



# **PDF417 Encoder**

**Version 2.2.1 for V7R1M0**

# **AS/400 Manual**

Silver Bay Software LLC  
100 Adams Street  
Dunstable, MA 01827  
Phone: (800) 364-2889  
Fax: (888) 315-9608  
[support@silverbaysoftware.com](mailto:support@silverbaysoftware.com)

Document Version 20120111

The information in this manual is subject to change without notice and should not be construed as a commitment by Silver Bay Software LLC. Silver Bay Software assumes no responsibility for any errors that might appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Products or brand names used herein are trademarks or registered trademarks of their respective companies

Copyright © 2009, 2012, Silver Bay Software LLC.  
All rights reserved.

# Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	CONTENTS OF THIS MANUAL.....	1
1.2	CONTENTS OF THE DISTRIBUTION CD-ROM.....	1
1.3	NAMING CONVENTIONS .....	2
<b>2</b>	<b>UPGRADING AN EXISTING ENCODER INSTALLATION.....</b>	<b>4</b>
<b>3</b>	<b>INSTALLING THE ENCODER .....</b>	<b>5</b>
3.1	REQUIREMENTS.....	5
3.2	INSTALLING THE ENCODER LIBRARY .....	5
3.2.1	<i>Installing from the CD-ROM .....</i>	<i>Error! Bookmark not defined.</i>
3.2.2	<i>Installing from Save Files.....</i>	<i>Error! Bookmark not defined.</i>
3.2.3	<i>Contents of the Installed Encoder Library.....</i>	6
3.3	PREPARING THE ENCODER FOR USE .....	7
3.3.1	<i>AFP Fonts.....</i>	8
3.3.2	<i>HP-PCL Fonts .....</i>	9
3.3.3	<i>Xerox Fonts.....</i>	9
<b>4</b>	<b>BINDING TO THE ENCODER.....</b>	<b>10</b>
4.1.1	<i>ILE Programming Environment .....</i>	10
4.1.2	<i>OPM Programming Environment.....</i>	10
<b>5</b>	<b>PRINTING.....</b>	<b>11</b>
5.1	PRINTING USING DDS .....	11
5.2	PRINTING USING HP-PCL FONTS .....	13
5.3	PRINTING USING XEROX FONTS.....	13
<b>6</b>	<b>SAMPLE PROGRAMS.....</b>	<b>14</b>
6.1	SAMPLE PRINTER FILE.....	14
6.2	COBOL SAMPLES .....	14
6.2.1	<i>Compiling and Running the Sample COBOL/400 Program.....</i>	14
6.2.2	<i>Compiling and Running the Sample ILE COBOL Program .....</i>	15
6.3	RPG SAMPLES.....	15
6.3.1	<i>Compiling and Running the Sample RPG/400 Program .....</i>	15
6.3.2	<i>Compiling and Running the Sample ILE RPG Program.....</i>	15
6.4	ILE C SAMPLE.....	16
6.5	OUTPUT OF THE SAMPLE PROGRAMS.....	16

# Silver Bay Software LLC PDF417 Encoder

Version 2.2.1 for V7R1M0

## AS/400 Manual

### 1 Introduction

#### 1.1 Contents of this Manual

This manual accompanies the Silver Bay Software LLC PDF417 Encoder version 2.2.1. It describes the aspects and operations of the encoder that are specific to the AS/400 (a.k.a. “System i” environment. Specifically, this manual covers:

- Installing and upgrading the encoder
- Steps required to prepare the encoder for use
- The sample programs that accompany the encoder

The actual programmer’s interface to the encoder is described in an accompanying manual.

#### 1.2 Contents of the Distribution CD-ROM

The distribution CD-ROM for the encoder contains the following files:

In the root directory:

File	Contents
readme.pdf	A brief file describing the contents of the CD-ROM

In the “savefiles” subdirectory:

File	Contents
PDF221S41.SAVF	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V4R1M0
PDF221S52.SAVF	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V5R2M0
PDF221S61.SAVF	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V6R1M0
PDF221S71.SAVF	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V7R1M0

In the “docs” subdirectory:

File	Contents
PDF_AS400.PDF	This manual.
PDF_API.PDF	An Adobe Acrobat file that contains the description of the API's supported by the encoder. Information for programmers is contained in this document.

