pwdespool c:\despool\  
Pwdespool c:\despool\*.txt  
Pwdespool c:\despool\mydespoolfile.txt
```

You can even include multiple directories, wildcards or file names in the command line for PwDespool to watch for:

```
Pwdespool c:\despool\ c:\temp\pw\*.pwml c:\temp\pw\test*  
c:\temp\mydespool\myfiletowatchfor.txt
```

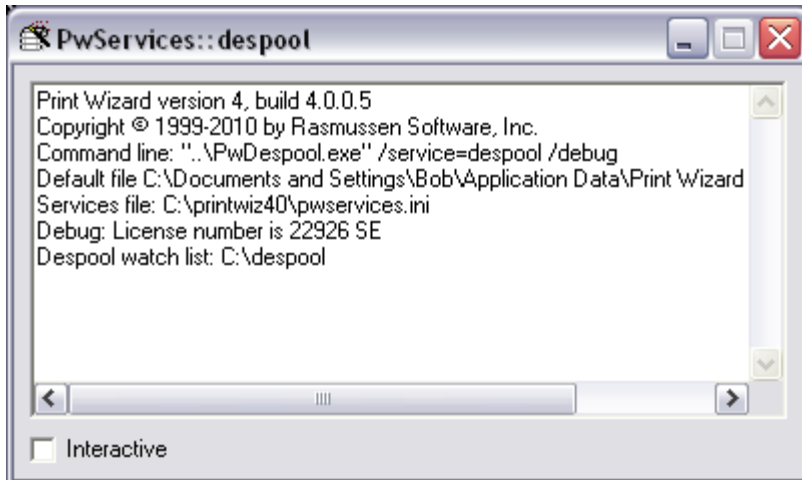
The above is similar to doing the following from the Print Wizard engine command line parameters.

```
Printwiz.exe /despool c:\temp\pw\*.pwml c:\temp\pw\test*  
c:\temp\mydespool\myfiletowatchfor.txt
```

If you run with a service record setting (set your list in the Service Record directory settings), you can also do a combination of /service=<servicename> and <wildcard>:

```
Pwdespool /service=MyDespool c:\despool\ c:\temp\pw\*.pwml
```

PwDespool starts minimized by default. It does have an active messages screen that display information as jobs are processed.



PwDespool watches the directory for all files that may end up there. It copies those files to Windows temporary work space, removes them from the watched directory and hands the printing of the file over to the Print Wizard engine. It then returns to watching for more files to process.

The temporary workspace, a print profile to use, the printer to use, etc. can all be controlled by a service record entry in the Print Wizard services file.



## 20 FTP Despool Service – pwftpdespool.exe

PwFtpDespool is a program that starts a Print Wizard FTP Despool service. This program requires you specify a service in order to run it. PwFtpDespool will watch a remote directory on an FTP server for files to print, utilizing the FTP protocol.

To start a standard Print Wizard FTP Despool, do:

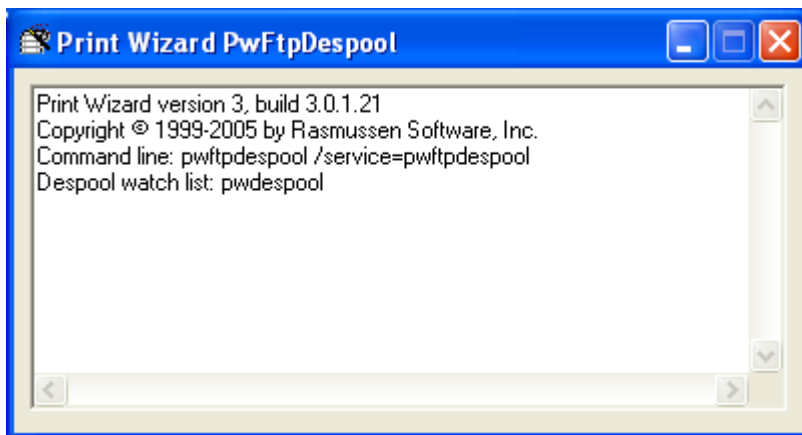
```
Pwftpdespool.exe /service=<servicename>
```

FTP Despool services can watch for and process any file that shows up in a directory located on some remote host. That host may or may not require a login and may or may not have full permissions.

You must set up a service record through the User Interface in order for PwFtpDespool to work properly (it needs to know how to access your FTP server).

At present no wildcard file names are watched for, all files are processed. However, through rules you can include or exclude specific file name extensions.

PwFtpDespool starts minimized by default. It does have an active messages screen that display information as jobs are processed.



PwFtpDespool watches the remote directory for all files that may end up there, polling the FTP server every so many seconds (you can set this time interval when setting up the service (default is once a minute). It then copies any files it needs to print to a Windows temporary workspace, removes them from the watched directory (if it has permission to do so) and hands the printing of the file over to the Print Wizard engine. It also will index these files in order to prevent them from printing again. It then returns to watching for more files to process.



The FTP host name and login, the poll time interval, the temporary work space, a print profile to use, the printer to use, etc. can all be controlled by a service record entry in the Print Wizard services file.

The index file can be cleared from the Print Wizard user interface; otherwise it contains the last 1000 print job names and their date/time stamp to make sure they do not print again. The “printed” index file name is:

`<printwiz home dir>\<servicename>\.idx`

and can be manually changed within notepad if a file should need reprinting and it still exists at the host (simply delete the line pertaining to the file to print).



## 21 Listen Service – pwlisten.exe

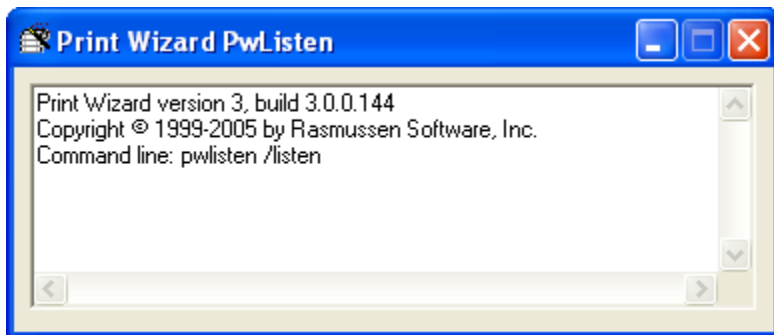
PwListen is a program that starts a Print Wizard Listen service. This program can be started with command-line parameters that include the network port to listen on. As with PwDespool, in order to take advantage of higher-level procedures, the command-line will need to specify an already defined service.

To start a standard Print Wizard Listen, with the default port of 9100, do:  
Pwlisten.exe /listen

To start a Print Wizard Listen on a specific port, such as 9101, do:  
Pwlisten.exe /listen /port=9101

To start listening with a specific service, do:  
Pwlisten.exe /service=<servicename>

PwListen starts minimized by default. It does have an active messages screen that display information as jobs are processed.





## 22 LPD Service – pwlpd.exe

A standard for UNIX/Linux/AIX remote printing is supported through a number of protocols often grouped together as the Remote Print protocol, often referred to as Remote LP, LPR or LPR/LPD (LPD being the server portion and LPR the client portion).

Along with this there are several supporting and extended versions of the protocol, such as LPRng, CUPS, etc.

These protocols are designed to watch for print jobs at the server end that are issued at the client end. Along with issuing regular print jobs to process, the protocol also lends itself to some basic print management options, such as requesting printer status, job status and killing pending print jobs.

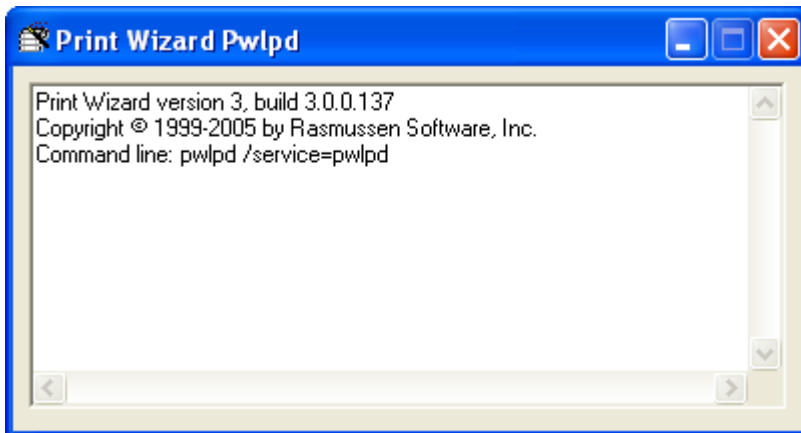
PwLPD is one of these servers. When running it listens on a specific port (the default is port 515) and waits for requests from a client.

Unlike PwDespool and PwListen, PwLpd cannot be run from command line parameters only as it needs to have defined certain information for the LPD protocol to work properly. In particular, it must have defined a “print queue” in order for the program to process requests from an LPR client.

To start PwLpd listening with a specific service, do:

```
Pwlpd.exe /service=<servicename>
```

PwDespool starts minimized by default. It does have an active messages screen that display information as jobs are processed.





## 23 Windows Services – pwservices.exe

Print Wizard Windows services are most often installed through the user interface for Print Wizard. However, it is also possible to install, start, stop and uninstall Print Wizard Windows services direct or through a script. In addition, you can manage the installed Windows services through your normal Windows management tools, such as Administrator Services.

### 23.1 Managing Windows Services Manually

Print Wizard Windows services (PwServices) are "started" by Windows, based on the ImagePath key inside the registry. This means a Windows service must be installed before it can be run. Likewise any parameters and service record information must be included in the same location.

When you install PwServices, it installs as the service name you specified on install. PwServices must already have an established service record found in the "pwservices.ini" file (created through the Print Wizard User Interface). Also a startup type of "auto" is assumed. This means that it will automatically start the next time Windows restarts.

You can do the install manually by running the program from the Start: Run Windows dialog, from a Windows script, or from the Windows command (CMD) prompt:

```
pwservices.exe <servicename> /install
```

where <servicename> is the predefined Print Wizard service. Likewise, you can uninstall the Windows service by doing:

```
pwservices.exe <servicename> /uninstall
```

You can also do this with the /service=<servicename> switch, i.e.

```
pwservices.exe /service=<servicename> /install  
pwservices.exe /service=<servicename> /uninstall
```

Likewise, there is a way to start and stop the installed service either from the Windows Start:Run dialog, from a Windows script, or from the Windows command (CMD) prompt:

```
sc /start <servicename>  
sc /stop <servicename>
```

You can also query to see if a service is running from the command-line:

```
sc /query <servicename>
```

or by running the PwServices program with a switch to query ALL Print Wizard services

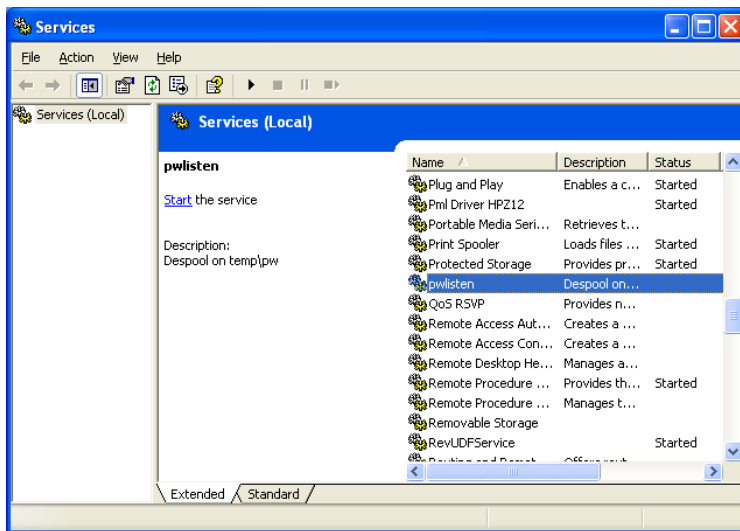
```
pwservices.exe /query
```

Sometimes, especially in scripts, it is necessary suppress any displayed messages by running pwservices silently:

```
pwservices.exe <servicename> /install /silent
pwservices.exe <servicename> /uninstall /silent
```

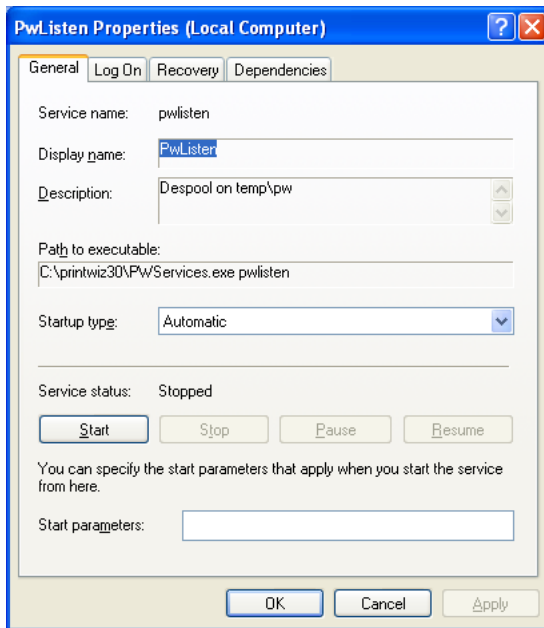
## 23.2 Managing Windows Services Through Windows

Print Wizard Windows services can be started and stopped through Windows tools as well. As an example, we will use the Control Panel:Administrative Tools:Services dialog of Windows XP. Other versions of Windows may have other administrative tools and there are other support tools that can manage services as well.



The above example shows the installed service “PwListen” and the status is “not started”. You can start the service from here, or right click on “PwListen” and look at its properties.





From the Properties dialog, you can change its “Startup type” or manually start the service running.

If something does not print once a service is started, check the printwiz.log (or the services log) to determine what may have gone wrong. If you have set up rules for a service, this is most often the culprit, but you will only know this by checking the log file. You can check it through the user interface, or by running the command:  
notepad <printwiz directory>\printwiz.log



## **24 Print Wizard Embedded**

Print Wizard Embedded is a version of Print Wizard embedded inside the AnzioWin, WēPO and Print Wizard DLL modules. Options for Print Wizard Embedded are passed to the engine using inter-program communications described elsewhere.

Refer to the documentation on Print Wizard Embedded found within the AnzioWin manual and also the sections elsewhere here. Additional information can be found in the sections below pertaining to the Print Wizard DLL and WēPO.



## 25 Print Wizard DLL – printwiz.dll

The Print Wizard DLL is Windows' dynamic link library that can be called from any other program to perform certain Print Wizard functions. Most anything you can do with the Print Wizard engine can also be called through the Print Wizard DLL.

The basic concept here is to configure the printer or set the printer name to a specific printer, print the file or the lines of text you wish and then flush the printer to close the print job. If you do not specify a printer (using either the "pw\_set\_printer" or the "pw\_configure\_printer") then the default printer will be used.

The various procedures are shown below (these may change and be added to over time, if something does not work or you need additional capabilities, contact us):

<b>Procedure</b>	<b>Return</b>	<b>Description</b>
pw_configure_printer(hwnd : handle):longint	Returns 1 if successful, else 0	Bring up the Windows standard "printer setup" dialog.
pw_set_printer(p : pchar)	None	Set the printer to the provided name.
pw_print_text(p : pchar; count : longint; cs : longint):longint	Returns 1 if successful, else 0	Print the text where "cs" is 0 for ISO 1 for OEM 2 for other (no translation) 3 for Unicode UTF-8 Count is the size of the text string to print.
pw_print_text_w(p : pwidechar; count : longint):longint	Returns 1 if successful, else 0	Print the string of text as wide Unicode characters. Count is the size of the text string to print.
pw_flush	None	Flush and end the print job.
pw_set_html_base(p : pchar)	None	Set the HTML base. This is the default directory used when an image file or other files are referenced by relative path (as defined in the HTML standards).
pw_set_preview(b : longint)	None	Set to 1 to preview print jobs and 0 to print them.
pw_set_orientation(b : longint)	None	Set to 0 for portrait, 1 for landscape and 2 for "auto" (automatically adjust orientation for text line lengths).
pw_set_font(p : pchar)	None	Set the name of the font to use.
pw_set_overlay(p : pchar)	None	Set the name of the overlay file.
pw_set_copies(n : longint)	None	Set the number of copies.

pw_print_emf(h : handle)	None	Set the EMF to print.
pw_eject_page	None	Eject the page.
pw_set_output(p : pchar; count : longint):longint	Returns 1 if successful, else 0	Set the file name to print to. Count is the size of the file name string.
pw_print_file(p : pchar; count : longint; cs : longint):longint	Returns 1 if successful, else 0	Set the name of the file to print. Count is the size of the file name and cs is the character set as described above.
pw_get_printer:variantstring	Returns the name of the printer or blanks if set to the PC default printer	Get the name of the printer that is being used.
pw_get_version:variantstring	Returns a string representing the version of Print Wizard's DLL	Get the version of Print Wizard as a string, i.e. "3.0".
pw_translate_pcl(n : longint)	none	Set to 0 for no translation; to 1 for translation of main file and overlay; to 2 for translation of main file only.
pw_device_caps(n : longint)	Returns the requested measurement page	Gets a measurement of the printer page. Passed value "n" indicates which measurement. Common examples are: 8 for horizontal width in pixels 10 for vertical height in pixels 12 for color bits per pixel 14 for number of color planes 88 for logical pixels per inch in horizontal 90 for logical pixels per inch in vertical 110 for physical width of page, in pixels 111 for physical height of page, in pixels 112 for horizontal unprintable offset, in pixels 113 for vertical unprintable offset, in pixels Other values are also possible, as documented for Windows call GetDeviceCaps.
pw_measure_text_w(p : pwidechar; ct : longint)	Returns the width	For the current active font and

	of the text, in pixels	other printing environment, tells the width of the indicated text, which must be in widechars.
--	------------------------	--





## 26 WēPO

Web Print Object, WēPO, is a separately licensed product.

Similar to the Print Wizard DLL, WēPO can be called from other sources, such as programs or from web pages, fairly simply.

The example below loads the WēPO ActiveX module from a specified web server the first time this page, or similar pages are called. It then uses the version on the local PC for each call thereafter. In this manner a web page can “force” a print job to a users local PC.

You can do this as a “web page control” (a “push” button), or set it to not show anything, but rather start from the page “OnLoad” function.

This only works with browsers that support ActiveX modules, such as Internet Explorer.

The first part is to define the control:

