



PDF417 Encoder

Version 2.2.1 for z/OS

z/OS Manual

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

Document Version 20120927

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

1	INTRODUCTION.....	1
1.1	CONTENTS OF THIS MANUAL	1
1.2	CONTENTS OF THE DISTRIBUTION CD-ROM	1
2	INSTALLING THE ENCODER	3
2.1	REQUIREMENTS.....	3
2.2	INSTALLING THE ENCODER LIBRARY	3
2.2.1	<i>Loading the datasets XMIT (Transmit) Files.....</i>	<i>3</i>
2.2.2	<i>Contents of the datasets</i>	<i>4</i>
2.3	PREPARING THE ENCODER FOR USE	5
2.3.1	<i>Font installation.....</i>	<i>5</i>
2.3.2	<i>AFP fonts</i>	<i>6</i>
2.3.3	<i>HP-PCL Fonts</i>	<i>6</i>
2.3.4	<i>Xerox Fonts.....</i>	<i>7</i>
3	LINKING THE ENCODER.....	8
4	PRINTING.....	9
4.1	PRINTING USING DDS	9
4.2	PRINTING USING HP-PCL FONTS	10
4.3	PRINTING USING XEROX FONTS	11
5	SAMPLE PROGRAMS.....	12
5.1	COMPLING AND RUNNING THE SAMPLE COBOL PROGRAM	12
5.2	COMPILING AND RUNNING THE SAMPLE C PROGRAM	12
5.3	OUTPUT OF THE SAMPLE PROGRAMS	13

Silver Bay Software LLC PDF417 Encoder

Version 2.2.1 for z/OS

z/OS 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 z/OS. Specifically, this manual covers:

- Installing 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 folder:

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

In the "transmit" folder:

File	Contents
PDF417.FNT.XMT	TSO/E Transmit file containing the PDF417 font resource files for AFP.
PDF417.OBJ.XMT	TSO/E Transmit file containing the PDF417 encoder object library.
PDF417.RSC.XMT	TSO/E Transmit file containing various sample programs, resource files, and JCL for the PDF417 encoder.

In the “docs” folder:

File	Contents
PDF_ZOS.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” folder:

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” folder:

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

2 Installing the Encoder

It is highly recommended that a skilled Systems Programmer with intimate knowledge of your z/OS environment perform the installation of the PDF417 encoder. The instructions that follow are written in a general sense and will require some customization, including changes to the sample JCL provided. Due to the wide and varied configurations possible, it is not possible 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.

2.1 Requirements

Under z/OS (MVS/OS390), the PDF417 encoder requires at a minimum, the C runtime libraries. These libraries are a standard part of most z/OS installations and should already be present on your system.

OS/390 and MVS do not include the C Language runtime libraries by default; they required additional software licenses from IBM for the C runtime libraries. Typically the libraries will be installed with:

- LE/370 (Language Environment 370)
- C/370 (C Development/Runtime 370)

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

- COBOL II
- COBOL 370
- C

Many of the JCL jobs use an address mode of 31 (AMODE=31). You will need to edit the JCL if you are using the older address mode of 24 for the LINK step parameters. It is up to the user to determine the settings for your environment and make the appropriate modifications to the provided JCL.

After completing the installation, the PDF417 encoder routines will be available as distinct load modules in a Partitioned Data Set (PDS) load library. Your object modules will need to be linked with this PDS object. 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.

2.2 Installing the Encoder Library

2.2.1 *Loading the datasets XMIT (Transmit) Files*

The software is delivered as three TSO/E XMIT files on the distribution CD which need to be transferred to your z/OS system (e.g., using binary FTP), then the datasets restored using RECEIVE.

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

File	Encoder Compiled For
pdf417.fnt.xmt	AFP fonts
pdf417.obj.xmt	PDF417 encoder object library
pdf417.rsc.xmt	PDF417 resources

The XMIT file contents can be restored to any dataset name, however, note that this documentation as well as all of the sample JCL assume the use of the dataset names below (note: the names below may not match the default name contained in the XMIT file).

The JCL examples will need to be modified with any dataset names you choose to use.