In the “fonts/hp” subdirectory:

File	Contents
HPPL3309.FLJ	HP-PCL 3x9 Landscape-mode font
HPPL3412.FLJ	HP-PCL 4x12 Landscape-mode font
HPPL3309.FLJ	HP-PCL 3x9 Portrait-mode font
HPPL3412.FLJ	HP-PCL 4x12 Portrait-mode font

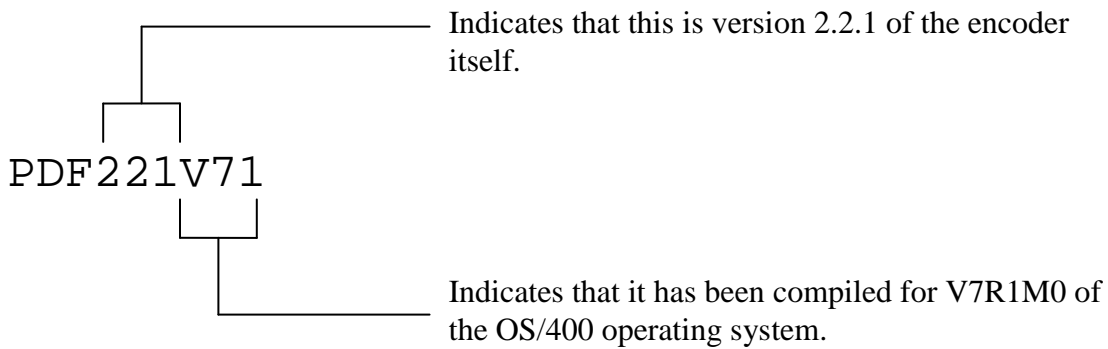
In the “fonts/xerox” subdirectory:

File	Contents
X5L309.FNT	Xerox 5-word 3x9 Landscape-model font
X5L310.FNT	Xerox 5-word 3x10 Landscape-model font
X5L412.FNT	Xerox 5-word 4x12 Landscape-model font
X5P309.FNT	Xerox 5-word 3x9 Portrait-model font
X5P310.FNT	Xerox 5-word 3x10 Portrait -model font
X5P412.FNT	Xerox 5-word 4x12 Portrait -model font
X9L309.FNT	Xerox 9700 3x9 Landscape-model font
X9L310.FNT	Xerox 9700 3x10 Landscape-model font
X9L412.FNT	Xerox 9700 4x12 Landscape-model font
X9P309.FNT	Xerox 9700 3x9 Portrait-model font
X9P310.FNT	Xerox 9700 3x10 Portrait -model font
X9P412.FNT	Xerox 9700 4x12 Portrait -model font

### 1.3 Naming Conventions

The “distribution name” of the encoder encapsulates two specific quantities – the version number of the underlying encoder software, and the version of the AS/400 operating system for which it is compiled.

This version of the encoder is 2.2.1, and the latest variant of the encoder that is provided with this distribution is designed for use on OS/400 V7R1M0 and later. Thus, the library associated with the primary variant encoder library is named as follows:



Understanding the naming convention is significant for some users, since the distribution also includes variants of the encoder compiled for earlier versions of OS/400 as well. If you choose to install one of these other variants, make the appropriate adjustments in the library names as compared to what is listed in this manual. Thus, if you choose to use the variant compiled for OS/400 V4R1M0 and later, the library name will be PDF221V41, while the variant compiled for V6R1M0 and later will have a library name of PDF221V61. Save files are named similarly, except that an “S” character replaces the “V” character.

In each case, the internal encoder version number is 2.2.1, indicating that this encoder uses the 2.1 API, and that this is the third sub-version of this library that has been shipped. Note that, for your convenience, the version numbering system that Silver Bay Software is using for the encoders is compatible with that used by NeoMedia Technologies, the original distributor of this software. Thus, the API supported by this library is the same as that supported by the 2.1.1 and 2.1.2 versions distributed by NeoMedia, since all are part of the version 2.1 family.

The remainder of this manual is written on the assumption that you will be installing and using the V7R1M0 variant of the encoder. If you are using an earlier variant, you will need to substitute the appropriate file name(s) into the example commands that are provided.

## 2 Upgrading an Existing Encoder Installation

**Warning:** *Always verify the upgrade procedure on a test or development machine before upgrading a production machine.*

If you are upgrading to this version of the encoder from a previous version, the steps you must perform differ depending on the version from which you are upgrading:

1. Install the new encoder library, using the steps listed in the next chapter.
2. Re-bind your applications to the service program that is part of the new library.

In some cases, rebinding your applications may not be necessary (it may be sufficient simply to replace the original PDF417 library with the newer library on your library list) however it is safest to explicitly rebind.