```
<OBJECT classid="clsid:4CE8026D-5DBF-48C9-B6E9-14A2B1974A3D"
codebase="http://www.anzio.com/controls30/printwizocx.cab#version=3,0,0,0"
width=0
height=0
align=center
hspace=0
vspace=0
ID="Pwbuttonx1" VIEWASTEXT>
<param name="mainurl"
value="http://www.anzio.com/pub/printwiz-samples/features.txt">
<param name="caption" value="Print">
</OBJECT>
```

The second is to call it from somewhere in the script, on button click, or from the OnLoad procedure:

```
<script language=javascript>
<!--
function click(){
var print;
alert('Going to change the caption');

document.Pwbuttonx1.caption='Wow!!!';
print = confirm('Going to print the sample without touching the button!!!');
if (print)document.Pwbuttonx1.Click();
document.Pwbuttonx1.caption='Double Wow!!!';
alert('Going to make it disappear!');
document.Pwbuttonx1.Visible=false;
alert('Going to make it reappear!');
document.Pwbuttonx1.Visible=true;
}

function buttonclick(){
```

```

document.Pwbuttonx1.Click();
}

function buttonhide() {
document.Pwbuttonx1.visible=false;
}

//-->
</script>
<div>&nbsp;&nbsp;&nbsp;</div>
<a href="javascript:click();">Hello World - Click here!!!!</a>

```

The print job can include any of the features described here, including barcodes, form overlays, page size selection, etc. Also, this approach can be an alternative to having the web browser print certain reports that the browser may not handle well.

The process of printing from a web page always involves two files. One file is the web page that includes a “print” button; the other file is the data to print – the print job. The two files can be on the same or different servers.

## 26.1 The Web Page

The web page is a standard HTML file that includes an <OBJECT> tag that references WēPO. The object tag indicates how WēPO appears and how it behaves. Typically, it will appear as a pushbutton that says “Print”.

The particular OCX is identified by its *classid*, a globally unique character string (see below). If the client’s PC does not have the Print Wizard OCX installed, the client will be advised of that by the browser, and given the chance to install it. This may require altering, at least temporarily, certain security settings. Once it is installed, the pushbutton will appear.

The web page also specifies certain parameters to the OCX that govern its behavior. These are in <param> tags. Most critical is the “mainurl”, which specifies the URL of the file that contains the print job.

A sample <OBJECT> tag is as follows:

```

<OBJECT
  classid="clsid:4CE8026D-5DBF-48C9-B6E9-14A2B1974A3D"
  codebase="http://www.anzio.com/controls/PWButtonXControl1.cab#version=3,0,0,0"
  width=100
  height=100
  align=center
  hspace=0
  vspace=0
>
<param name="Caption"      value="Click to print">
<param name="mainurl"     value="filetoprint">

```

```

<param name="debug"           value="1">
<param name="fontname"       value="courier new">
<param name="initfile"      value="c:\printwiz\_prtinit.txt">
<param name="LICENSEcode"   value="DFD5-NK45-MUKD-K7">
<param name="orientation"   value="auto">
<param name="printersetup"  value="0">
<param name="charset"       value="utf-8">
<param name="duplex"        value="left">
</OBJECT>

```

Following is an explanation of each part of the <OBJECT> tag:

Classid	A globally unique character string by which WēPO is registered. It is required. This string may change with changing versions of WēPO. Contact us for the correct code.
Codebase	Tells the browser where the OCX can be installed from, if it is not already on the client's PC.
Width	The width in pixels of the button to appear on the screen.
Height	The height in pixels of the button to appear on the screen.
Align	Where to display the button, relative to the current printing position. Options are "texttop", "middle", "left", "textmiddle", "right", "baseline", and "center". Consult a book on HTML for specifics.
Hspace	The amount of space to the left and right of the object, in pixels.
Vspace	The amount of space above and below the object, in pixels.

Following is an explanation of the various <PARAM> tags. Some of these tags are optional; the MAINURL is required. In each case, you can change what is in the "value" part.

Caption	The text to appear in the middle of the button.
Enabled	A zero value marks the button as disabled (grayed), so the user can not click it.
Visible	A zero value makes the button invisible.
MainURL	The file to be printed when the user clicks the button. Can specify a file on the local disk, or a URL starting with ftp://, http://, or https://.
Debug	A non-zero value tells the OCX to run in debug mode, displaying a lot of useful information in a popup window.
Fontname	Specifies an initial font for printing.
Initfile	Specifies a print initialization file. Can be on a local disk, or a URL.
LicenseCode	If a licensecode is specified, and it correlates to the location of the MainURL, then the OCX will NOT print out a line indicating it was printed with a demo version. Licensecodes are provided by Rasmussen Software.
Orientation	Sets the printing orientation. Possible values are "portrait", "landscape", or "auto".
Duplex	Tells the printer to print on both sides of the paper, if possible. Possible values are "off", "vertical", "horizontal", "left", "right", or blank. If blank, the duplexing will follow the printer driver's configuration.
PrinterSetup	A non-zero value will cause WēPO to prompt the user for Printer Setup, allowing them to choose a printer and set its properties.

PrinterName	Indicates which printer to use, by its Windows printer name. This can be a wildcard, consisting of some text followed by "*", such as "HP*"; it will be matched with the first available printer whose name fits this pattern. After printing, this parameter will contain the name of the printer that was used, such as the one selected via Printer Setup.
Charset	Specifies the character set of the print data. Options are "ISO", "OEM", and "UTF-8". Default is ISO.
Overlay	The name of a file to be used as a full-page form overlay.
TranslatePCL	Zero causes WēPO to pass any escape codes to the printer for processing. A value of 1 causes WēPO to translate any PCL codes in the main print job and/or the overlay file(s), allowing them to be printed on any printer. A value of 2 causes WēPO to translate the main file only; an overlay file that is PCL will be handled by the printer.
CaptionFont	The name of the font used to display the caption in the button. The default is set by the font of the preceding HTML.
CaptionHeight	The size (in points) of the font used for the caption. The default is set by the font of the preceding HTML.
CaptionStyle	A string representing the style to be used for the caption. Include, in any order, "B" for bold, "I" for italics, and/or "U" for underline. The default is set by the font of the preceding HTML.
Printed	After the button is clicked, this parameter will be non-zero to indicate that the printing succeeded. This parameter would not likely be set by the web page.

## 26.2 The Print Job File

The print job, that is the content, comes from a text file on an HTTP or FTP server. Or, it can be generated using CGI techniques. The file can be plain text, text with escape sequences (that would be unusual, though), PWML, or HTML (as long as it was within the features handled by Print Wizard).

## 26.3 Samples

To see the Web Print Object in action, visit the web site

<http://www.anzio.com/products/wepo.htm> .



## 27 MapiSend

There are a number of other small programs that may or may not be included with your Print Wizard program. There are some that are deprecated, some that are only included with the embedded version of Print Wizard and some available only if you have purchased the Service Edition.

We will not cover all these in depth here. There is one, however, that should be mentioned. This is MapiSend.

### 27.1 What Is MAPI?

MAPI is the Microsoft Messaging API, a set of message handling tools for Windows 95, 98, ME, 2000 and NT. These services are commonly loaded when you install Internet Explorer or Outlook Express (as basic MAPI), or when you install Microsoft Outlook, Office or Exchange (extended MAPI). Among other possibilities, the basic MAPI protocols support e-mail and faxing.

For e-mail, you must have a configured MAPI e-mail reader and have set up the reader for sending mail. This is basically available once you have set up Outlook or Outlook Express.

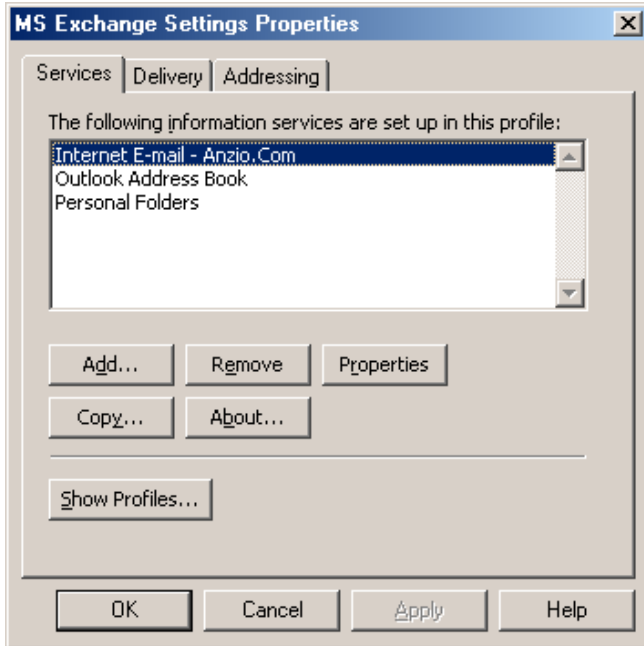
For faxing, you must have installed a MAPI-compatible fax application, and must be working with Microsoft Outlook. With Windows 98, Microsoft Fax was included and works nicely (this can also be installed in Windows 95, but may not be installed by default – it depends on which service packs you have installed). For Windows ME, 2000 and NT, other third-party faxing software may be necessary.

### 27.2 How do you set/change MAPI options?

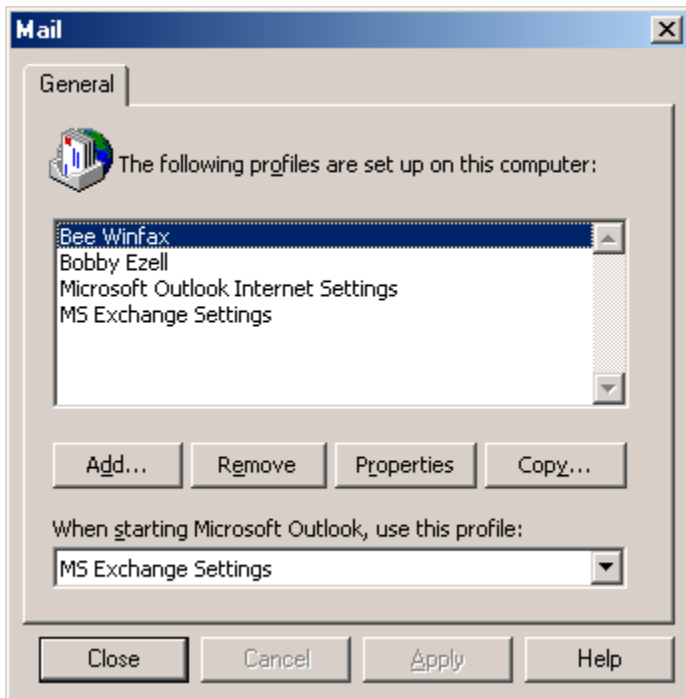
Basic MAPI works from the installed default e-mail setup. MAPI works off the idea of profiles, a set of services available to a particular logged in person. These are controlled by two different Control Panel dialogs, MAIL and INTERNET OPTIONS.

#### 27.2.1 Mail

From the Control Panel:Mail dialog, you can set up the user profiles and the properties for the various services. When you install Outlook or Outlook Express certain defaults are assumed, so you may want to check these settings.



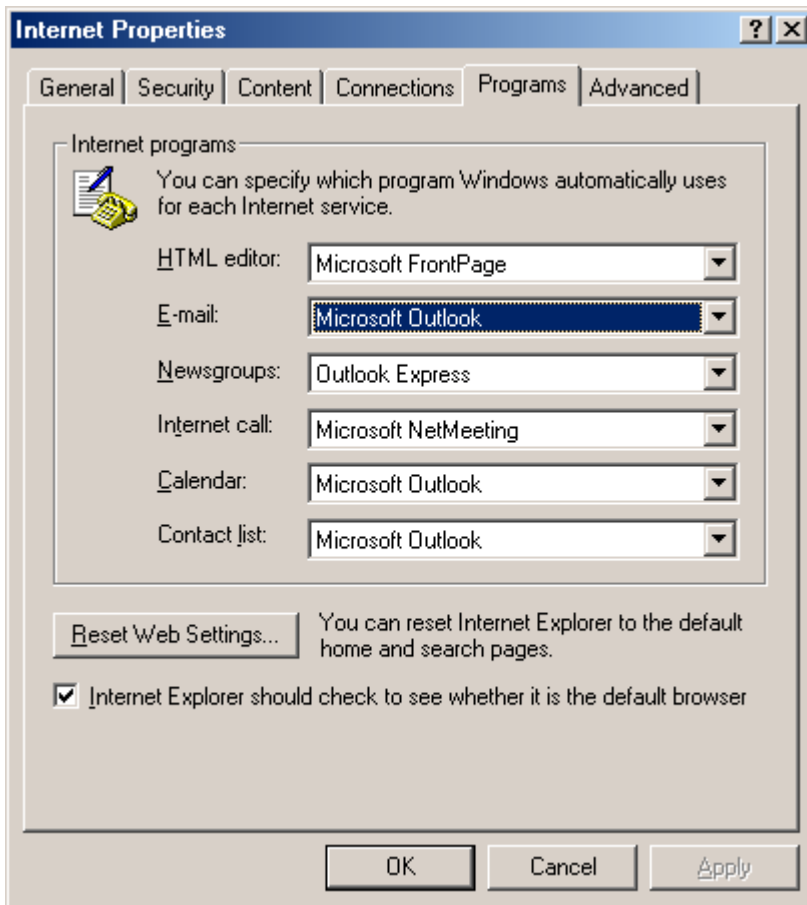
Mail determines the available services to your system, and hence a selected profile. Clicking on “Show Profiles” will bring up the following dialog.



Clicking on Properties here displays a screen similar to the prior screen, but showing only those services available for a selected profile.

## 27.2.2 Internet options

Internet Options do not have a lot to do with the various MAPI settings, except to determine the default application to use for e-mail services. This option covers more than just e-mail however, it covers the behavior of MAPI in general. If Outlook Express is selected in the applications for e-mail, then simple MAPI is used, whereas if Outlook or Exchange is selected, extended MAPI is used. Also this setting determines whether or not faxing will work correctly.



Normal options include Microsoft Outlook, Outlook Express or Netscape Messenger. All three will work for sending e-mail (see our notes below), but only Microsoft Outlook will work for interfacing to faxing.

## 27.3 Running MapiSend

MapiSend is designed to provide a command-line interface between the various MAPI functions and the user, without having to delve into writing your own MAPI application. Primarily, MapiSend provides Print Wizard with a method of e-mailing and faxing through the default installed mail and fax services.



The general format for MapiSend is:

MapiSend [options] filename [filename2...]

Where options can be:

/n<note text>	The text to include as the body of the message if there are embedded spaces, be sure to include the text in quotes).
/f<faxnumber>	To fax, include the number here, just as you would dial it
/r<recipient>	The recipient name to include (quote it if embedded spaces).
/s<subject>	The subject text to include (quote it if embedded spaces).
/e<email>	The email address to send to
/p<profile>	The Outlook/Exchange profile to use (if this is included, the default service is used, be sure to enclose in quotes if spaces).
/i<include note file>	The full path to a file that contains a body of text (replaces the /n note (if spaces, enclose file name in quotes).
/d	Turn on debug messages to aid in debugging errors, including bringing up the send dialogs to manually send messages.

At least one file name is required. You must include a /e or /f option to either send e-mail or send a fax. If you are using Microsoft Outlook, you must also include the /p option with a valid user profile.

## **27.4 How it works – Notes on Various Configurations**

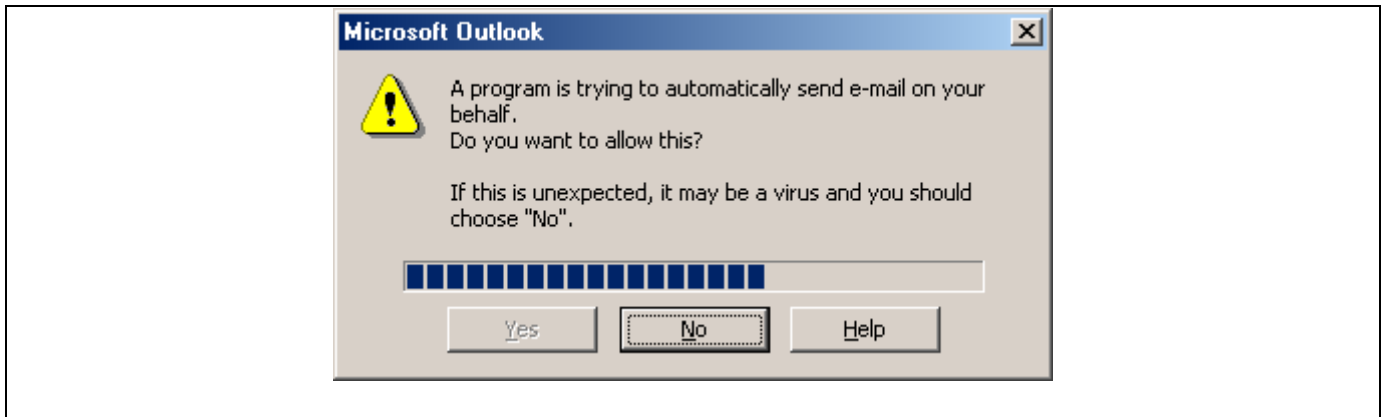
The following will go through several configuration notes. These fall in to the categories of:

Microsoft Outlook e-mailing  
 Outlook Express e-mailing  
 Netscape Messenger (version 4.0 or later) e-mailing  
 Microsoft Outlook faxing  
 Outlook Express and Netscape Messenger faxing  
 Others

Note that Microsoft Outlook and Exchange behave pretty much the same way.

### **27.4.1 Microsoft Outlook e-mailing**

- You must specify a profile name with the /p<profile> option.
- The “From” address and name will be from the profile setting.
- You may get the following Security dialog in later versions of Outlook, depending on your configuration



- You may experience delays, depending on a) whether you are set to send e-mails automatically, and b) if your server experiences any delays. If you are not set for automatic e-mailing for some reason, you may need to bring up Outlook to begin the send.
- Attachments come in as regular attachments, even if they are standard text (.txt) files.

### 27.4.2 Outlook Express e-mailing

- You don't need to include any profile name
- The "From" address is determined from the default e-mail settings, not from a profile, so this could appear different than that for a profile
- If any bad or unsent e-mails are in the outbox, you may get a dialog box up even if there is no "/d" option. You will have to close this to continue. If everything in your message is formatted correctly, your message should still go out (read through the messages carefully).
- Attachments come in as regular attachments, even if they are a standard text (.txt) file.

You may have trouble getting the latest releases of Outlook Express to work properly with MapiSend. This is partly because several vulnerabilities in Outlook Express, and MAPI in general, have been patched to require some type of user intervention when Outlook is not present. Also, there are various versions of Microsoft's MAPI.DLL that get installed depending on the version of Internet Explorer and Outlook Express.

You may need to install Outlook in order to get all features to work automatically (no user intervention).

### 27.4.3 Netscape Messenger e-mailing

- No Netscape profile is necessary and does not apply for MAPI.
- "From" comes from the default e-mail configuration.
- Any attachments with the .txt extension will usually become part of the body text (no way around this that I know of).

### 27.4.4 Microsoft Outlook faxing

- Use the “/f<phone number>” tag with complete phone number (no dialing properties apply).
- Include a profile that has the Microsoft Fax or the third-party fax service available.
- “From” comes from the profile name.
- This does not apply to Windows ME, 2000 or NT. Windows 98 comes with Microsoft Fax installed, and it depends on which service pack you use for Windows 95. It can also be downloaded from Microsoft for Windows 95, but will not work under ME, 2000 or NT.

### 27.4.5 Outlook Express and Netscape Messenger faxing

If you try to send a fax when either program is set as the default e-mail client (Internet Options), they send the request through as an e-mail service with the phone number becoming the e-mail address to send to. Most all e-mail servers will then reject this as “User Unknown” and cause an error dialog box to appear.

### 27.4.6 Others

There are other third-party e-mail applications available. The question is whether they support MAPI the same way. Many applications claim to be MAPI compliant, but may not support inter-program access.

There are also other third-party faxing applications available. Again while these may support MAPI, they may not support inter-program access correctly. Also note that a fax application without an e-mail application will be totally dependent on the selected e-mail application for MAPI and may not work correctly as a remote service (Winfax Pro is an example of this – while it supports MAPI as an Outlook helper application, it does not support MAPI as the default fax for Outlook).

### 27.4.7 Examples

#### **Sending an email with Microsoft Outlook as the email reader:**