File	Dataset name
pdf417.fnt.xmt	NEOMEDIA.PDF417.FNTLIB
pdf417.obj.xmt	NEOMEDIA.PDF417.OBJLIB
pdf417.rsc.xmt	NEOMEDIA.PDF417.RSCLIB

2.2.2 Contents of the datasets

Contents of the **NEOMEDIA.PDF417.RSCLIB** dataset:

Member	Description
IPDFLNK	Sample JCL for linking encoder objects into a load module
ISAMLCB	Sample JCL for building SAMPLCOB, no pre-linker
ISAMLEC	Sample JCL for building SAMPLEC, no pre-linker
ISAMPLCB	Sample JCL for building SAMPLCOB, using pre-linker
ISAMPLEC	Sample JCL for building SAMPLEC, using pre-linker
NMPDF417	COPYLIB for sample COBOL program
PDFDEFS	C include file (definitions)
PDFENC	C include file (prototypes for encoder API)
PDFREND	C include file (prototypes for rendering API)
SAMPLCOB	Sample COBOL program
SAMPLEC	Sample C program
SINKFILE	C include file (prototypes for sink to file)
SINKFILS	C source: sample sink to file
SINKMEM	C include file (prototypes for sink to memory)
SINKMEMS	C source: sample sink to memory

Member	Description
XSAMPLCB	Sample JCL for executing SAMPLCOB
XSAMPLEC	Sample JCL for executing SAMPLEC

Contents of the **NEOMEDIA.PDF417.OBJLIB** dataset:

Object	Description
PDF417EN	PDF417 Encoder object module

Contents of the **NEOMEDIA.PDF417.FNTLIB** dataset:

Object	Description
C0PD2206	AFP character set, bounded box, 240 dpi. 2x6 pixel module size
C0PD2309	AFP character set, bounded box, 240 dpi. 3x9 pixel module size
C0PD3309	AFP character set, bounded box, 300 dpi. 3x9 pixel module size
C0PD3412	AFP character set, bounded box, 300 dpi. 4x12 pixel module size
T1PDF417	AFP code page, bounded box.
X0PD2206	AFP coded font, bounded box, 240 dpi. 2x6 pixel module size
X0PD2309	AFP coded font, bounded box, 240 dpi. 3x9 pixel module size
X0PD3309	AFP coded font, bounded box, 300 dpi. 3x9 pixel module size
X0PD3412	AFP coded font, bounded box, 300 dpi. 4x12 pixel module size

2.3 Preparing the Encoder for Use

The PDF417 encoder is distributed in object form and must first be linked with the C runtime environment as a load library, including the creation of the named entry points. This is a simple link edit step and sample JCL has been provided with the distribution:

```
NEOMEDIA.PDF417.RSCLIB(IPDFLNK)
```

Inputs to the link edit are your system's C runtime library (e.g., CEE.SCEELKED) and the PDF417 object module **NEOMEDIA.PDF417.OBJLIB(PDF417EN)**. Output is the load library **NEOMEDIA.PDF417.LOADLIB(PDF417)**.

Examine, modify, then SUBMIT this JCL to create the load library for the PDF417 Encoder software.

2.3.1 *Font installation*

The PDF417 Encoder uses custom fonts for the printing of the barcode. Before a barcode can be printed, the fonts must be properly installed in your environment. AFP fonts have been provided in the font resource dataset (**NEOMEDIA.PDF417.FNTLIB**). If you require either Xerox or HP-PCL fonts, these can be found in the **fonts** folder on the original distribution CD-ROM.

You will need to use a desktop system (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.

2.3.2 *AFP fonts*

The AFP fonts for the PDF417 Encoder have been provided in a separate dataset (NEOMEDIA.PDF417.FNTLIB). Included are fonts for both 240 DPI and 300 (600) DPI printers. The table below provides details on the member names.

Font files	Resolution
C0PD2206 X0PD2206	240 dpi
C0PD2309 X0PD2309	
C0PD3309 X0PD3309	
C0PD3412 X0PD3412	
T1PDF417	Common to both

This library can be added to PSF's system library definition, or specified in the OUTPUT JCL statement of an application requiring them. For example:

```
//OUT1      OUTPUT      USERLIB=(NEOMEDIA.PDF417.FNTLIB)
```

2.3.3 *HP-PCL Fonts*

HP-PCL fonts are provided in the fonts/hp folder on the distribution CD-ROM. There are two sets of 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.

2.3.4 Xerox Fonts

Xerox fonts are provided in the `fonts/xerox` folder 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.

3 Linking the Encoder

Even though the PDF417 Encoder is written using the C language, it does not require the use of the pre-linker (it does not rely utilize any system programming facilities of the C environment).

Thus, the PDF417 Encoder can be linked with or without the pre-linker, as your application requires. Sample JCL has been provided for linking with and without the pre-linker for the both the example COBOL and C programs in `NEOMEDIA.PDF417.RSCLIB`. Refer to section 5 of this guide for more information on these sample programs and the JCL for linking them.