Note that, in either case, it is not necessary to replace the fonts that were delivered with the previous encoder – the current version of the encoder uses the same fonts as were previously delivered.

Because both the library name(s) and the service program name included as part of this distribution differ from previous releases provided by NeoMedia Technologies, it is possible for both encoders to co-exist on your system. Thus, if you have more than one application using the encoder, you can migrate one application at a time, if you wish.

### 3 Installing the Encoder

It is highly recommended that a skilled System Administrator with intimate knowledge of your AS/400 environment perform the installation of the PDF417 encoder. The instructions that follow are written in a general sense and will work on most AS/400 systems. However, due to the wide and varied configurations possible, it is impossible to provide precise instructions for all possible scenarios.

Installing the encoder consists of two steps. First, the encoder library itself must be installed. Secondly, steps must be taken to make the fonts available for use by the application.

#### 3.1 Requirements

Under OS/400, the PDF417 encoder requires at a minimum, OS/400 V4R1M0. If you are running an earlier release of OS/400, please contact Silver Bay Software LLC.

The encoder is supported for use with the following calling languages:

- COBOL/400 and ILE COBOL
- RPG/400 and ILE RPG
- ILE C

After completing the installation, the PDF417 encoder will be available as a service program for ILE programmers and as a callable program for OPM language programming. While using the encoder with development languages and products other than those listed above may be possible, these types of configurations have not been tested and are not supported by Silver Bay Software.

#### 3.2 Installing the Encoder Library

##### 3.2.1 *Loading the Encoder Library from a Save File (SAVF)*

The encoder is delivered as an AS/400 save file (SAVF). Multiple versions of the save file are provided, with versions of the encoder compiled for different versions of the operating system. In general, IBM provides compatibility for two full major versions forward. Thus, for example the variant of the encoder compiled for V6R1M0 will typically be compatible with versions of the operating system from V6 through V8.

The current version of the distribution disk contains the following save files:

File	Encoder Compiled For
pdf221s41.savf	V4R1M0 through V6R1M0.
pdf221s52.savf	V5R2M0 through V7R1M0
pdf221s61.savf	V6R1M0 and later.
pdf221s71.savf	V7R1M0 and later.

The steps to install from one of these files are as follows:

Step 1: On the AS/400, create a save file:



```
CRTLIB LIB(SBSPDF) TYPE(*TEST)
CRTSAVF FILE(SBSPDF/PDF221S71)
```

This example assumes that you choose to upload the savefile into the SBSPDF library during the FTP operation. You may obviously choose to create another library name or use an existing library, based on the conventions of your local site and systems organization.

Step 2: From a personal computer on which you have the CD-ROM mounted, FTP the save file up to the AS/400:

```
ftp [as400_name_or_address]
[provide username and password]
cd SBSPDF
binary
put D:\SAVEFILES\PDF221S71.SAVF PDF221S71
```

The above assumes the use of the Microsoft Windows command-line FTP program, and that the distribution CD-ROM is installed in the D: drive. It is important to ensure that the file is FTP'd up as binary to prevent ASCII-to-EBCDIC conversions from corrupting the image.

Step 3: Restore the encoder library from the save file:

```
RSTLIB DEV(*SAVF) SAVF(SBSPDF/PDF221S71) SAVLIB(PDF221V71)
```

The instructions above assume that you are loading the V7R1M0 version. The corresponding file names for all the supported variants are:

Variant	Save file on disk	Default *SAVF name	Default *LIB name
V4R1M0	pdf221s41.savf	PDF221S41	PDF221V41
V5R2M0	pdf221s52.savf	PDF221S52	PDF221V52
V6R1M0	pdf221s61.savf	PDF221S61	PDF221V61
V7R1M0	pdf221s71.savf	PDF221S71	PDF221V71

### 3.2.2 Contents of the Installed Encoder Library

Regardless of the variant installed, the encoder library contains the following objects:

Object	Type	Attribute	Contains
PDF221SRV	*SRVPGM	CLE	Service program containing the encoder API's. ILE programs using the encoder will bind to this service program.
PDFINITF	*PGM	CLE	Program implementing the PDFINITF API call for use by OPM programs.
PDFENCOD	*PGM	CLE	Program implementing the PDFENCOD API call for use by OPM programs.
H	*FILE	PF-SRC	Source physical file containing the "C" include files for the encoder API's.
QCBLLSRC	*FILE	PF-SRC	Source physical file containing the ILE COBOL sample program.
QCSRC	*FILE	PF-SRC	Source physical file containing the ILE C sample program.
QDDSSRC	*FILE	PF-SRC	Source physical file containing the DDS used by the sample programs.
QLBLSRC	*FILE	PF-SRC	Source physical file containing the COBOL/400 (OPM) sample program.
QRPGLESRC	*FILE	PF-SRC	Source physical file containing the ILE RPG sample program.
QRPGSRC	*FILE	PF-SRC	Source physical file containing the RPG/400 (OPM) sample program.
QFNTRSC	*FILE	PF-DTA	Physical file containing the AFP font objects