```
Mapisend /s"Subject line" /n"Note for the body" /euser@domain.com /p"My Profile" file1.txt file2.xls
```

#### **Sending an e-mail with Microsoft Outlook and an “include” file for the body:**

```
Mapisend /s"Subject line" /ibody.txt /euser@domain.com /p"My Profile" file1.txt file2.xls
```

#### **Sending an e-mail with Outlook Express or Netscape Messenger as the e-mail reader:**

```
Mapisend /s"Subject line" /n"Note for the body" /euser@domain.com file1.txt file2.xls
```

#### **Sending an e-mail with Microsoft Outlook and an “include” file for the body:**

```
Mapisend /s"Subject line" /ibody.txt /euser@domain.com file1.txt file2.xls
```

#### **Sending a fax with Microsoft Outlook as the e-mail reader:**

```
Mapisend /p"My Profile" /f1-503-624-0760 file1.txt file2.xls
```

## 27.5 Errors

The following are just some of the possible errors (taken from the Microsoft include libraries). See Mapi or Outlook specific documentation for more information on each error. We do not guarantee the accuracy of these as they may change from release to release of Mapi. These are merely here as an example and to perhaps help understand what error Windows Mapi may be passing to you.

MAPI_E_NO_SUPPORT	02
MAPI_E_BAD_CHARWIDTH	03
MAPI_E_STRING_TOO_LONG	05
MAPI_E_UNKNOWN_FLAGS	06
MAPI_E_INVALID_ENTRYID	07
MAPI_E_INVALID_OBJECT	08
MAPI_E_OBJECT_CHANGED	09
MAPI_E_OBJECT_DELETED	0A
MAPI_E_BUSY	0B
MAPI_E_NOT_ENOUGH_DISK	0D
MAPI_E_NOT_ENOUGH_RESOURCES	0E
MAPI_E_NOT_FOUND	0F
MAPI_E_VERSION	10
MAPI_E_LOGON_FAILED	11
MAPI_E_SESSION_LIMIT	12
MAPI_E_USER_CANCEL	13
MAPI_E_UNABLE_TO_ABORT	14
MAPI_E_NETWORK_ERROR	15
MAPI_E_DISK_ERROR	16
MAPI_E_TOO_COMPLEX	17
MAPI_E_BAD_COLUMN	18
MAPI_E_EXTENDED_ERROR	19
MAPI_E_COMPUTED	1A
MAPI_E_CORRUPT_DATA	1B
MAPI_E_UNCONFIGURED	1C
MAPI_E_FAILONEPROVIDER	1D
MAPI_E_UNKNOWN_CPID	1E
MAPI_E_UNKNOWN_LCID	1F

/\* Flavors of E\_ACCESSDENIED, used at logon \*/

MAPI_E_PASSWORD_CHANGE_REQUIRED	20
MAPI_E_PASSWORD_EXPIRED	21
MAPI_E_INVALID_WORKSTATION_ACCOUNT	22
MAPI_E_INVALID_ACCESS_TIME	23
MAPI_E_ACCOUNT_DISABLED	24

/\* MAPI base function and status object specific errors and warnings \*/

MAPI_E_END_OF_SESSION	100
MAPI_E_UNKNOWN_ENTRYID	101
MAPI_E_MISSING_REQUIRED_COLUMN	102
MAPI_W_NO_SERVICE	103



## 28 NetPrint

The Print Wizard product (but not AnzioWin) includes a program for various Unix (and similar) platforms, called "netprint". This program is designed to work with Print Wizard's "JetDirect" protocol. It will also work with HP JetDirect and similar devices. It takes piped input and sends it to a particular machine name (or IP address) and port (default 9100).

You will need to identify which NetPrint object is correct for your host system, by noting the file extension. For instance, the file "netprint.sco" in the "netprint.obj" directory is for SCO Open Server 5. Check the README.TXT file for other versions. Contact Rasmussen Software if you need NetPrint compiled for another system. Once you've identified your object, move it to the host system, such as with FTP. Be sure to transfer it as binary. Rename it, make it executable, and put it in your PATH; for instance:

```
mv netprint.sco netprint
chmod +x netprint
cp netprint /usr/bin
```

The print data must always be piped into NetPrint. For instance:

```
ls -l | netprint
```

will send a directory listing to NetPrint, with a default port number of 9100.

But what machine will NetPrint send the data to? The NetPrint program has a unique feature in that it assumes you want to send the data to the same PC you are logged in from in a telnet session. (To do this, NetPrint looks at the utmp or utmpx file in Unix.) This results in a form of "follow-me printing". Regardless of where I log in from, even with dynamic IP assignments, NetPrint will send the print job to my IP address, where Print Wizard can process it and send it to the correct printer. Note, however, that this will not work if the Unix spooler is trying to process the output, because it will have lost track of the login name. So generally, to use NetPrint, you will want to pipe your print out to NetPrint **instead** of to the Unix spooler.

Of course, you can also specify a particular destination machine. The complete syntax of NetPrint is:

```
netprint [options]
```

where options include

- p *nnn* use port number *nnn*
- h *xxx* send to hostname (or IP number) *xxx*
- d *nnn* set debug level to *nnn*.

Note that if the destination machine is behind a firewall (or other device) that is doing Network Address Translation (NAT), you will need to configure the firewall to direct these print jobs to the correct PC, based on the port number.

# Appendices

## Appendix A PWML Technical Reference

The following table indicates all the tags and parameters that Print Wizard recognizes. Items marked with “+” are extensions to the HTML standard. Where units of measure are involved, the table shows what unit is assumed; see above for explanation. The table also indicates when relative or percentage changes are allowed, and where exact units can be specified.

```

<!DOCTYPE>
    establishes document as HTML/PWML
    otherwise ignored

<HTML> ... </HTML>
    establishes document as HTML/PWML
    otherwise ignored

<PWML> ... </PWML>
    establishes document as HTML/PWML
    otherwise ignored

<HEAD> ... </HEAD>
    (ignored)

<META>
    CONTENT="CHARSET=UTF-8"

<PAGESIZE> +
    DOTSPERINCH
        establishes DPI and DOT
    IMGDOTSPERINCH
        establishes IMG-DPI and IMG-DOT
    X=n
        in DOTs
        relative, %, units
    Y=n
        in DOTs
        relative, %, units
    PAPER=n
        1 or "letter"
        2 or "lettersmall"
        3 or "tabloid"
        4 or "ledger"
        5 or "legal"
        6 or "statement"
        7 or "executive"
        8 or "A3"
        9 or "A4"
        10 or "A4small"
        11 or "A5"
        12 or "B4"
        13 or "B5"
        14 or "folio"
        15 or "quarto"

```



```

20=#10 envelope
or other Windows standard sized, by number
or printer-specific paper name
<pattern>
BIN=n
  1=upper
  2=lower
  3=middle
  4>manual
  5=envelope
  6=envelope manual
  7=auto
  8=tractor feed
or other Windows standard sized, by number
or printer-specific paper name
<pattern>
MEDIA=n
  1 or "Standard"
  2 or "Transparency"
  3 or "Glossy"
or printer-specific media name
<pattern>
COPIES=n
  number
  <pattern>
COLLATE=n
  0 or "Off"
  1 or "On"
  <pattern>
ORIENTATION=s
  1 or "landscape"
  0 or "portrait"
  2 or "auto"
  ROTATEWIDTH=n
    Number of characters
  ROTATEPITCH=n
    DOTS
    Relative, %, units
LENGTH=n
  lines per page
LINESPACING=n
  vertical, DOTS
  relative, %, units
OVERLAY=filename
  always printed at full page size
  currently supports PDF, BMP, JPEG, JPG, GIF, TIF, WMF, EMF, PWML, and PCL
formats
  or "scan://"
PRELOAD=n
  Stores overlay into PCL printer
PREVIEWOVERLAY=filename
  displayed in preview window, but not printed
  currently supports PDF, BMP, JPEG, JPG, GIF, TIF, WMF, EMF, PWML, and PCL
formats
  or "scan://"
USEOVERLAY=n
  number

```

```
pattern
  use PCL macro(s) already in printer
DUPLEX=xxx
  VERTICAL
    Binding on long edge
  HORIZONTAL
    Binding on short edge
  LEFT
    Binding on left edge, respecting orientation
  TOP
    Binding on top edge, respecting orientation
  OFF
OFFSETX=xxx
  in DOTS
    moves all printer output to the right
OFFSETY
  in DOTS
    moves all printer output down
JOBOPTIONS=xxx
  applies only when creating PDF
  in format "<variable>=<value>?<variable=<value>..."
  variables:
    PageLayout=xxx
      SinglePage
      OneColumn
      TwoColumnLeft
      TwoColumnRight
      TwoPageLeft
      TwoPageRight
    PageMode=xxx
      UseNone
      UseOutlines
      UseThumbs
      FullScreen
      UseAttachments
    Page=n
      page to display at startup
    OpenAction=xxx
      Fit
      FitV
      FitH
      FitB
      FitBH
      FitBV
      numeric zoom factor
      default
    FitWindow=xxx
      True (or blank)
      False
    CenterWindow=xxx
      True (or blank)
      False
    HideMenuBar=xxx
      True (or blank)
      False
    HidewindowUI=xxx
      True (or blank)
```

```

    False
    DisplayDocTitle=xxx
    True (or blank)
    False
    PrintScaling=xxx
    AppDefault
    None

```

<BASE>

```

    HREF=s
        base location for images
    EXT=s +
        assumed extension for IMG files

```

<BODY> ... </BODY>

```

    DOTSPERINCH +
        establishes DPI and DOT
    IMGDOTSPERINCH +
        establishes IMG-DPI and IMG-DOT
    TOPMARGIN=n
        in DOTS
        relative, %, units
    BOTTOMMARGIN=n
        in DOTS
        relative, %, units
        measured from top edge of paper
    LEFTMARGIN=n
        in DOTS
        relative, %, units
    GUTTER=n
        in DOTS
        relative, %, units
        adds space on left side (or top, if DUPLEX setting indicates binding
        there)of odd-numbered pages, right side (or bottom) of even-numbered pages
    RIGHTMARGIN=n
        in DOTS
        relative, %, units
        measured from left edge of paper
    COLUMNOFFSET=n +
        in DOTS
        units
    COLUMNGUTTER=n +
        In DOTS
        units
    ROWOFFSET=n +
        in DOTS
        units
    ROWGUTTER=n +
        In DOTS
        units
    LABELORDER=ROW
        otherwise in column order
    WRAP
        as in <PRE> below
    ALIGN
        as in <PRE> below
    LINESPACING

```

as in <FONT> below

```
<PRE> ... </PRE>
  can contain tags
  can contain character entities
  WIDTH=n
    characters per line
  WRAP=s
    NONE
      extra characters slop over margins
    CHAR
      break line on any character
    WORD
      break line on a word
    CHOP
      discard characters that don't fit
    SQUEEZE
      reduce font size until text fits
  ALIGN=s
    LEFT
    CENTER
    RIGHT
    FILL
```

```
<LISTING> ... </LISTING>
  implies 132 characters per line
  no embedded tags
  no character entities
  WRAP
    as in <PRE> above
  ALIGN
    as in <PRE> above
    deprecated
```

```
<XMP> ... </XMP>
  implies 80 characters per line
  no embedded tags
  no character entities
  WRAP
    as in <PRE> above
  ALIGN
    as in <PRE> above
    deprecated
```

```
<PLAINTEXT>
  no tags can follow, not even </PLAINTEXT>
  no character entities
  WIDTH=n
    characters per line
  WRAP
    as in <PRE> above
  ALIGN
    as in <PRE> above
    deprecated
```