4 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
OOEDGJOBCFENAIHENKHHLGNDFKJOIAOMADH
OOEDECMNBB IA LLDNNLLGMJOBILNAOMADH
OOEDGJDHCMADALKIHNOMGMHFLKNIFAOMADH
OOEDECCNCMCMADDANLNDGENLLCODNMOMADH
OOEDFHNGJAIJAHFJ LJF E FCBHLAIOMADH
OOEDGDNAKACJACGM 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.

4.1 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.

4.2 Printing using HP-PCL Fonts

The design of an application program 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 print 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:

$^E_c * c n n D$ (where E_c is the ASCII ESC character, and nn 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

$^E_C)nnX$ (where E_C is the ASCII ESC character, and nn 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:

$^E_C*pxxxX^E_C*pyyyY$ (where xxx and yyy 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.

4.3 Printing using Xerox Fonts

The design of an application program 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.

5 Sample Programs

Two sample programs have been provided:

- A COBOL program
- A C program

The purpose of the sample programs is to demonstrate invoking of the PDF417 encoder in a simple fashion. The sample programs do not attempt to generate output to a printer, since this behavior is very environment specific.

JCL for compiling and linking with and without the pre-linker for both sample programs has been provided. Note that compiling and running the sample programs is not required as part of the installation of the software.

The sample programs and JCL can all be found in the resource library provided (NEOMEDIA.PDF417.RSCLIB).

5.1 Compiling and running the sample COBOL program

The following files comprise the sample COBOL program:

Member	Description
ISAMLCB	Sample JCL for building SAMPLCOB, no pre-linker
ISAMPLCB	Sample JCL for building SAMPLCOB, using pre-linker
NMPDF417	COPYLIB for sample COBOL program
SAMPLCOB	Sample COBOL program
XSAMPLCB	Sample JCL for executing SAMPLCOB

- To compile and link the sample program without using the pre-linker, examine, modify, then submit the ISAMLCB JCL.
- To compile and link the sample program using the pre-linker, examine, modify, then submit the ISAMPLCB JCL.
- To execute the sample COBOL program, examine, modify, then submit the XSAMPLCB JCL. Note that output from the program will be directed to the TPDFFONT DD (NEOMEDIA.PDF417.TPDFFONT in the sample JCL).
- Examine the output file; it should resemble the example output characters demonstrated in section 5.3 of this guide.

5.2 Compiling and running the sample C program

The following files comprise the sample C program:

Member	Description
ISAMLC	Sample JCL for building SAMPLC, no pre-linker

Member	Description
ISAMPLC	Sample JCL for building SAMPLC, using pre-linker
PDFENC	Prototypes for encoder functions
PDFREND	Prototypes for rendering functions
SAMPLC	Sample C program
SINKFILE	Prototypes for SINKFILS
SINKFILS	C routine for file output
XSAMPLC	Sample JCL for executing SAMPLC

- To compile and link the sample program without using the pre-linker, examine, modify, then submit the ISAMLC JCL.
- To compile and link the sample program using the pre-linker, examine, modify, then submit the ISAMPLC JCL.
- To execute the sample C program, examine, modify, then submit the XSAMPLC JCL. Note that output from the program will be directed to the TPDFFONT DD (NEOMEDIA.PDF417.TPDFFONT in the sample JCL).

Examine the output file; it should resemble the example output characters demonstrated in section 5.3 of this guide.

5.3 Output of the Sample Programs

The sample programs generate a single PDF417 symbol to TPDFFONT DD defined in the JCL for executing the program. Here is an example of what the output will look like:

This is a test of the NeoMedia Technologies PDF417 Encoder.
The error correction will encode at level 3 with ECC padding.

```
OOEDGEALCJBNALBHNIC LFB KCJILAOMADH
OOEDGMDCCBOKINOGJKLDOGDF KOEAMOMADH
OOEDFJGLCCHEMKHILLOJHGICLKE OIOMADH
OOEDGMBOK ECAIJA OCBNDE HBKNOIOMADH
OOEDGELAKDNAAMF DLGJGMAFCJNAIOMADH
OOEDGJOBCFENAIHENKHHLGNDFKJOIAOMADH
OOEDECMNBB IA LLDNNLLGMJOBILNAOMADH
OOEDGJDHCMADALKIHNOMGMGHFLKNIFAOMADH
OOEDECCNCMCADDANLNDGENLLCODNMOMADH
OOEDFHNGJAIJAHFJ LJF E FCBHLAIOMADH
OOEDGDNAKACJACGM ODHDFBGBCJGAIOMADH
OOEDEA OBHBOACHHFIHDNGCNIJHLOIOMADH
```

Sending this file to a printer with the correct PDF417 font is beyond the scope of this sample program. More detailed instructions on font selection and the actual printing of the barcode are provided in the programmer's reference.