The encoder is shipped with all objects owned by the QDFTOWN (default owner) profile. Ownership of the objects can be changed to match your system's conventions using the CHGOBJOWN command. Similarly, the library and its contents are set to have public authority of \*ALL. This can be changed to match your system's conventions using the RVKOBJAUT and/or GRTOBJAUT commands.

### 3.3 Preparing the Encoder for Use

The distribution library contains the source files for the AFP fonts. If you are going to print using these fonts, the font resources will need to be compiled on your system. This operation is described below.

If you require either Xerox or HP-PCL fonts, these are not included in the AS/400 library restored from the save file or CD-ROM. These fonts will be found in a subdirectory on the CD-ROM. You will need to use a non-AS/400 computer (for example, a Microsoft Windows-based, Macintosh or UNIX system) to retrieve these files.

Please refer to your License Agreement for specific details on which fonts have been licensed for your use. Use of unlicensed fonts is a violation of the Agreement and is strictly prohibited.

### 3.3.1 AFP Fonts

If you intend to use the AFP fonts, you will need to create the PDF417 font resources from the source files provided. Two fonts have been supplied: one for 240 DPI printers and the other for 300, 600, and 1200 DPI printers. We recommend that you compile and install both fonts.

Font files	Resolution	
C0PD2206 T1PD2206 X0PD2206	240 dpi	
C0PD2309 T1PD2309 X0PD2309		
C0PD3309 T1PD3309 X0PD3309		300 dpi
C0PD3412 T1PD3412 X0PD3412		

PSF/400 searches a default set of single byte character set libraries (QFNT01 through QFNT19) when looking for AFP font resources. We recommend that you install the PDF417 fonts in one of these libraries. If you have not installed custom fonts on your system before, you may have to create a library (e.g., via the command `CRTLIB LIB(QFNT01)`). Note that we do **not** recommend that the fonts be installed in the IBM-supplied font library QFNTCPL, since a system upgrade could overwrite this library, causing your fonts to be deleted.

The following commands demonstrate how to compile the font resources, installing them in the QFNT01 library. Note that the order in which the files are compiled is important: a coded font's character set and code page must be compiled first (e.g., C0PD2206 and T1PD2206 must be compiled before X0PD2206 and so forth).

```

CRTFNTRSC FNTRSC(QFNT01/C0PD2206) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/T1PD2206) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/X0PD2206) FILE(PDF221V71/QFNTRSC)

CRTFNTRSC FNTRSC(QFNT01/C0PD2309) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/T1PD2309) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/X0PD2309) FILE(PDF221V71/QFNTRSC)

CRTFNTRSC FNTRSC(QFNT01/C0PD3309) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/T1PD3309) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/X0PD3309) FILE(PDF221V71/QFNTRSC)

CRTFNTRSC FNTRSC(QFNT01/C0PD3412) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/T1PD3412) FILE(PDF221V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/X0PD3412) FILE(PDF221V71/QFNTRSC)

```

These fonts use the standard AFP font naming conventions. Thus, those are zeros in the file names, not O's.

Any printer writers that will be using these fonts must be stopped and restarted to guarantee that they find the new fonts.

### 3.3.2 HP-PCL Fonts

HP-PCL fonts are provided in the `fonts/hp` subdirectory on the distribution CD-ROM. There are two font files, one for portrait mode printing and one for landscape mode printing:

File	Contents
HPPL3309.FLJ	HP-PCL 3x9 Landscape-mode font
HPPL3412.FLJ	HP-PCL 4x12 Landscape-mode font
HPPL3309.FLJ	HP-PCL 3x9 Portrait-mode font
HPPL3412.FLJ	HP-PCL 4x12 Portrait-mode font

How you use these fonts depends a great deal on your particular printing system design. Depending on your particular setup, these fonts could be directly installed into your printer, or could be downloaded to the printer as part of your print job.