```
<LEGACY> ... </LEGACY>
  can contain tags prefixed by trigger
```

can contain character entities prefixed by trigger  
 TRIGGER=n  
 describes the trigger character in decimal  
 default trigger is control-Z

WRAP  
 as in <PRE> above

ALIGN  
 as in <PRE> above

<P> ... </P>  
 paragraph break

WRAP  
 as in <PRE> above

ALIGN=s  
 as in <PRE> above

LEFTMARGIN=n  
 in DOTs

RIGHTMARGIN=n  
 in DOTs

<DIV> ... </DIV>  
 division break, treated as paragraph break

WRAP  
 as in <PRE> above

ALIGN=s  
 as in <PRE> above

LEFTMARGIN=n  
 in DOTs

RIGHTMARGIN=n  
 in DOTs

<SMALL> ... </SMALL>  
 makes text 20% smaller

<BIG> ... </BIG>  
 makes text 25% larger

<H1> ... </H1> through <H6> ... </H6>  
 headers  
 sets text size to 24pt, 18pt, 14pt, 12pt, 10pt, 7pt respectively

<FONT> ... </FONT>  
 FACE=s  
 Windows font name

FACE="face1, face2, face3"  
 Windows font names for primary, additional fonts  
 use "-face4" to prohibit use of face4  
 end with "auto" for continued auto-selection

SIZE=n  
 1 through 7, arbitrarily assign to point sizes  
 relative, %, units  
 affects linespacing

POINTSIZE=n +  
 exact height in points  
 relative, %, units  
 affects linespacing

CELLSIZE=n +  
 height of character cell in points  
 relative, %, units

affects linespacing  
 PITCH=n +  
   in DOTS  
   relative, %, units  
   forces horizontal spacing of characters  
 LINESPACING=n +  
   in DOTS  
   relative, %, units  
 ROTATION=n +  
   Any value, in degrees  
 COLOR=xxx +  
   Colors listed elsewhere  
 BACKGROUND=xxx +  
   Colors listed elsewhere

<SUP> ... </SUP>  
 Superscript

<SUB> ... </SUB>  
 Subscript

<GOTO> +  
 X=n  
   in DOTS  
   relative, %, units  
   accounts for unprintable area  
   specifies left side of character cell  
 Y=n  
   in DOTS  
   relative, %, units  
   accounts for unprintable area  
   specifies top of character cell by default (see ALIGN)  
 ALIGN=xxx  
 TOP  
   The Y argument applies to the top of the character cell  
 BASE  
   The Y argument applies to the baseline of the character  
 BOTTOM  
   The Y argument applies to the bottom of the character cell  
 SAVE=X  
   save X position for later RESTORE  
 SAVE=Y  
   save Y position for later RESTORE  
 SAVE=XY or SAVE=YX  
   save both X and Y for later RESTORE  
 RESTORE=X  
 RESTORE=Y  
 RESTORE=XY  
 RESTORE=YX

<B> ... </B>  
 bold on/off

<U> ... </U>  
 underline on/off

<I> ... </I>  
 italics on/off

<BR>

Break line

<LINE> +

X=n  
 in DOTs  
 relative, %, units  
 default is current position

Y=n  
 in DOTs  
 relative, %, units  
 default is current position

X2=n  
 in DOTs  
 relative, %, units

Y2=n  
 in DOTs  
 relative, %, units

COLOR=s  
 color names described elsewhere

WIDTH=  
 in DOTs  
 relative, %, units  
 default is 1 pixel

<HR>

horizontal rule or rectangle

SIZE=n  
 height in DOTs  
 relative, %, units

WIDTH=n  
 in DOTs  
 relative, %, units

COLOR=xxx  
 Colors listed elsewhere

OUTLINE=n  
 width of outline, in DOTs  
 relative, %, units  
 always black

X=n +  
 in DOTs  
 relative, %, units  
 accounts for unprintable area  
 defaults to centered on page

Y=n +  
 in DOTs  
 relative, %, units  
 accounts for unprintable area  
 defaults to next line

<IMG>

SRC=s  
 file name, with forward slashes  
 can be URL  
 affected by BASE .. HREF, EXT  
 can be "SCAN://" or variations  
 currently supports BMP, JPEG, JPG, GIF, TIF, WMF, and EMF formats

ALIGN=s  
 TOP

```

    top of image to top of text
MIDDLE
    middle of image to baseline of text
BOTTOM
    bottom of image to baseline of text
    default
BORDER=n
    in DOTS
    relative, %, units
    width of black line around image
HEIGHT=n
    in IMG-DOTS
    defaults to bitmap's height
WIDTH=n
    in IMG-DOTS
    defaults to bitmap's width
HSPACE=n
    horizontal space around image
    in DOTS
    relative, %, units
VSPACE=n
    vertical space around image
    in DOTS
    relative, %, units
X=n +
    in DOTS
    relative, %, units
    accounts for unprintable area
    defaults to right after last text
Y=n +
    in DOTS
    relative, %, units
    accounts for unprintable area
    defaults to next line

<BARCODE> +
    STYLE=s
        UPCA
        UPCE
        EAN13
        EAN8
        2OF5
        CODABAR
        CODE11
        CODE39
        CODE93
        CODE128
        CODE128B
        CODE128C
        POSTNET
        PDF417
        MAXICODE
    FACE=s
        Windows font name for caption text
        does not affect normal text
    POINTSIZE=n
        in points

```



```

    relative, %, units
SIZE=n
    in points
    relative, %, units
X=n
    DOTs
    relative, %, units
    specifies left edge
Y=n
    DOTs
    relative, %, units
    specifies top edge
SRC=s
    String to be barcoded
SUPPLEMENT=s
    String for barcode supplement
BARWIDTH=n
    inches
    relative, %, units
    defaults to .012in
WIDTHRATIO=n
    wide-to-narrow ratio
    no units
    default is 2
BEARERBARS=b
    OFF or ON
TALLGUARDBARS=b
    OFF or ON
CAPTION=b
    OFF or ON
SHOWGUARDCHARS=b
    OFF or ON
ADDCHECKCHAR=b
    OFF or ON
ROTATION=n
    0, 90, 180, 270
ALIGN=s
    TOP
        top of barcode to top of text cell
        default
    BASE
        bottom of barcode to baseline of text
    BOTTOM
        bottom of barcode to bottom of text cell

<REPLACE> +
    IN=s
        regular expression
    OUT=s
        regular expression
    LINE=n
        line number to which REPLACE operation should be applied
<TABLE> ... </TABLE>
    HTML-style tables
X=xxx
    in DOTs
    relative to current X position

```

```

Y=xxx
  in DOTS
  relative to current Y position
HEIGHT=xxx
  in DOTS
  in % of margin height
WIDTH=xxx
  in DOTS
  in % of margin width
ALIGN=xxx
  Horizontal alignment within table
  RIGHT, CENTER, JUSTIFY, LEFT
CELLSPACING=xxx
  in DOTS
  default 2
CELLPADDING=xxx
  in DOTS
  default 1
BORDER=xxx
  in DOTS
  default 1
RULES=xxx
  Lines between table elements
  ROWS, COLS, ALL
FRAME=xxx
  Lines around whole table
  ABOVE, BELOW, LHS, RHS, HSIDES, VSIDES, BOX, BORDER, VOID
BORDERCOLOR=xxx
  Color of border
  Colors define elsewhere
BGCOLOR=xxx
  Background color of table
  Colors defined elsewhere
<TBODY> ... </TBODY>
  Table body
  (no params)
<TR> ... </TR>
  HEIGHT=xxx
  in DOTS
  default to 12 points
  ALIGN=xxx
  Horizontal alignment within row
  RIGHT, CENTER, JUSTIFY, LEFT
  VALIGN=xxx
  Vertical alignment within row
  TOP, MIDDLE, CENTER, BOTTOM
  BGCOLOR=xxx
  Background color of row
  Colors defined elsewhere
  DIR=xxx
  RTL
  LTR
<TD> ... </TD>
  ALIGN=xxx
  Horizontal alignment within cell
  RIGHT, CENTER, JUSTIFY, LEFT
  VALIGN=xxx

```

Vertical alignment within cell  
 TOP, MIDDLE, CENTER, BOTTOM

WIDTH=xxx  
 in DOTS  
 in % of table width

BACKGROUND=xxx  
 Background color of cell

HEIGHT=xxx  
 Height of column  
 In DOTS

BGCOLOR=xxx  
 Background color of cell

COLSPAN=n  
 Number of columns this cell spans

DIR=xxx  
 RTL  
 LTR

<TH> ... </TH>  
 Table column header  
 (same options as TD)

<STRONG> ... </STRONG>  
 (same as <B>)

<FORMFEED> +  
 Advance to next page  
 no parameters

<LABELFEED> +  
 Advance to next label in labeled output

<INDEX> +  
 TEXT  
 ACTION  
 ARG1

<ATTACH> +  
 SRC  
 DESC

<TAIL> ... </TAIL>  
 Text to be applied at end of document  
 (no params)

<ARCTEXT> +  
 SRC=s  
 The text to be printed along an arc  
 X=n  
 The horizontal location of the starting text point  
 Default is current printing position  
 Y=n  
 The vertical location of the starting text point  
 Default is current printing position  
 X2  
 The horizontal location of the center of the circle  
 Default is equal to X  
 Y2  
 The vertical location of the center of the circle  
 Default is one inch below the Y value above.  
 DIR=s  
 CW = Clockwise (default)  
 CCW = Counterclockwise  
 ALIGN=s  
 Which part of the text is at (X, Y)

```

    LEFT
    RIGHT
    CENTER
<PASSTHRU> +
    Text to be sent at low level to printer
    SRC=s
<SIGNATURE> +
    Information for a digital signature (certificate) for a PDF
    AUTHOR=s
        The author name to be shown in the signature
    REASON=s
        The reason to be shown in the signature
    CERTIFICATE=s
        The pathname of the certificate file to be used to sign the document
    PASSWORD=s
        The password necessary to read the CERTIFICATE file
    STYLE=s
        SIGN
        VISIBLE
        INVISIBLE
    SRC=s
        Pathname for an image file to show in the document, behind the signature
details.
    WIDTH=n
        Width of the signature image
    HEIGHT=n
        Height of the signature image
    X=n
        Horizontal location of the signature image
    Y=n
        Vertical location of the signature image
    ALIGN=n
        Alignment (vertically) to which Y applies
        TOP
        MIDDLE
        BOTTOM
    BORDER=n
        Width of the border around the signature image
<A> ... </A>
    HREF
<SPAN> ... </SPAN>
    WRAP=xxx
        as in <PRE> above
    ALIGN=xxx
        as in <PRE> above
    WIDTH=xxx
        in DOTs
<TITLE> ... </TITLE>
    Information within this section is ignored.
<FORM> ... </FORM>
    Information within this section is ignored.
<APPLET> ... </APPLET>
    Information within this section is ignored.
<SCRIPT> ... </SCRIPT>
    Information within this section is ignored.
<STYLE> ... </STYLE>
    Information within this section is ignored.

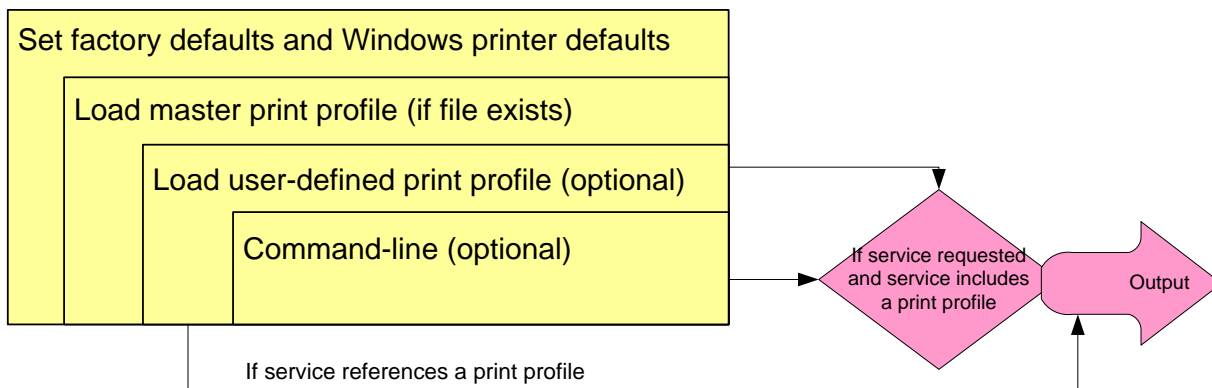
```

```
<CENTER> ... </CENTER>
  Information within this section is ignored.
<ENV>
  Field to be replaced
  SRC=s
```

## Appendix B Overriding Print Wizard Defaults

Print Wizard and many of its associated applications contain the same startup procedures with few exceptions. This includes loading factory and Windows defaults first, followed by a master profile if present, then any requested user profiles, and finally, and top priority, are any command-line parameters.

The exception to this is if a Windows or Foreground Service Name is being used, and that Service Name record points to another print profile. This print profile then is loaded last, overriding any current settings.



The procedures and files involved in the initialization of Print Wizard are as follows:

1. Print Wizard factory settings are set.
2. If “printwiz.ini” exists, it is loaded. This file contains global settings for the user interface, as well as default program name and object name settings.
3. If the print profile “master.profile” exists, it is loaded over the top of the factory settings.
4. If the command line specifies a “/profile=<profilename>”, and that file exists, the profile is then loaded next.
5. If the command line contains any switches other than “/profile”, those options are applied next.
6. If the command line specifies a “/service=<servicename>”, and that service points to a print profile, that print profile gets loaded prior to each print job.

## Appendix C Print Wizard Examples

The Print Wizard engine can be run from the user interface, by a Print Wizard or Windows service, via a script, from the Windows Start:Run dialog or through some third-party program. You can also drag and drop files on the desktop icons or on the running program. In other words, there are several ways of providing input to the engine.

Likewise, there are several things you can do with the output from the Print Wizard engine. It can be printed, previewed, faxed, emailed, used to generate a PDF, or sent to a file.

Below are several examples of using Print Wizard engine directly. These examples cover a wide range of possibilities.

```
printwiz testfile
printwiz testfile /fpdf://
printwiz /x5036240760 testfile
printwiz testfile testfile.txt test.txt
printwiz testfile /l /debug /ffax32://5036240760
printwiz /s testfile
printwiz testfile /overlay=c:\printwiz30\examples\mountain.jpg
printwiz testfile /preview /previewoverlay=c:\pritrnwiz30\examples\mountain.jpg
printwiz testfile /preview

printwiz /x5036240760 /t"this is to me" /n"this is line 1|and this is line 2" /j"this is the
subject" /ac:\printwiz30\test.doc testfile

printwiz /x5036240760 /t"this is to me" /n"c:\printwiz30\testfile.txt /j"this is the subject"
/ac:\printwiz30\test.doc testfile

printwiz /x5036240760 /t"this is to me" /n"c:\printwiz30\testfile.txt /j"this is the subject"
/ac:\printwiz30\test.doc

printwiz /x5036240760 /n"test notes line 1|and line 2|and line 3" testfile

dir | printwiz -

printwiz /waitforfile c:\temp\pw\testfile
    (waits for the file to show up, print it, then quits)

printwiz /despool c:\temp\pw\*.txt c:\temp\pw2\

printwiz /despool c:\temp\pw\*.txt c:\temp\pw2\r1 c:\temp\pw3\*.
    (despools *.txt, a file called 'r1', *.* from appropriate locations)

printwiz -listen

printwiz /service=pwlisten
    (starts service listening on port 9100 via a service called "pwlisten")

printwiz /service=pwdespool
```

(despools from a directory found in the service called "pwdespool")

Drop a file on printwiz.exe shortcut

(with no profile and smart print on  
(any file works fine if file-ext defined)

Drop a file on printwiz.exe shortcut with "winfaxpro.profile"

(with a "to", fax number and default fax of "winfaxpro")

Drop a file on printwiz.exe shortcut with "fax32.profile"

(not doing rendering right now to DC, does everything as attachments)

Drop a file on printwiz.exe shortcut with no profile and smart print off

(be careful if file is not printable by Print Wizard, at present we do  
no error checking here)

Drop a file on printwiz.exe shortcut with "customprint.profile"

(set up a test profile to show text in notepad)

Drop a file on printwiz.exe shortcut with "bytypeextsub.profile"

(set a shortcut to force a new extension with override for .doc and  
then fire off the document - similar to Smart Print)

Drop a file on printwiz.exe shortcut with "bytypeextsub.profile"

(set to only apply extension if none exists for .doc, it then changes  
the extension and fires off the document file)

Drop a file on printwiz.exe shortcut with no profile but a '-x<>'

(fine if printwiz can handle the file, it will then fax it according  
To default fax method)

Drop a file on printwiz.exe shortcut with no profile but '-email=<>'

(handles all files as attachments so works fine)

printwiz a file with !WINFAX /x<> /a<> <data>

(faxes single page plus attachment page, 2 pages total)

printwiz a file with !WINFAX /x<> /a<> /a<> /a<> /n<> <data>

(faxes multiple pages, one for each attachment, one for data, one for  
coverpage (notes))

printwiz a file with !ACROBAT <data>

(creates PDF and places it as a temp name in "documents and  
Settings"\bobby ezell\local settings\temp\)

printwiz a file with !ACROBAT <data> and /f<filename> on command-line

(creates PDF where you told it to)

printwiz a file with !ACROBAT /f<filename> <data>

(creates PDF where you told it to)

printwiz a file with !ACROBAT /x<> /n<> /j<> /p<> <data>

(send it via email, note that the /p is a profile, not printer name  
For bang commands, this also removes the file even if it fails to  
email it)

printwiz a file with !ACROBAT /preview <data>

(creates a file and previews it under a temp name)

printwiz a file with !ACROBAT /x<> /pOutlook /n<> /j<> /a<> <data>

(create a file and forces it to go out MAPI)

printwiz a file with !FAX32 /x<> <data>