### 3.3.3 Xerox Fonts

Xerox fonts are provided in the `fonts/xerox` subdirectory on the distribution CD-ROM. Two types of fonts are provided – Xerox “5-word” fonts, and older “9700” fonts. There are two font files for each font, one for portrait mode printing and one for landscape mode printing.

File	Contents
X5L309.FNT	Xerox 5-word 3x9 Landscape-model font
X5L310.FNT	Xerox 5-word 3x10 Landscape-model font
X5L412.FNT	Xerox 5-word 4x12 Landscape-model font
X5P309.FNT	Xerox 5-word 3x9 Portrait-model font
X5P310.FNT	Xerox 5-word 3x10 Portrait -model font
X5P412.FNT	Xerox 5-word 4x12 Portrait -model font
X9L309.FNT	Xerox 9700 3x9 Landscape-model font
X9L310.FNT	Xerox 9700 3x10 Landscape-model font
X9L412.FNT	Xerox 9700 4x12 Landscape-model font
X9P309.FNT	Xerox 9700 3x9 Portrait-model font
X9P310.FNT	Xerox 9700 3x10 Portrait -model font
X9P412.FNT	Xerox 9700 4x12 Portrait -model font

The appropriate fonts should be loaded onto your printer using standard Xerox procedures.

## 4 Binding to the Encoder

The AS/400 has two distinct development environments: OPM and ILE. The OPM (Original Programming Model) consists of language products like RPG/400, COBOL/400, and PL/I. ILE (Integrate Language Environment) consists of language products like ILE RPG, ILE COBOL, and ILE C.

The way the PDF417 encoder library is *used* in these two environments is identical; however, the way the encoder is *called* differs.

### 4.1.1 ILE Programming Environment

The encoder was written and compiled using ILE C, and the software is primarily distributed as an ILE service program. In ILE environments, linking with the encoder is straightforward; it is simply specified as a Bind Service Program to the Create Program command (CRTPGM). For example, if your calling ILE program were named MYLIB/MYPROGRAM, the program would be linked as follows:

```
CRTPGM PGM(MYLIB/MYPROGRAM) BNDSRVPGM(PDF221V71/PDF221SRV)
```

If the PDF221V71 library has been added to the library list, you may instead wish to use the following:

```
CRTPGM PGM(MYLIB/MYPROGRAM) BNDSRVPGM(*LIBL/PDF221SRV)
```

### 4.1.2 OPM Programming Environment

Using the PDF417 encoder with an OPM language is a little different. An OPM program cannot directly call an ILE service program. Therefore, an interface program (PDF221V71/PDFENCOD) has been provided. This is a “stub” C program that, when invoked, in turn calls the PDF221SRV service program, since. The PDFENCOD program was linked using the following command:

```
CRTPGM PGM(PDF221V71/PDFENCOD) MODULE(PDFENCOD)
      BNDSRVPGM(*LIBL/PDF221SRV) ACTGROUP(*CALLER)
```

There is a subtle implication here; the PDF221SRV service program was linked from the library list. Thus, if you are using an OPM language, the PDF221SRV service program **must** be in a library that appears in the job’s library list when it executes. Thus, you may either add the PDF221V71 library to the job’s library list, or you may copy the PDF221SRV service program into a library that is in the job’s library list.

A second OPM program (PDF221V71/PDFINITF) is provided to handle the optional initialization call. The same comments apply to this program as to the PDFENCOD program.

Note for advanced programmers: The PDF417 API provides a number of advanced function calls other than the PDFINITF and PDFENCOD calls. Only these two calls are available to OPM programs, however. If you have a need for other PDF417 API functions in an OPM environment, contact Silver Bay Software LLC.

## 5 Printing

The encoder software returns a block of characters to the caller. If you were to examine the output of the PDF417 encoder, the actual character output of the encoder looks something like this:

```
OOEDGEALCJBNALBHNIC LFB KCJILAOMADH
OOEDGMDCCBOKINOGJKLDOGDF KOEAMOMADH
OOEDFJGLCCHEMKHILLOJHGICLKE OIOMADH
OOEDGMBOK ECAIJA OCBNDE HBKNOIOMADH
OOEDGELAKDNAAMF DLGJGMAFCJNAIOMADH
OOEDGJOB CFENAIHENKHHLGNDFKJOIAOMADH
OOEDECMNBB IA LLDNNLLGMJOBILNAOMADH
OOEDGJDH CMADALKIHN OGMGHFLKNIFAOMADH
OOEDECCNCMCMADDANLNDGENLLCODNMOMADH
OOEDFHNGJAIJAHFJ LJF E FCBHLAIOMADH
OOEDGDNACJACGM ODHDFBGBCJGAIOMADH
OOEDEA OBHBOACHHF IHDNGCNIJHLOIOMADH
```

When printed using the supplied PDF417 font, the bars and spaces of the PDF417 symbol are printed:



It is up to the application software and printing environment to cause the PDF417 font to be selected when the characters associated with the symbol are printed.

In addition, the characters must be properly spaced. The horizontal positioning of the PDF417 characters within a single line is controlled by information within the font. That is, as each character of a line is rendered by the printer, the position of the adjacent character is automatically determined by the printer. However, *the line-to-line spacing must be controlled by the programmer.*

Line spacing can be expressed in a couple of ways. For example, many printers use lines per inch. The lines of the PDF417 symbol need to be printed at a spacing that depends on the particular font being used. Unfortunately, this spacing rarely coincides with one of the supported line spacing values supported by the print system.

The following sections provide suggestions on how to perform font selection and proper vertical spacing in different printing environments.

### 5.1 Printing using DDS

The PDF\_API .PDF manual provides more detail on printing using the AFP font. As mentioned above, the line spacing you will use is based on which font you select. Most DDS users are accustomed to setting line spacing using the LPI function. However, LPI cannot set “custom” line per inch settings (it can only set 4, 6, 8, 9, and 12 lines per inch). A PDF-417 symbol, even using the largest font, requires much finer line spacing.

The solution then is to use explicit POSITION calls for each line of output. The DDS must place each line of output at its precise location on the page. The table for AFP fonts provided earlier in this section lists the necessary line spacing (in thousandths of an inch). The following is a sample for a 300 DPI printer using the X0PD3309 font. Note that this font uses a line spacing of 0.0300 inches:

A			CDEFNT(X0PD3309)
A	PDFOUT01	37A	POSITION(1.700 1.350)
A	PDFOUT02	37A	POSITION(1.730 1.350)
A	PDFOUT03	37A	POSITION(1.760 1.350)
A	PDFOUT04	37A	POSITION(1.790 1.350)
A	PDFOUT05	37A	POSITION(1.820 1.350)
A	PDFOUT06	37A	POSITION(1.850 1.350)
A	PDFOUT07	37A	POSITION(1.880 1.350)
A	PDFOUT08	37A	POSITION(1.910 1.350)
A	PDFOUT09	37A	POSITION(1.940 1.350)
A	PDFOUT10	37A	POSITION(1.970 1.350)
A	PDFOUT11	37A	POSITION(2.000 1.350)
A	PDFOUT12	37A	POSITION(2.030 1.350)
A	PDFOUT13	37A	POSITION(2.060 1.350)
A	PDFOUT14	37A	POSITION(2.090 1.350)
A	PDFOUT15	37A	POSITION(2.120 1.350)
A	PDFOUT16	37A	POSITION(2.150 1.350)
A	PDFOUT17	37A	POSITION(2.180 1.350)
A	PDFOUT18	37A	POSITION(2.210 1.350)
A	PDFOUT19	37A	POSITION(2.240 1.350)
A	PDFOUT20	37A	POSITION(2.270 1.350)
A	PDFOUT21	37A	POSITION(2.300 1.350)
A	PDFOUT22	37A	POSITION(2.330 1.350)
A	PDFOUT23	37A	POSITION(2.360 1.350)
A	PDFOUT24	37A	POSITION(2.390 1.350)
A	PDFOUT25	37A	POSITION(2.420 1.350)

This DDS prints a symbol 1.70 inches down and 1.35 inches over on the page. Each of the successive 25 lines of output is 0.030 inches lower than the previous one (as each row of this particular PDF-417 symbol is 0.030 inches high).

The CDEFNT function selects the font – in this case the X0PD3309 font. The X0PD3309 and X0PD3412 fonts are for use with 300, 600, and 1200 DPI page printers. If your printer is a 240 DPI printer, use the X0PD2206 or X0PD2309 font.

## 5.2 Printing using HP-PCL Fonts

The design of AS/400 software to produce HP-PCL output commands is beyond the scope of this manual; however the general approach is as follows:

Step 1: As part of the beginning of your AS/400 job, include the contents of the appropriate font as part of the binary stream sent to the printer. Typically, you will precede the font with the HP-PCL escape sequence to assign the font an ID of your choosing. For example, if you chose to use a font ID of 12, you would send the sequence:

$$^E_c *cnnD \quad (\text{where } ^E_c \text{ is the ASCII ESC character, and } nn \text{ is the font ID})$$

followed by the contents of the font. The font is binary information, so it is important to ensure that no EBCDIC-to-ASCII conversion occurs on this data in the path between the system and the printer.