```

    {faxes with MS Fax}

printwiz a file with !PDF /f<filename> <data>
    {creates PDF file}

printwiz a file with !MAPISEND /pOutlook /rbee@anzio.com <data>
    (sends email via MAPISEND program and Outlook profile)

printwiz a file with !MAPISEND /pFaxing /f5036240760 <data>
    (sends fax via MAPISEND program and faxing Outlook profile)

printwiz a file with !MAPISEND /pOutlook /ebee@anzio.com <data>
    (sends email via MAPISEND program and Outlook profile)

printwiz a file with !PRINT /p<printer><data>!PRINT <data>!PRINT <data>
    (sends each job to printer as print job)

printwiz a file with !PREVIEW <data> !PRINT <data> !FAX32 /x<> <data>
    (previews until "Print" or "Cancel", sends next to printer and
    third to local fax)

printwiz /email=bee@anzio.com /n"note line 1|note line 2" /j"subject line" testfile
    (email using default email method)

printwiz /email=bee@anzio.com testfile

printwiz /email=bee@anzio.com /n"note line 1|note line 2" /j"subject line" test.txt
/ac:\printwiz30\test.doc
    (email using Outlook and "Outlook" profile from default email method
    Outlook)

printwiz /emailmapi=bee@anzio.com /emailprofile=Outlook /emailfromaddress=beepc@anzio.com
testfile /j"subject line" /n"note 1|note 2"

printwiz /emailoutlook=bee@anzio.com /emailprofile=Outlook /emailfromaddress=beepc@anzio.com
testfile /j"subject line" /n"note 1|note 2"

printwiz /emailsmtp=bee@anzio.com /emailserver=mail.anzio.com /emailfrom="Bee PC"
/emailport=25 /emailfromaddress=beepc@anzio.com testfile /j"subject line smtp" /n"note 1|note
2"

printwiz /emailcdo=bee@anzio.com /emailserver=mail.anzio.com /emailfrom="Bee PC"
/emailfromaddress=beepc@anzio.com testfile /j"subject line cdo" /n"note 1|note 2"

printwiz /fax=5036240760 /faxfrom="My PC" /faxfromnum=5035555555 testfile.txt
    (faxed using default fax method of "winfaxpro")

printwiz /faxmapi=5036240760 /faxprofile=Faxing /faxfrom="My PC" /faxfromnum=5035555555
testfile.txt /j"Subject line note" /n"body note 1|body note 2"

printwiz /fax32=5036240760 /faxfrom="My PC" /faxfromnum=5035555555 testfile.txt /j"Subject
line note" /n"body note 1|body note 2"

printwiz /fax32=5036240760 /faxcoverpg="\\Documents and Settings\Bobby Ezell\My
Documents\Fax\Personal Coverpages\faxcoverpage.cov" /faxfromnum=5035550099
/faxfrom=BeePCAlso /t"Bee Fax" /n"note 1|note 2" /vdocname="My Fax32 test" /j"subject line 1"
test.doc test.xls
    (fax32 with 2 files and a cover page with info on it)

printwiz /x5036240760 /faxcoverpg="\\temp\bovy.cvp" /faxfromnum=5035550099 /faxfrom=BeePCAlso
/t"Bee Fax" /n"note 1|note 2" /vdocname="My Fax32 test" /j"subject line 1" testfile
    (WinFax PRO with a single file to render, but a custom cover sheet)

```

to also include)

```
printwiz /pdf /fc:\temp\mytest.pdf testfile
printwiz /pdf /fc:\temp\mytest1.pdf,c:\temp\test3.pdf testfile test3
printwiz /pdf /view /fc:\temp\test.pdf testfile.txt
printwiz /pdf /view /fc:\temp\test.pdf,c:\temp\test3.pdf testfile.txt test3
```

If you do view and have more than 1 file to view, they both display in the same copy of Acrobat Reader. Check the Windows menu to see the files.

```
printwiz "c:\printwiz30\test3" "c:\printwiz30\testfile.txt" /pdf /email=bee@anzio.com
/fc:\temp\mytest3.pdf,c:\temp\mytest.pdf /log /emailcdo /emailserver=anzio.com
/emailfrom=beepc@anzio.com /j"my subject line" /n"my body of notes line 1|notes line 2"
(this takes 2 files, creates 2 output PDFs and will email it via the
CDO method with body text and subject)
```

These examples show how creative you can be with the Print Wizard engine itself in order to accomplish what you need.

## Appendix D WēPO Examples

WēPO allows you to push print jobs out to a remote user's printer through a web page by simply specifying the file to print. You can do this by including a reference to the ActiveX WēPO object or actually including the ActiveX component (a push button) on the page.

The example displays a user-defined button and includes some javascript demonstrating how to cause a "print" action to occur. Additional example documents are available at our web site.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 7.0">
<TITLE>ActiveX test</TITLE>
</HEAD>

<BODY OnLoad="javascript:buttonhide();">

<div>Hi World</div>

<OBJECT
classid="clsid:4CE8026D-5DBF-48C9-B6E9-14A2B1974A3D"
codebase="http://www.anzio.com/controls30/printwizocx.cab#version=3,0,0,0"
width=0
height=0
align=center
hspace=0
vspace=0
ID="Pwbuttonx1" VIEWASTEXT>
<param name="mainurl"
value="http://www.anzio.com/pub/printwiz-samples/features.txt">
<param name="caption" value="Print">
</OBJECT>

<input type=button value="my print" OnClick="javascript:buttonclick();">

<script language=javascript>
<!--
function click(){
var print;
alert('Going to change the caption');

document.Pwbuttonx1.caption='Wow!!!';
print = confirm('Going to print the sample without touching the button!!!');
if (print)document.Pwbuttonx1.Click();
document.Pwbuttonx1.caption='Double Wow!!!';
alert('Going to make it disappear!');
document.Pwbuttonx1.Visible=false;
alert('Going to make it reappear!');
document.Pwbuttonx1.Visible=true;
}

```

```
function buttonclick() {
document.Pwbuttonx1.Click();
}

function buttonhide() {
document.Pwbuttonx1.visible=false;
}

//-->
</script>
<div>&nbsp;</div>
<a href="javascript:click();">Hello World - Click here!!!!</a>

</BODY>
</HTML>
```

## Appendix E More Developer Examples

The Print Wizard engine, the Print Wizard DLL and the WēPO ActiveX module are all available for developers to incorporate into their products. Through our reseller program, special discounts can be provided.

### E.1. The Print Wizard Engine

The Print Wizard engine is easily scripted to do all the functions described in this document. Everything from faxing, to emailing to PDF generation and printing, can be added to any third party program or to any Windows scripting with little effort.

When run with the “quiet” switch, the Print Wizard engine will always return any error level codes to the executing program, allowing your program to run Print Wizard with no user interface visible to the user, and completely under your application’s control.

#### E.1.1. Script example

The example below uses a Print Wizard script to take a file and fax it to a specific fax number with no user interaction necessary. Using the examples for Print Wizard found throughout this manual, you can put the Print Wizard engine to use without needing to be an advanced programmer.

```
' Run this from Cmd as      wscript xxxxxx.vbs    for dialog messages
' or                        cscript xxxx.vbs     for console messages

set shell = CreateObject("WScript.Shell")

for each filename in WScript.arguments
  'Process each command-line filename

  set pw = shell.Exec("c:\printwiz30\printwiz.exe /q /x555-1212 " + filename)

  do while pw.status = 0
    WScript.echo "Print Wizard processing, processid is ", pw.processid
    ' current process id
    WScript.sleep 100
  loop

  WScript.echo "Print Wizard complete for file " + filename
  WScript.echo "  return code", pw.exitcode           ' PW Halt Code
  'WScript.echo "  program status code ", pw.status   ' PW status (0-running, 1-terminated)

next
```

## E.1.2. “Bang” command examples

By embedding “bang” commands inside your data stream, you can accomplish several print jobs with one data stream or file.

**BANGACROBAT.TXT** - *Generate a PDF – Depending on Acrobat driver, this may ask for a file name or place it with a default file name.*

```
!acrobat
sdfsda fsdfsdfsdfsdfsf
test of bang acrobat
```

**BANGACROEMAIL.TXT** - *Generate a PDF and then email it with an attached file included (this uses a call to MAPISEND to email the file).*

```
!acrobat /xabc@def.com /pOutlook /n"note line 1|note line 2" /j"subject" /ac:\printwiz30\test.doc
sdfsda fsdfsdfsdfsdfsf
sdfsda fsdfsdfsdfsdfsf
test of bang acrobat
```

**BANGACROFILE.TXT** - *Generate a PDF with a specific file name.*

```
!acrobat /fc:\temp\myacrobattest.pdf
sdfsda fsdfsdfsdfsdfsf
sdfsda fsdfsdfsdfsdfsf
test of bang acrobat
```

**BANGACROPREVIEW.TXT** - *Generate a PDF and then open it with the Windows default Acrobat Reader.*

```
!acrobat /preview
sdfsda fsdfsdfsdfsdfsf
sdfsda fsdfsdfsdfsdfsf
test of bang acrobat
```

**BANGFAX32.TXT** - *Fax the data using Microsoft’s Fax printer driver.*

```
!fax32 /x5036240760
dffddfffffffffffffffffffffffffffff
fffffffffffffffffffffffffffff
test of bang fax32
```

**BANGFAX32MULTI.TXT** - *Fax multiple faxes using Microsoft’s Fax printer driver.*

```
!fax32 /n"note line 1 and|note line 2" /x5036240760 /j"fax 1"
this is a multi-part file for fax32 #1
fax with no attachments but a default cover page
(cover page determined because of "/n")
blablabla
blablabla
!fax32 /aC:\TEMP\TEST.DOC /n"note line 1 and| line 2" /x5036240760 /j"fax 2"
```

```

this is a multi-part file for fax32 #2
fax with attachments and a default cover page
(cover page determined because of "/n" from fax32 default settings)
blablabla
blablabla
!fax32 /x5036240760
this is a multi-part file for fax32 #3
fax with no attachment and no cover page
blablabla
blablabla
!fax32 /x5036240760 /c"c:\temp\faxcoverpage.cov" /n" line 1 and| line 2" /j"fax 4"
this is a multi-part file for fax32 #4
fax with no attachment and different cover page
(overriding default cover page with '/c<name>')
blablabla
blablabla
!fax32 /x5036240760 /n"Fax cover page only #5|note line 2 and|note line 3" /j"fax 5"
!fax32 /x5036240760 /a"c:\temp\test.doc,c:\temp\test file.xls" /n" line 1 and| line 2" /j"fax 6"
this is a multi-part file for fax32 #6
fax with two attachment (one with an embedded space) and default cover page
blablabla
blablabla

```

**BANGMAPIEMAIL.TXT** - *Use MapiSend to email data using an Outlook email profile of "Outlook".*

```

!mapisend /pOutlook /eabc@def.com
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
cccccccccccccccccccccccccccccc
test bang mapisend email

```

**BANGMAPIFAX.TXT** - *Use MapiSend to fax the data using Outlook and a fax profile of "Faxing".*

```

!mapisend /pFaxing /f5036240760
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
cccccccccccccccccccccccccccccc
test bang mapisend faxing

```

**BANGPDF.TXT** - *Generate a PDF using Print Wizard's PDF generator and email it using the default email method defined in the Print Wizard default file.*

```

!pdf /fc:\temp\pdfname.pdf /email=abc@def.com
dsfsdfsda
test bang pdf

```

**BANGPDFMAPINOBODY.TXT** - *Generate a PDF using Print Wizard's PDF generator and then email it using MapiSend.*

```

!PDF /fc:\temp\mytest.pdf

```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbb
test bang PDF followed by !MAPISEND
!MAPISEND /p"Outlook" /eabc@def.com c:\temp\mytest.pdf
```

BANGPDFMAPIWITHBODY.TXT - *Generate a PDF and email it with MapiSend.*

```
!PDF /fc:\temp\mytest.pdf
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbb
test bang PDF followed by !MAPISEND
!MAPISEND /p"Outlook" /eabc@def.com c:\temp\mytest.pdf /n
The above file is emailed as an attachment and this is the body passed to MapiSend
```

BANGPDFSTARTRUN.TXT - *Generate a PDF and then view it.*

```
!PDF /fc:\temp\mytest.pdf
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbb
test bang PDF followed by !MAPISEND
!RUN start c:\temp\mytest.pdf
```

BANGPDFSTARTRUN2.TXT - *Generate a PDF and as a separate job, save the data after !RUN into a specified file and view it (%printtoname comes from command-line via Expand Variables routine, not from the !PDF line).*

```
!PDF /fc:\temp\mytest.pdf
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbb
test bang PDF followed by !MAPISEND
!RUN start %printtoname
VVVVVVVVVVVVVVVVVVVVVVVVVV
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
YYYYYYYYYYYYYYYYYYYYYYY
```

BANGPREVIEWPRINTFAX.TXT - *Do three operations from one file for the default printer.*

```
!preview
Aaaaaaaaaaaaaaaaaaaaaaaaaaaa
Bbbbbbbbbbbbbbbbbbbbbbbbbbbb
Last line of bang-preview
!print
Dddddddddddddddddddddddddd
Eeeeeeeeeeeeeeeeeeeeeeeeeeee
last line of bang-print
!fax32 /x5036240760
Mmmmmmmmmmmmmmmmmmmmmmmmm
sdfsdfsdfsdfsdfsdfsdfsdfsdfsdfs
cccccccccccccccccc
```





## E.2. The Print Wizard DLL

Like the Print Wizard engine, the Print Wizard DLL is usable from within any third-party program that supports calls to Windows style DLL modules. The Print Wizard DLL supplies these programs with direct control of printed output via the DLL calls, and includes support for all the Print Wizard features discussed elsewhere in this manual.

### E.2.1. Visual basic example

There are many exposed functions of the Print Wizard DLL. The first box below shows some of those functions, followed by a short example utilizing them. This can be done through many other languages as well (see our included examples and other examples at our web site).

```
Private Declare Function pw_configure_printer Lib "pwdll" (ByVal window As
    _ Integer) As Long
    'returns 1 if successful, else 0
Private Declare Sub pw_set_printer Lib "pwdll" (ByVal lpchar As String)
    'set printer by name
Private Declare Function pw_print_text Lib "pwdll" (ByVal lpchar As String, _
    ByVal count As Long, ByVal cs As Long) As Long
    'returns 1 if successful, else 0
    'cs is 0 for ISO
    '    1 for OEM
    '    2 for other (no translation)
    '    3 for Unicode UTF-8'
'WIDE VERSION ###
Private Declare Sub pw_flush Lib "pwdll" ()
    'terminates print job
Private Declare Sub pw_set_html_base Lib "pwdll" (ByVal lpchar As String)
    'sets the html_base for relative filenames
Private Declare Sub pw_set_preview Lib "pwdll" (ByVal b As Long)
    'b is 1 for preview on
    '    0 for preview off
Private Declare Sub pw_set_orientation Lib "pwdll" (ByVal b As Long)
    'b is 0 for portrait
    '    1 for landscape
    '    2 for auto
Private Declare Sub pw_set_font Lib "pwdll" (ByVal lpchar As String)
    'specify font name(s)
Private Declare Sub pw_set_overlay Lib "pwdll" (ByVal lpchar As String)
    'specify name of overlay
Private Declare Sub pw_set_copies Lib "pwdll" (ByVal n As Long)
    'set number of copies
Private Declare Sub pw_eject_page Lib "pwdll" ()
    'equivalent to printing a formfeed
Private Declare Function pw_set_output Lib "pwdll" (ByVal lpchar As String, _
    ByVal count As Long) As Integer
    'returns 1 if successful, else 0
    'sets type of output
    '    empty string for printing
    '    PDF://          Generate PDF file with default name
    '    PDF://<filename> Generate PDF file with given name
    '    FAX32://<phonenumber> Send fax
Private Declare Function pw_print_file Lib "pwdll" (ByVal lpchar As String, _
    ByVal count As Long, _
    ByVal cs As Long) As Integer
    'prints an entire file
    'returns 1 if successful
    'cs is charset as above
```

This example is for a button that when clicked will create a PDF file through the Print Wizard DLL.

```
Private Sub Command1_Click()
  ct = Len(Text2.Text)
  pw_set_orientation (1)                \ landscape
  res = pw_print_text(Text2.Text, ct, 0)
  If res <> 0 Then
    pw_flush
  End If
  If res = 0 Then
    MsgBox ("Function failed")
  End If
End Sub
```

## E.2.2. Delphi example

...

```
function pw_configure_printer(hwnd : THandle) : longint; stdcall;
  external 'pwdll.dll' name 'pw_configure_printer';
function pw_print_file(p : pchar; ct : longint; cs : longint):longint;stdcall;
  external 'pwdll.dll' name 'pw_print_file';
function pw_set_output(p : pchar; ct : longint):longint; stdcall;
  external 'pwdll.dll' name 'pw_set_output';
procedure pw_set_copies(n : longint);stdcall;
  external 'pwdll.dll' name 'pw_set_copies';
function pw_print_text(p : pchar; count : longint; cs : longint):longint;stdcall;
  external 'pwdll.dll' name 'pw_print_text';
function pw_print_text_w(p : pwidechar; count : longint):longint;stdcall;
  external 'pwdll.dll' name 'pw_print_text_w';
procedure pw_flush; stdcall;
  external 'pwdll.dll' name 'pw_flush';
function pw_get_printer:OleVariant;safecall;
  external 'pwdll.dll' name 'pw_get_printer';
function pw_get_version:OleVariant;safecall;
  external 'pwdll.dll' name 'pw_get_version';

var
  filename : ansistring;
  output   : ansistring;
  i        : integer;
  s        : string;
begin
  // Check file name
  filename := paramstr(1);
  if not FileExists(filename) then
    begin
      Writeln('File not found');
      Readln;
      exit;
    end;

  // Ask for a printer name
  s := pw_get_printer;
  i := pw_configure_printer(0);
  if i = 0 then
    begin
      Writeln('Configure printer failed');
```

```
        Readln;
        exit;
    end;
s := pw_get_printer;

// Check to see if we are writing to a file
output := paramstr(2);
if output <> '' then
    i := pw_set_output(pchar(output), length(output));

// Wiz print text
s := 'This is printed with the test program';
i := pw_print_text(pchar(s), length(s), 0);

// Others
pw_set_copies(2);
// Get printer name and version. Configure_printer will set this.
// If you do not do Configure_printer sometime earlier, this is left blank
// to print to the default printer.
s := pw_get_printer;
writeln(s);
s := pw_get_version;
writeln(s);

// Print a file
i := pw_print_file(pchar(filename), length(filename), 1);

// End this print job
if i <> 0 then
    pw_flush;
end.
```

## E.3. WēPO ActiveX

While the WēPO ActiveX module is normally used through web pages, it is also easily adapted to any third-party application as a standard ActiveX module, including support for most all the features described elsewhere in this manual.

The following example prints an entire sheet of pharmacy labels without ever having to display anything in the web browser. When someone pushes the “Print” button, *WēPO* “pushes” the print job out to the local browser’s printer, without needing to know anything about the printer.



The source for this is simple and straightforward:

```
<HTML>
<div align="center">
  <center>
    <table border="0" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" id="AutoNumber1" width="649">
      <tr>
        <td width="654"></td>
      </tr>
      <tr>
        <td width="654">
```

```

<h2 align="center"> <br>
<font color="#008080">Print Wizard ActiveX Test Page </font> </h2>
<h4 align="center"> Multi-image Pharmacy Labels Example</h4><p>
Click below to demonstrate Print Wizard push printing.
<p>
If the Print Wizard object is not installed on your machine, you will be asked to download
it. You
must have your security set to allow downloads from http://www.anzio.com for this to work.
<p>
This demonstrates printing a multi-image pharmacy label, with rotated text and barcode.
<center><P>
<OBJECT
    classid="clsid:4CE8026D-5DBF-48C9-B6E9-14A2B1974A3D"
    codebase="http://www.anzio.com/controls30/printwizocx.cab#version=3,0,0,0"
    width=100
        height=100
        align=center
        hspace=0
    >
<param name="mainurl" value="http://www.anzio.com/pub/printwiz-samples/rx-label.txt">
<param name="caption" value="Print">
    <param name="Cancel" value="0">
    <param name="Default" value="0">
    <param name="DragCursor" value="-12">
    <param name="DragMode" value="0">
    <param name="Enabled" value="-1">
    <param name="Font" value="MS Sans Serif">
    <param name="Visible" value="-1">
    <param name="DoubleBuffered" value="0">
    <param name="Cursor" value="0">
    <param name="licensecode" value>
    <param name="printersetup" value="0">
    <param name="charset" value>
    <param name="debug" value="0">
    <param name="initfile" value>
    <param name="orientation" value>
    <param name="duplex" value>
    <param name="fontname" value>
    <param name="overlay" value>
    <param name="bitmap" value>
    <param name="preview" value="0">
    <param name="faxnum" value>
</OBJECT>
    <p>&nbsp;</td>
</tr>
</table>
</center>
</div>
<H1> &nbsp;</H1>
</HTML>

```

Additional WēPO examples can be found at our web site.

## Appendix F Print Wizard Support Files

There are several files to be aware of in Print Wizard that may be used by the engine or by supporting programs. Here is a brief description of the file structures and files in use by Print Wizard:

printwiz.ini	Defaults file for PrintWiz and components (includes object names).
printwiz.log	The log file for PrintWiz and components debug messages. There may also be a separate log file for services if the user manually edits the default file.
master.profile (optional)	The master profile file to set to override global defaults if found.
pwui.job	When the UI fires off printwiz, it first saves all parameters as this file and passes them to printwiz as /profile=<> .
.<ext>.profile	If there is no profile specified, if the UseExtProfile option is true, and this file exists for the extension, this profile is used.
<name>.profile	The profile file containing the settings to use (overrides defaults).
pwservices.ini	The services file for Print Wizard and Win32 Services.
<servicename>.idx	This file contains a list of already printed files during the FTP Despool process, tracking them by file name and date/time stamp in order to prevent reprinting.

## Appendix G Print Wizard Services

### G.1. Program Calls

The default file "printwiz.ini" contains the defaults for what programs to call when the UI or a service fires off another program (see the section in the default file called "Externals").

```
ObjPrintWiz      // Location and name of Print Wizard
CmdDespool      // Print Wizard UI string for starting "despool"
CmdListen       // Print Wizard UI string for starting "listen"
CmdLpd          // Print Wizard UI string for starting "lpd"
SpawnDespool    // The command spawned by a "despool" or "ftpdespool"
SpawnListen     // The command spawned by a "listen"
SpawnLpd        // The command spawned by a "lpd"
```

Lpd queues also contain a default command for a specific queue, overriding the prior defSpawnLpd string (by default, the initial value is the same).

```
Spawn           // The command spawned for a specific "lpd" queue
```

If a profile is selected it may override the behavior of the action taken for a service.

PrintWiz (running in Listen or Despool mode), PWDESPOOL, PWLISTEN and PWLPD utilize the same pwservices.ini file for its service record. This is also the same as a Print Wizard Windows service would use. This includes for the called program, so be careful if you manually change any of these settings.

### G.2. Handling of work files

- A service of type “despool” has a directory to watch (or wildcard).
- A service of type “ftpdespool” has a remote directory to watch on an FTP server.
- All service types have an optional working directory to move work files to (the default is the user’s Windows "temp" directory).
  - If the service is type “despool” and no working directory is specified, we first try the user's Windows Temp directory. If this fails, we default to the watched directory and "saved", i.e. if we are watching \A then the temp becomes \A\saved.
- A service of type “LPD” has a Print Queue Directory, which is the working directory.
  - If the service is type LPD and no working directory is specified, first try the Windows Temp directory for the user. If this fails, LPD fails.



- Both the LPD and despool services have a collision flag whereby if a file exists, this flag states what to do. The action taken can be to "Overwrite", "Keep New / Rename Old" or "Keep Old / Rename New".

### G.3. FTPDespool Files

As with all other services, the actual settings for the service are included by “service name” in the pwservices.ini file, located in the Print Wizard home directory. However, there is one other file that is used for the FTP Despool service.

When you run a service of type “FTPDespool”, the file is fetched from the host FTP site and stored in a temporary location on your Windows PC. This is the actual file that is spawned off to the Print Wizard engine in order to print. The file is then deleted from the host if that option is set.

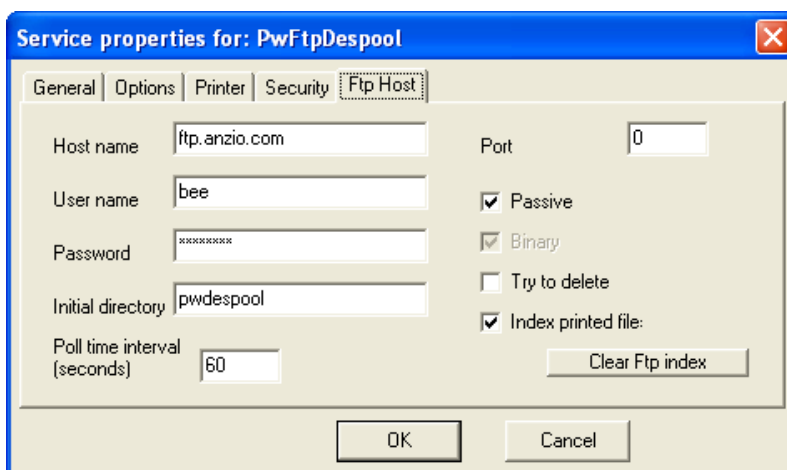
However, what if you do not have delete file permissions for your FTP host? If the file you are printing continues to exist on the host, you would end up in a loop, constantly reprinting the same file or files. And what if you needed to reprint a file?

To solve this problem, each “FTPDespool” service has an additional file in the Print Wizard home directory made up of the service name and the extension ‘.idx’. This file contains a single line entry for each file it has processed, whether it passed the file off to the Print Wizard engine, or whether it failed the services rules and was unable to print.

You can delete a line in this file and if the file still exists in the host despool directory, that file will reprint.

Files will not continuously print, but you do need to keep an eye on a couple of things in order to clean up after yourself. These can be done through some daily script or done manually on a regular basis:

- If the user does not have permission to delete host files, the system administrator should clear out the host FTP despool directory on a regular basis.
- You occasionally also need to clear out the “printed” index file so it does not keep growing. This should be done at least as often as you clear out the host files. This can be done manually through the Print Wizard user interface, or by simply deleting the file with the service name and the “.idx” extension, i.e. `c:\program files\printwiz30\pwftpdesspool.idx`



## Appendix H Windows Services

### H.1. Program Calls

The default file "printwiz.ini" contains the defaults for the Print Wizard services program, normally called "pwservices.exe". This string will be used when Windows services fires off the services program.

PwServices utilize the same pwservices.ini file as do other Print Wizard services, for its service record. This is also the same as a Print Wizard Windows service would use. This includes for the called program, so be careful if you manually change any of these settings.

### H.2. Handling of work files

- A service of type "despool" has a Directory to watch (or wildcard).
- A service of type "despool" has a Working Directory to move work files to (the default is the user's Windows "temp" directory).
  - If the service is type "despool" and no working directory is specified, we first try the user's Windows Temp directory. If this fails, we default to the watched directory and "saved", i.e. if we are watching \A then the temp becomes \A\saved.
- A service of type "LPD" has a Print Queue Directory, which is the working directory.
  - If the service is type LPD and no working directory is specified, we first try the Windows Temp directory for the user. If this fails, LPD fails.
- Both the LPD and despool services have a Collision flag whereby if a file exists, this flag states what to do. The action taken can be to "Overwrite", "Keep New / Rename Old" or "Keep Old / Rename New".

## Appendix I Log Files and Debug Information

### I.1. Debug Flags

There are numerous "debug" flags used throughout the Print Wizard programs. These "stack" similar to the way in which the default files do:

- On startup, defaults are loaded with a "debug" flag setting.
- A master profile may be loaded, overriding the default debug flag.
- A profile may be loaded overriding the current debug setting.
- If applicable, a service may be loaded overriding the current debug setting.
- If you are running Print Wizard engine from the UI (PWUI), there is a menu item for debug that may get passed as a command-line switch.
- Within the default file, the profile and the service, there are spawn commands for the Print Wizard engine, whereby they could be manually edited to include a command line parameter turning debug on or off.
- The Print Wizard engine itself has a command line option to show debug information on what it prints. This is done with the /debug or /vdebug command line switch. The additional information displayed gives the results of the Print Wizard analyze section, showing how it arrived at its results. This can be quite revealing of the data you received and what is actually in the data stream.

### I.2. Log Files

By default, the usual log file name of "printwiz.log" is used and is located in the same directory as the executables.

You can change the name of this file by manually editing the "LogName" field in the "printwiz.ini" default file.

If you are running a Foreground Service or a Windows service a different log file name may be used by manually changing the name in the "printwiz.ini" default file. Change the name of "LogSvc" to override the log file used for services. This affects the stand-alone programs, PwDespool, PwListen and PwLpd, as well as the Windows services program, PwServices.

Also in the default "printwiz.ini" file, there is a setting for "LogMax" which determines the maximum number of bytes that the file can be. Once the file gets to this size, lines of log entries will drop off, oldest first. By default this value is set at 20,480,000 bytes.

## Appendix J Print Wizard Superfiles

Print Wizard has the concept of “super files”. These are files for both input and output that may be unusual, and are referenced by a standard URI (Universal Resource Indicator), such as the URI for web pages. These file are usually referenced by a “://” as part of the file name, such as that for web pages, scan://.

When Print Wizard encounters one of these resource indicators as part of a source file name, Print Wizard will attempt to use it to locate the file or the source input device. Likewise, if Print Wizard encounters one of these resource indicators as part of the /f<filename> command (the output file), Print Wizard will attempt to write the file out to that URI.

At present Print Wizard supports the following “super file” types, though many may not be practical except under special circumstances:

URI	Source	Target
file://	√	√
wprn://name		√
wprn://		√
com://		√
lpt://		√
scan://	√	
pdf://filename		√
fax32://		√
emf://	√	

As an example, you can run Print Wizard to select a scanned item from the Windows default TWAIN-compliant scanner and output it as a PDF file by doing:

```
printwiz.exe /fpdf://mypdf.pdf scan://
```

## Appendix K Emailing

Currently Print Wizard has email support as follows. They all have their advantages, and may or may not be installed on your PC. You will want to play with them to see what works best for your Windows configuration.

All of these protocols require different items to successfully connect to an email server or gateway, and your email server or email gateway may not allow access from all of these variations.

Likewise, certain firewalls and routers, both on the local PC and on your network, may block some of these protocols.

- CDO

Collaboration Data Objects are automation compatible object models that wrap MAPI for easier use. Whereas MAPI requires some configuration, CDO handles this for you and makes some assumption to simplify sending and receiving of email, faxes and calendar events. We have implemented the sending of email within the Print Wizard engine.

- MAPI

The Messaging Application Programming Interface comes in various flavors with support for many 3<sup>rd</sup> party applications such as Outlook (MAPI), Outlook Express (Simple MAPI), Exchange (Extended MAPI), and others. Print Wizard implements standard MAPI with support for sending of email and the handling of Outlook profiles. MAPI is available on most all Windows platforms in some form or another, but it does assume user profiles and some initial setup have been performed.

- OUTLOOK

For those users with Outlook installed, Outlook provides object level interfaces direct to the program. Print Wizard has implemented the basic calls for sending email directly into Outlook, obeying any rules, permissions and profiles that Outlook has defined.

- SMTP

The Simple Mail Transport Protocol is the standard Internet protocol for transporting email messages. The implementation of SMTP by Print Wizard assumes that your Windows machine has permissions to send email through your local email gateway (using the email gateway as a relay). This may not always be the case, but usually if any of the above methods work, this will.

There are several command line parameters used for email (certain required items may be pulled from default email type):

```

/email=<to address>                (use defaults in printwiz.ini)
/emailcdo=<to address>
/emailfrom=<from name>              (required for all types)
/emailfromaddress=<from address>   (optional for all types)
/emailmapi=<to address>
/emailoutlook=<to address>
/emailpass=<password>              (optional for smtp)
/emailport=<port #>                 (optional for smtp)
/emailprofile=<Outlook/MAPI profile> (required for MAPI, Outlook)

```