Step 2: Map the font ID specified for the PDF417 font as the PCL secondary font

$$^E_c )nnX \quad (\text{where } ^E_c \text{ is the ASCII ESC character, and } nn \text{ is the font ID})$$

Step 3: When it is time to print the PDF417 symbol, invoke the secondary font using the ASCII  $^S_o$  (Shift Out) character.

Step 4: Print the characters associated with the PDF417 symbol. Each individual line of the symbol must be explicitly positioned below the previous line, thus achieving the desired line spacing. Each individual line may be positioned using the Horizontal Cursor Positioning and Vertical Cursor Positioning escape sequence:

$$^E_c *pxxxX^E_c *pyyyY \quad (\text{where } xxx \text{ and } yyy \text{ are the horizontal and vertical positions of the individual line expressed in "PCL Units"})$$

Recall that HP-PCL uses 300dpi "PCL Units" when positioning, so a spacing of 0.030 inches would result in an increase of 9 of the yyy value for each line.

Step5: After the symbol has been completely printed, return to the primary font using the ASCII  $^S_I$  (Shift In) character to print the remainder of the page.

There are obviously other ways that can be used to select a particular font at the appropriate point in the print stream – the above is included as one example. Consult the *PCL Printer Language Technical Reference Manual*, available from Hewlett-Packard, for more details on the use of HP-PCL soft fonts.

Note that unless a "reset" escape sequence is sent to the printer during the print stream, it is only necessary to download the font to the printer once at the beginning of the print job.

## 5.3 Printing using Xerox Fonts

The design of AS/400 software to select fonts via Metacode is beyond the scope of this manual. As with the AFP and HP-PCL fonts, the basic procedure is to invoke the appropriate font while printing the characters associated with the PDF417 symbol, and to position each individual line of the symbol based on the particular font being used.



## 6 Sample Programs

Five sample programs have been provided:

- A COBOL/400 program
- An ILE COBOL program
- An RPG/400 program
- An ILE RPG program
- An ILE C program

For simplicity of compiling and running the samples, you may wish to change your current library to the PDF221V71 library.

### 6.1 Sample Printer File

The three sample programs use the same printer file, PDF221V71/PDF417. Before you can compile any of the sample programs, you must create the printer file. Two different sample DDS source files are provided – one for 240 dpi printing and one for 300 dpi printing.

The sources for the printer file are PDF221V71/QDDSSRC(PDF240) and PDF221V71/QDDSSRC(PDF300). These are a very simple DDS files that print a PDF417 symbol using the supplied font.

To create the printer file for 240 dpi printing:

```
CRTPRTF FILE(PDF221V71/PDF417) SRCFILE(PDF221V71/QDDSSRC)
      SRCMBR(PDF240) DEVTYPE(*AFPDS) REPLACE(*YES)
```

For 300 dpi printing, use this instead:

```
CRTPRTF FILE(PDF221V71/PDF417) SRCFILE(PDF221V71/QDDSSRC)
      SRCMBR(PDF300) DEVTYPE(*AFPDS) REPLACE(*YES)
```

### 6.2 COBOL Samples

#### 6.2.1 *Compiling and Running the Sample COBOL/400 Program*

The sample COBOL/400 program calls the PDF221V71/PDFENCOD program. The source for the sample program is in PDF221V71/QLBLSRC(CBLPDF). It can be compiled using the following command:

```
CRTCBLPGM PGM(PDF221V71/CBLPDF) SRCFILE(PDF221V71/QLBLSRC)
      SRCMBR(CBLPDF) REPLACE(*YES)
```

The program is now ready to run. It will generate a spool file named PDF417 to the default output queue:

```
CALL CBLPDF
```

Remember that, because of the way that the PDFENCOD program is linked, the PDF221V71 library must be in your library list in order for this program to run correctly.