```

/emailsmtplib=<to address>
/emailserver=<server> (required for CDO, smtp)
/emailuser=<user> (optional for smtp)
/a<attachmnt or CSV list of attachmnts> (optional for all types)
/n<notes or note include file> (optional for all types)
/j<subject line> (optional for all types)
/t<to name> (optional "to" name for all types)

```

## K.1. CDO

There are two different ways of using CDO. The first uses MAPI and the second uses SMTP calls wrapping around MAPI calls. They both go through the Windows standard routines to send email. Print Wizard implements CDO SMTP since it already has implemented MAPI elsewhere, and there really would be no difference (just in the call names).

Print Wizard CDO will take all files as attachments and email them over whatever default network protocol (usually TCP) that Windows recognizes. With this method you only need the Server name as the default port is set to the standard mail network port and authorization is assumed as anonymous.

- With anonymous and CDO, Outlook does not need to run to send email.
- Email attachments may come back with "raw format" on some email readers.
- Protocol failure occurs if the "To" or "From" addresses are badly formatted or b) this may include errors as a result of reverse DNS (domain name) lookups.
- Both Symantec and McAfee internet security (and especially the anti-virus portion) may see all attachments as viruses and you may have to "Yes" through several dialogs.
- Certain internet security suites, such as Symantec and McAfee may return an error if it cannot resolve the "To" and "From" address through DNS and reverse DNS lookups.

## K.2. MAPI

MAPI is the standard set of calls used by Outlook, Outlook Express, Exchange and others. There are several levels of MAPI including Simple, Standard, Extended and Exchange Server versions.

Print Wizard implements basic MAPI calls which should work with Simple MAPI (Outlook Express), Standard and Extended.

Print Wizard MAPI sends all files as attachments. It also requires a specific Outlook Profile depending on the "transport" application. The default transport application is set in the Windows Control Panel:Internet Options:Programs:E-Mail field. Default Profiles (not to be confused with Print Wizard print profiles) are set in the Control Panel:Mail:Show Profiles dialog. If you are using Outlook Express or Netscape Email, default settings for the profile are assumed and sometimes difficult to figure out (more on this later.)

When emailing via the MAPI protocol, you are dependent on two major items:

- 1) Security software may prohibit or at least warn you before allowing a third-party program, such as Print Wizard, from sending email.
- 2) Outlook settings determine if and when the email gets sent. In typical setups, you have to bring up Outlook and tell it to Send/Receive before they actually leave the PC.

Other specifics about MAPI

- Profiles seem to be case-sensitive under certain circumstances.
- Setting up a MAPI profile for faxing requires work.
- Can't include leading or trailing spaces in attachment names (embedded are ok).
- Will not send if Outlook set to wait for user to do something ("Send/receive all") unlike Outlook, which sends right away.

### **K.3. Outlook**

For those PCs with Outlook installed, Outlook allows third party programs access through a series of COM object calls directly into Outlook. While this works more smoothly than MAPI, it may also be restricted by Security software and user permissions. The advantage is that in a typical Outlook setup, the email is sent out immediately by the COM objects without waiting for user intervention.

Like MAPI this will require an Outlook profile be already set up. This also does not work if using Outlook Express or Netscape Mail.

Other specifics about Outlook:

- Print Wizard may actually bypass asking for User Profile by doing a login session if allowed.
- While MAPI may wait for a user to force Send/Receive, this does not, it sends right away.

### **K.4. SMTP**

Print Wizard does have support for direct calls to a mail server over a network running simple Internet email. This is done through the SMTP setting where you do need the server name, the port to relay through, and you may also need a user name and password for authentication. This is typically the same as you would see from a UNIX/Linux host, accessing mail over IPOP, POP2, POP3, IMAP or some variation of remote email.

SMTP is straightforward but may not work in all environments, i.e. Microsoft Exchange servers may require the client be a MAPI or Outlook client, and Domino Notes and others may require their own variation of email, blocking SMTP relaying.



Also some personal firewalls may prevent SMTP from sending direct over port without adding Print Wizard to the "Acceptable Applications" list.

## Appendix L Faxing

Print Wizard currently has fax support for the following:

- Fax32

Windows faxing is referred to here as Fax32. Windows XP, Windows 2000 and Windows 2003 support this type of faxing without the need to purchase additional software. While it may not be installed initially ("Fax" should show as a printer driver), you should be able to install it from your Windows CD.

- MAPI

The Messaging Application Programming Interface comes in various flavors with support for many third-party applications such as Outlook (MAPI), Outlook Express (Simple MAPI), Exchange (Extended MAPI), and others. Print Wizard implements standard MAPI with support for sending of faxes based on the profiles default transport mechanism being set for faxing.

- WinFax PRO

Symantec's WinFax PRO supports many object calls into their software for the control of faxing through third-party applications. Print Wizard implements a basic set of calls to do WinFax PRO faxing.

There are several Print Wizard command line parameters used for faxing (certain required items may be pulled from default fax type):

```

/fax=<to number>                (uses defaults from printwiz.ini)
/fax32=<to number>
/faxcoverpg=<cover page file>   (optional for Fax32)
/faxfromnum=<originating fax #> (optional for all types)
/faxfrom=<originating sender name> (optional for all types)
/faxprofile=<Outlook/Mapi profile> (required for faxing with Mapi)
/faxmapi-<to number>
/faxwinfaxpro=<to number>
/j<subject>                     (cover page subject)
/n<notes>                       (cover page notes or include file)
/t<to number>                   (optional "to" name for all types)

```

### L.1. Fax32

Back in Windows 95, Microsoft released a fax engine that would allow applications, such as Word scripting, VBA (Visual Basic for Application) and Word itself, to fax. They then dropped this support until Windows 2000 came out. Fax32 may not be installed by default on Windows 2000, 2003 or XP but can be added as a Windows component.

The new version of Fax32 is supported by Print Wizard both as a "superfile" (print to a file named fax://...), and as a fax target.

Currently, from the command line you can direct output to a “print-to-file” name of fax32:// or you can specify the current "target" type as fax32 (/Fax32=<faxnumber>).

- In the case of the "superfile", fax32://, the current files to process get handled one at a time as a write to the printer driver for faxing. This does assume what you are faxing is a file type that Print Wizard can handle.
- In the command line option, all files are passed to fax32 as attachments and ran thru a TIF file creation routine instead of writing them to the printer driver directly. In this manner, Print Wizard can handle multiple attachments of any file type (as long as there is a “file association” application).
- If a cover page name is specified in the CoverPage field of Fax32 in the Print Wizard default file (printwiz.ini), then a cover page is looked for by that name. Otherwise we obey the UseServerCoverPage option set in the fax printer driver. In this case, the driver looks for a “.cov” file as specified in the registry.

CSIDL\_PERSONAL\Fax\Personal Coverpages\ directory  
 i.e.  
 C:\Documents and Settings\User\My Documents\Fax\Personal Coverpages  
 or in the case of a file server in  
 \SERVERNAME\FAX\\$

If no name is given but “include cover page flag” is set, then MS Fax looks in  
 CSIDL\_APPDATA\Microsoft\Windows NT\MSFax\coverpg.cov  
 or  
 \\SERVERNAME\FAX\$\Coverpg.cov

When using the command line parameter to fax with Fax32, all files are handled as attachments, including regular file names, so some file association must exist for the file to get rendered. We then merge all these TIFs into one and re-pass it back to Fax32.

## L.2. MAPI

MAPI is the standard set of calls used by Outlook, Outlook Express, Exchange and others. There are several levels of MAPI including Simple, Extended and Exchange Server versions. Print Wizard implements basic MAPI calls, which should work with Simple MAPI (Outlook Express) and Extended (Outlook).

In order to fax with MAPI, you must set up a profile that uses the “fax transport” method rather than an email transport. You must then reference this as the profile to use when faxing.

Starting with Print Wizard, MAPI sends all files as attachments, letting the fax transport render them as necessary. The default transport application for faxing is normally Windows Fax32 in the case of Windows 2000 or later. However, this can be overridden if some other fax transport is loaded, such as WinFax PRO. Check your Outlook settings.

When faxing via the MAPI protocol, you are dependent on two major items:

- 1) Security software may prohibit or at least warn you before allowing a third-party program, such as Print Wizard to fax.
- 2) Outlook settings determine if and when faxes get sent and may also require recipients have fax numbers in some cases.

MAPI fax support in Print Wizard takes all files as attachments, no writing to the printer. So a file association must exist for the file extension.

### L.3. WinFax PRO

WinFax PRO from Symantec supports several third-party calls into its objects. Print Wizard uses these calls, writing to the fax device. These calls should work with WinFax PRO version 9.6 (from Delrina, though the object name may need to be manually changed inside the printwiz.ini file), and with version 10.0 or something later than version 10.03.

When Symantec took over support, they determined that there was too much support necessary with the third-party object calls. Hence support was dropped in version 10.01. Due to numerous complaints, in 10.02 some were put back in, but full support was not restored till 10.03. This latest version has to be ordered from Symantec tech support (tell them you are using a 3rd party program to fax through WinFax PRO and they will send a CD). As of November 2004, this version should be shipping as the "Windows XP Version".

WinFax PRO support in Print Wizard does support writing to the printer device and then rendering attachments separately. This works fine provided the file name is capable of being handled by Print Wizard. Otherwise the file should be passed as an attachment.

Cover pages are handled with the command line parameter

```
'/faxcoverpg=<>'
```

or by the variable in the profile file (it must be a complete path).

Otherwise a QuickCoverPage flag inside the WinFax PRO setup is used. QuickCoverPage can include a logo and a file of text to show, but the general layout cannot be changed. Any textual change of this needs to be done through the CoverPage Editor included with WinFax PRO.

Custom cover pages can be developed within WinFax PRO Cover Page Editor, but watch for the variables.

Including only a notes line will cause Print Wizard to automatically use a QuickCoverPage, otherwise you should specify a cover page file.

## Appendix M Windows Default Printer Issues

There is the idea in Windows of a default printer. You can determine your default printer through the Control Panel:Printers and Faxes. Additional information can be found at Microsoft's web site.

As this pertains to Print Wizard, there are some concepts that may help you understand the behavior of Windows and Windows printing a little better. This is a brief discussion of the issues surrounding printing with Print Wizard.

When Print Wizard and any of its supporting programs need to print, and no printer is specified, the logged-in user's default printer is automatically used. This is fine for most things, and causes no problems.

When a Windows system is first brought up, a "default user" name is used to allow you to load and install software. Prior to Windows XP Service Pack 2, this user was, by default, called whatever name you put in and given full access to everything.

Certain other Windows applications also had to run under some user and would inherit the permissions of the default user, making that user's printer their defaults also. Most of these other applications would run under the SYSTEM or LOCALSYSTEM user name. This includes things done at startup and prior to a user logging in, such as Windows services, Windows disk scans, some anti-virus and anti-spam software, and a number of other applications.

If a new machine is brought up, or an older machine is upgraded from a prior version of Windows to Windows XP, the default printer for LOCALSYSTEM is now the same default printer as the user. However, if the user changes their default printer, this does not necessarily change the default printer for these other system users.

Where this really comes into play is with Print Wizard Windows services. If PwServices is installed by the default /INSTALL, there is no USER name associated with this. Instead the default installation user is LOCALSYSTEM. The problem is that this may have defined a different printer than one may expect. As an example:

- a) If you were running on a prior version of Windows and then install XP or XP SP2, the current printer becomes the LOCALSYSTEM default printer as well.
- b) The first printer installed on Windows becomes the users default as well as the printer for LOCALSYSTEM.
- c) If the user then changes default printers, the LOCALSYSTEM printer may still be set to the original default printer from a) or b) above, and not print where expected.
- d) If that printer is deleted, then there may in fact be a problem - no default printer will exist for LOCALSYSTEM, and no output will be done (the user may never know why, if this is from Windows service).

The best manual procedure available involves deleting the old printer first, then installing a new printer as the default. Of course typical users do not want to have to do this.

So, when you create Print Wizard service records, you should always verify that it will print on a specific printer and not the default printer, when setting it up through the user interface.

For the curious, here are the registry entries where you can check this out if you are having problems:

The user registry entries for default printer are:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\  
  Current Version\Windows  
  "Device" (default)  
  "Devices" (list of what printers are available to Windows users)
```

The system defaults registry entry (LOCALSYSTEM and other system Users)

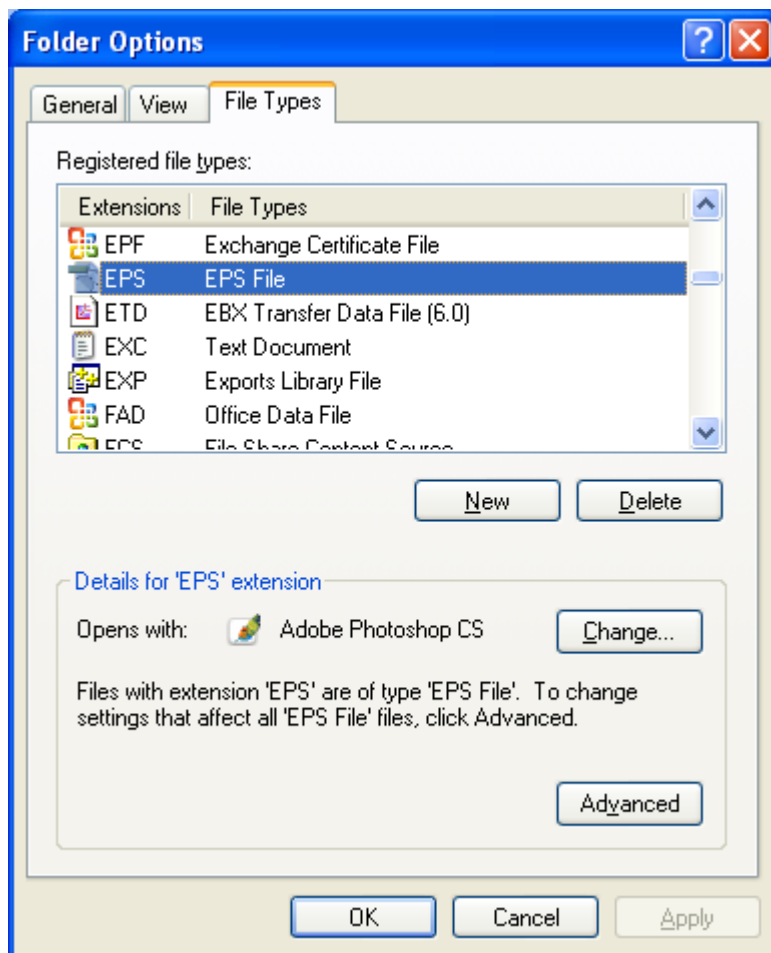
```
HKEY_USERS\.\DEFAULT\Software\Microsoft\Windows NT\  
  Current Version\Windows  
  "Device" (current default for these type of users)  
  "Devices" (list of what printers are available to these system users)
```

## Appendix N Windows File Associations & Smart Print

### N.1. File Type Associations

Windows uses File Associations and we take full advantage of this in several areas of Print Wizard.

File associations are handled by either their internal file type, such as with compressed and “zip” files, or by their file name extension, and associate a specific application to that file name. You can adjust, change and remove support for application to file associations through the Folder Options in Internet Explorer or through the Control Panel:Folder Options:File Types.



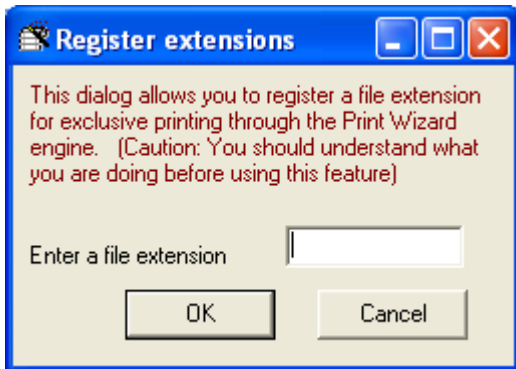
This association determines what application should be used for opening, printing and previewing this particular file.

Certain file types have no association at all, and hence cannot be opened by file name alone. You must first open an application in these cases.

While Print Wizard works with its own set of file extensions, it does not register any particular extension as needing to be printed every time by Print Wizard. However, you can request that a file association be



set up for any file type you wish. From the Tools: “Register a file extension” inside the Print Wizard user interface, you can add a single extension at a time for Windows to associate printing of that file extension with Print Wizard.



The “print” and the “printto” actions are the only actions for the file extension that will be updated/added.

## N.2. Smart Print

Print Wizard also has the concept of “Smart Print”. When Smart Print is turned on (checked in the Edit menu of the Print Wizard user interface), any file can be printed through Print Wizard as long as a File Association exists.

Within the Print Wizard default file, `printwiz.ini`, there is a list of file extensions, “FileExt”, that Print Wizard can handle. This, along with the Smart Print flag, determines how Print Wizard will print a file:

- If Smart Print is turned on, and an extension is in the list, Print Wizard will handle the file printing itself. Otherwise:
  - If a printer is specified, Print Wizard will try to do a “printto” action.
  - If a printer is not specified or the printer name is set to use the default printer, Print Wizard will attempt to do a “print” action.
  - If “preview” is requested, Print Wizard will try to do an “open” action.
- If Smart Print is turned off, Print Wizard attempts to handle all files. This can be a problem if Print Wizard does not understand the file type, such as printing a Word document through Print Wizard.

## Appendix O Application Notes

### O.1. Print Wizard and FilePro

Print Wizard includes two printer definition files for use with FilePro (a database system). They are:

- PRINTWIZ.PRT           For use with monochrome printers
- PRINTWZC.PRT         For use with color printers

In FilePro a PRT file defines a “printer type”. It contains some general information about the printer, some initialization and termination codes that are used in general, and some “printcode entries”. Printcode entries are identified by number. Each printcode entry consists of a control sequence to send to the printer and a description. A programmer designing a report can include printcodes in the report. At the time the report is printed, the user will by some means have chosen a printer type, which corresponds to a PRT file. The FilePro system uses the PRT file to translate each printcode into the control sequence that is appropriate for that printer. This achieves a certain amount of device independence, assuming there is consistency in what function is assigned to what numbered printcode.