### ***6.2.2 Compiling and Running the Sample ILE COBOL Program***

The sample ILE COBOL program links with the PDF221V71/PDF221SRV service program. The source for the sample program is in PDF221V71/QCBLLESRC(CBLPDF). You must first compile the COBOL module:

```
CRTCBLMOD MODULE(PDF221V71/CBLPDF)
      SRCFILE(PDF221V71/QCBLLESRC)
      SRCMBR(CBLPDF)REPLACE(*YES)
```

Now create the ILE program (i.e., link it with the PDF417 service program):

```
CRTPGM PGM(PDF221V71/ILECBLPDF) MODULE(PDF221V71/CBLPDF)
      BNDSRVPGM(PDF221V71/PDF221SRV) REPLACE(*YES)
```

The program is now ready to run. It will generate a spool file named PDF417 to the default output queue:

```
CALL ILECBLPDF
```

## **6.3 RPG Samples**

### ***6.3.1 Compiling and Running the Sample RPG/400 Program***

The sample RPG/400 program calls the PDF221V71/PDFENCOD program. The source for the sample program is in PDF221V71/QRPGSRC(RPGPDF). It can be compiled using the following command:

```
CRTRPGPGM PGM(PDF221V71/RPGPDF) SRCFILE(PDF221V71/QRPGSRC)
      SRCMBR(RPGPDF) REPLACE(*YES)
```

The program is now ready to run. It will generate a spool file named PDF417 to the default output queue:

```
CALL RPGPDF
```

Remember that, because of the way that the PDFENCOD program is linked, the PDF221V71 library must be in your library list in order for this program to run correctly.

### ***6.3.2 Compiling and Running the Sample ILE RPG Program***

The sample ILE RPG program links with the PDF221V71/PDF221SRV service program. The source for the sample program is in PDF221V71/QRPGLESRC(RPGPDF). You must first compile the RPG module:

```
CRTRPGMOD MODULE(PDF221V71/RPGPDF)
      SRCFILE(PDF221V71/QRPGLESRC) SRCMBR(RPGPDF)
      REPLACE(*YES)
```

Now create the ILE program (i.e., link it with the PDF417 service program):

```
CRTPGM PGM(PDF221V71/ILERPGPDF) MODULE(PDF221V71/RPGPDF)
      BNDSRVPGM(PDF221V71/PDF221SRV) REPLACE(*YES)
```

The program is now ready to run. It will generate a spool file named PDF417 to the default output queue:

```
CALL ILERPGPDF
```

## 6.4 ILE C Sample

The sample ILE C program links with the PDF221V71/PDF221SRV service program. The source for the sample program is in PDF221V71/QCSRC(CPDF). You must first compile the C module:

```
CRTCMOD MODULE(PDF221V71/CPDF) SRCFILE(PDF221V71/QCSRC)
SRCMBR(CPDF) REPLACE(*YES)
```

Now create the ILE program (i.e., link it with the PDF417 service program):

```
CRTPGM PGM(PDF221V71/CPDF) MODULE(PDF221V71/CPDF)
BNDSRVPGM(PDF221V71/PDF221SRV) REPLACE(*YES)
```

The program is now ready to run. It will generate a spool file named PDF417 to the default output queue:

```
CALL CPDF
```

## 6.5 Output of the Sample Programs

All of the sample programs generate a single PDF417 symbol to the output spooler. If you view the spooler, you should see 17 lines of text, comprised of alphabetic characters and spaces. Here's an example of what the spool file would look like:

```
OOEDGEALCJBNALBHNIC LFB KCJILAOMADH
OOEDGMDCCBOKINOGJKLDOGDF KOEAMOMADH
OOEDFJGLCCHEMKHILLOJHGICLKE OIOMADH
OOEDGMBOK ECAIJA OCBNDE HBKNOIOMADH
OOEDGELAKDNAAMF DLGJGMAFCJNAIOMADH
OOEDGJOB CFENAIHENKHHLGNDFKJOIAOMADH
OOEDECMNBB IA LLDNNLLGMJOBILNAOMADH
OOEDGJDHC MADALKIHN OGMGHFLKNIFAOMADH
OOEDECCNCMCMADDANLNDGENLLCODNMOMADH
OOEDFHNGJAIJAHFJ LJF E FCBHLAIOMADH
OOEDGDNACJACGM ODHDFBGBCJGAIOMADH
OOEDEA OBHBOACHHFIHDNGCNIJHLOIOMADH
```

When sent to the printer using one of the provided fonts, rather than the digits seen above, a PDF417 symbol comprised of bars and spaces will be printed.

If you do not get a PDF417 symbol, but rather the rows of numbers “all bunched” up, then PSF/400 is performing a font substitution (check the messages for QSYSOPR). Possible causes include:

- The PDF417 fonts have not been properly installed in a library that PSF/400 can find.
- The printer writer was not restarted after the PDF417 fonts were installed.
- The DDS is using the incorrect font for your printer's density.