The Print Wizard PRT files translate these printcodes to PWML sequences. For instance, printcode 5, labeled “Underline on”, sends out the sequence “<U>”, which Print Wizard understands to mean “underline on”. The printcode numbers used in the Print Wizard PRT files are synchronized where possible with laser printer printcode files delivered as part of the FilePro product.

When the user selects Print Wizard as their printer type, FilePro will render each report in PWML. Now as long as that print job goes through Print Wizard in its way to the printer (either standalone Print Wizard or the Print Wizard logic built into AnzioWin), it will be printed correctly on virtually any printer that Windows supports.

#### O.1.1. Installation and use – Windows

To make the Print Wizard definition files available in Unix/Linux-based FilePro, copy one or both PRT files into the “lib” directory of your “fp” directory tree. For instance, into “c:\fp50\fp\lib”.

To use, select “printwiz” or “printwzc” as your printer type in “Printer Maintenance”.

To route the printer output through Print Wizard, use one of these methods:

- Configure FilePro to print to disk. Set the PFPOSTPRINT environment variable to execute the appropriate command after closing the file; that command would be to run PRINTWIZ.EXE with the necessary parameters.
- Configure FilePro to print to disk, in a particular directory. Have a copy of Print Wizard running in the background, in despooler mode, watching that directory, and printing and deleting any files that are placed there.

#### O.1.2. Installation and use – Unix/Linux

To make the Print Wizard definition files available in Unix/Linux-based FilePro, copy one or both PRT files into the “lib” directory of your “fp” directory tree. For instance, into “/appl/fp/lib”.

Make sure the file name is in all lowercase.

To use, select “printwiz” or “printwzc” as your printer type in “Printer Maintenance”.

There are many ways, in this environment, to route the printer output through Print Wizard. Following are some brief descriptions:

- Use AnzioWin as your terminal emulator, and tell FilePro to do passthrough print by setting the PFPT environment variable.
- Define a “remote printer” to Unix/Linux, which tells it to use the LPD protocol. Then run PWLPD to receive and process the data, on the same or a different PC.
- Route the data through the netprint program, to Print Wizard (freestanding) running on a PC.
- Have FilePro print to file, in a particular directory that is visible to the PC via Samba, Vision FS, etc. Run Print Wizard (freestanding) on a PC, in despooler mode, watching that directory.

### O.1.3. Usage notes

Once everything is installed and working, you can start making creative use of the Print Wizard definitions. Most printcodes are self-explanatory. Following are some general notes on using the printcodes defined for Print Wizard:

- The general printer initialization code is #3. This leaves Print Wizard in LEGACY mode, with a ½” left margin.
- Codes 55 to 58 can be used to initialize particular print jobs to certain orientation and line spacing.
- To include a bitmap inline, do #75 (Start bitmap), then the name of the bitmap file, then #76 (End bitmap). The bitmap file can be in any of the formats Print Wizard supports (.BMP, .GIF, etc.). It is wise to include an entire pathname.
- If you don’t want to include the extension (“.GIF”) when specifying a bitmap, use #68 (Initialize bitmaps) once at the start of the print job. Alter this code as needed.
- You could have a report that specified a bitmap inclusion as “logo.gif”. Then, before printing, your processing could copy a specific file into “logo.gif”. This would have the effect of changing which bitmap got printed, without changing the report specification.
- To print barcodes, do #72 once at the beginning of the print job, to establish the style and size of the barcode. You can alter this code as needed. Then, for each barcode, do #73 (Start barcode), then the data to be bar-coded, then #74 (End barcode).
- Box drawing will work very well.
- Several special characters, such as the Euro, are provided.

Of course you are free to modify the PRT files as you like, with these tips and warnings:

- The editor built into FilePro (at least in some versions) will mess up files with lines that are too long, such as these. Use a freestanding editor such as vi or notepad.
- Any embedded space must have a backslash before it.
- Initialization codes leave Print Wizard in LEGACY mode. Any following Print Wizard tag must be preceded by a trigger character control-Z, represented as “^Z” in this file.
- After the first PAGESIZE tag (such as in #3), you are in LEGACY mode. To have another PAGESIZE tag, it must immediately follow (in printing sequence); it must take you OUT of LEGACY mode, then do the PAGESIZE tag, then go back into LEGACY mode. This is demonstrated in #55.
- It would be possible to add or modify a code with a PAGESIZE tag to include an OVERLAY parameter. Then, if the named file existed, it would be used as a form overlay. If the file did not exist, no overlay would be used. Other coding could be used to create or copy, and delete, the named file.
- The BASE tag in #68 could be changed to include an HREF, which would be the assumed directory location for bitmap files.

Finally, if you are not getting the output you expect, you can do these things:

- Turn on the debug option in Print Wizard.
- Have FilePro print to a file, and then open that file in an editor to see what is there. Study the PWML code in relation to this manual. Try editing the file and running it through Print Wizard. In general, see a) what needs to be generated in this file to get the printout you want, and b) what you need to do in FilePro in order to put the right thing in the file.

## O.2. Print Wizard and III Millennium and Innopac

Millennium and Innopac are products from Innovative Interfaces Inc., for the library industry. Print Wizard can be used to enhance, correct, adapt, and otherwise manipulate the print jobs coming out of these packages.

### O.2.1. Connecting to Innopac via attached print

Innopac is a character-based application. Library staff can use AnzioWin as their telnet client to run all the functions of Innopac. To print in this environment, simply tell Innopac to use “attached print” (or something similarly named), that prints through the terminal session to the attached printer. In AnzioWin, turn on Print Wizard, and select the printer.

### O.2.2. Connecting to Millennium or Innopac via Workstation JetDirect

Millennium is a Java-based system, so a telnet client is not needed. For this system, and also for Innopac, III has designed the “Workstation JetDirect” (WJD) printer type. When you tell Millennium or Innopac to print to a WJD printer, it sends the data using the JetDirect protocol, to either a) a specific IP address, or b) the same IP address as the client connection is coming in from. The latter is more popular, resulting in print data coming back to the user’s PC, for processing by Print Wizard.

What III calls the Workstation JetDirect protocol, Print Wizard calls “listen mode”. For this mode to work, both players must use the same “socket”; both default to socket 9100. To get started, on your PC run the command:

```
printwiz /listen /L /debug /u /preview
```

This command can be entered in a “DOS window”.

This runs Print Wizard in listen mode, with logging and debugging information going to the screen. The “/u” specifies the UTF-8 character set (see below). Print preview is enabled. You will now be able to see what Print Wizard receives.

Now in Millennium or Innopac, start a print job that goes to a WJD printer. It should be received by Print Wizard on your PC, resulting in some information appearing in the Print Wizard window, and the Print Preview window appearing.

In Millennium, you can configure each login to have four printer types: standard, receipt, label, and form. Each of these can be tied to a system-attached printer, an email address, or a WJD printer. Note that you could go so far as to configure each printer type to go to a different WJD printer, each configured to use a different port (9100, 9101, 9102, 9103). Then you could run four instances of Print Wizard on your PC, each configured for a different port, and each configured to use a specific printer.

Once everything is working, you could cause Print Wizard to be started up automatically by Windows, by putting a shortcut in the Startup group.

### O.2.3. Character sets

Innopac/Millennium is able to deal with international character sets, including diacritics and CJK. When it goes to print some output, it checks the printer's "diac map" setting to determine if or how to output non-ASCII characters. You can change this setting to change how international characters are output. Then you would want to match Print Wizard's (or AnzioWin's) character set configuration to match.

The most inclusive setting is Unicode UTF-8. This encoding can contain all the MARC diacritics, Chinese, Japanese, Korean, Cyrillic, etc. So we advise setting the "diac map" to UTF-8, and configuring Print Wizard or AnzioWin the same.

If the Innopac/Millennium diac table is set to "HP" or some other sets, the data from the host could contain escape codes. This might throw Print Wizard into Spooler mode, which is not good. Your debug information will tell you if this is occurring.

### O.2.4. What Print Wizard Will Do

- Once you have Print Wizard processing the print jobs from the III software, Print Wizard will do some things for you automatically: Although III does not output form feeds between pages, Print Wizard will figure out how many lines of text go on each page. (If Print Wizard has trouble doing so, configure III to use 66 lines per page.)
- Full-page reports will be auto-fitted to the page.
- Diacritics, CJK, and other international characters will be printed properly, with automatic font selection. Short output, such as labels and receipts, will trigger the mini-page logic. Print Wizard will tell Windows to make a custom page size, which will be just as long as is needed, so there will be no excess paper feeding at the end of the job.

Beyond this automatic handling, it is possible for you to modify the output format, by use of a print initialization file (as explained elsewhere). For this environment, you might want to:

- Adapt III's single-column spine label format into multi-column full sheet label stock, for use in laser printers.
- Add text and/or logos to checkout receipts.

Adjust margins for reports that will be bound.

## Appendix P Expandable Variables

In several places, Print Wizard can handle a variable name to actually represent an internal string within Print Wizard. Under certain circumstances, when Print Wizard sees the “%<string>” (percent sign followed by a variable name); that percent string will be replaced by the value of the internal string.

This happens in select places within Print Wizard in order to allow the user more control over what gets passed to spawned procedures. Places this is used:

- !RUN START <parameters>           Expand the parameters for a “start” run
- !RUN <executable> <parameters>   Expand the parameters for an execute
- !MAPISEND <parameters>           Expand parameters passed to the MapiSend program
- PRINT BYCUSTOM                   Expand when printing with a third-party program
  
- LPD Custom Spawn                   A special case where parameters for a third-party program can be filled with certain strings passed by the host LPR client

There are really two different procedures, one for the general “Expand Variables” and one covering the “LPD Variables” (we do hope to consolidate all this in the future and expand the use of these capabilities).

Expand Variables Procedure (most are as passed in the Profile or on the command-line):

The variable	The substitution (Default unit)
%1	TheFileName (current file)
%profile	ProfileName
%svcname	SvcSettings.srName
%printer	ProfileSettings.prPrinter
%filenamesex	TheFileNamesEx (file names w/ path)
%filenames	TheFileNames (as would be on CLP)
%filename	TheFileName (for current file)
%printtonamesex	ThePrintToNamesEx (print names w/ path)
%printtonames	ThePrintToNames (as would be on CLP)
%printtoname	ThePrintToName (for current file)
%attachmentsex	TheAttachmentsEx (attachments w/ path)
%attachments	TheAttachments (as would be on CLP)
%faxnumber	ProfileSettings.prFaxTo
%docname	ProfileSettings.prDocName
%email	ProfileSettings.prEmailTo

LPD Variables that can be passed from the host printing program:

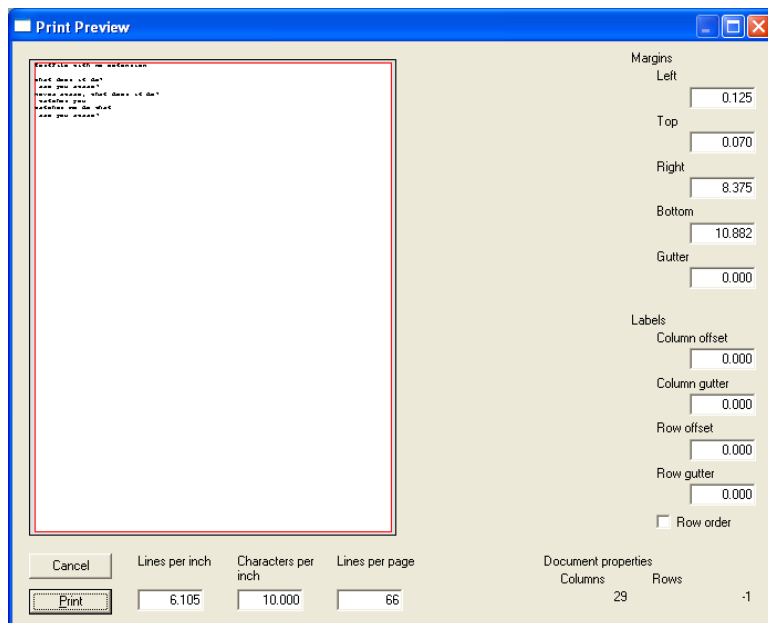
The variable	The substitution (Default unit)
%J	"jobname" from host
%T	"title" from host
%U	"user" from host
%M	"mail" (email address) from host
%C	"class" from host
%H	"host" from host
%1	"fullname" filename from host

## Appendix Q Print Preview, Labels, Ink & More

The capabilities of Print Wizard's Preview continue to expand. From the Print Preview, you can adjust printing issues such as margins and offsets on the fly, you can add Ink to your printout, such as a signature or highlighting, you can even select which label to start printing on when working with sheet labels.

### Q.1. Drag to Resize

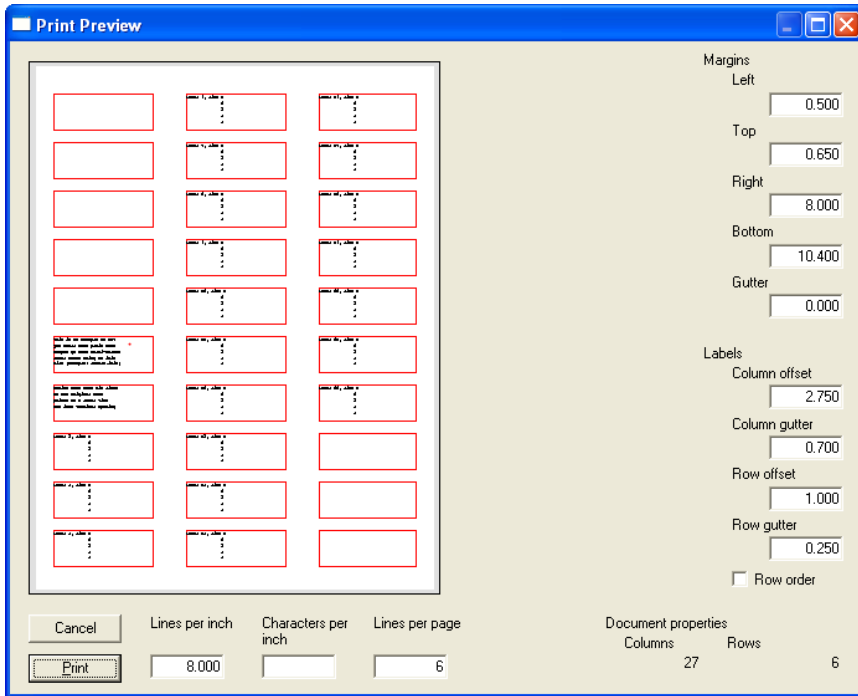
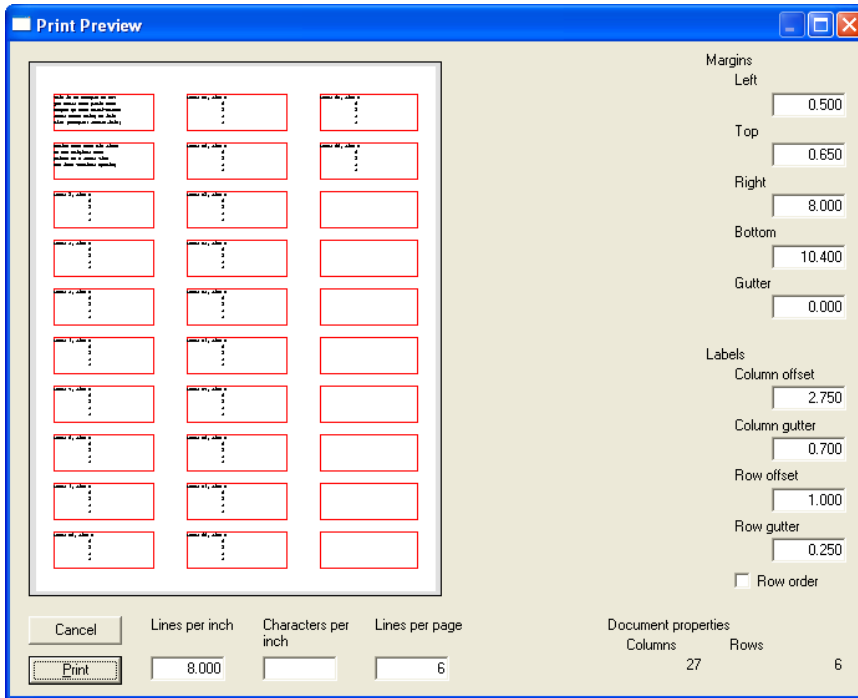
By clicking on an edge or corner of the dialog, you can drag and expand or shrink the Print Preview windows, allowing you to better see the formatting.



This is actually true of most Print Wizard dialogs with text boxes on them. You can click in the edges and corners and expand the dialogs to see more information.

### Q.2. Labels

Print Wizard's Print Preview also supports labels in a new way. Once you have the labels laid out for proper printing, you can then go to the Print Preview and actually select by right clicking with the mouse on the desired start location. The first label will adjust to start your printing at the selected location, adjusting the rest of the print job as you go.







## **Appendix R Troubleshooting**