



PDF417 Encoder

Version 2.2.1

Linux Manual

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

Document Version 20130409

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 © 2013, 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	1
2	INSTALLING THE ENCODER	3
2.1	INSTALLING THE ENCODER LIBRARY	3
2.1.1	<i>Extract the distribution TAR</i>	3
2.1.2	<i>Using the installation script</i>	3
2.1.3	<i>Manual installation</i>	4
2.1.4	<i>Versions of the Encoder Library</i>	5
2.2	INSTALLING THE FONTS	5
2.2.1	<i>HP-PCL Fonts</i>	6
2.2.2	<i>Xerox Fonts</i>	6
3	COMPILING AND LINKING WITH THE ENCODER.....	7
3.1	COMPILING	7
3.2	LINKING WITH THE STATIC LIBRARY	7
3.3	LINKING WITH THE SHARED LIBRARY	7
4	SAMPLE PROGRAM	9

Silver Bay Software LLC PDF417 Encoder

Version 2.2.1

Linux 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 Linux environment. Specifically, this manual covers:

- Installing and upgrading the encoder
- 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

The distribution CD-ROM or ZIP file for the encoder consists of the product documentation, font resources, and the distribution TAR file. The distribution media contains the following:

Root folder:

Folder	Contents
docs	Documentation folder
fonts	Font folder
pdf417v2.2.1.tar	PDF417 tar file for Linux

In the "Docs" folder:

File	Contents
pdf_linux.pdf	Product installation guide (this manual).
pdf_api.pdf	API guide (information for programmers is contained in this document).

In the “fonts” folder:

Folder	Contents
hp/	HP-PCL fonts
Xerox/	Xerox 5-word and 9700 series fonts

In the “pdf417v2.2.1.tar” archive:

Directory/file	Contents
pdf417v2.2.1/install.sh	BASH script for installing the PDF417 encoder
pdf417v2.2.1/include/	C include files
pdf417v2.2.1/lib32/	32-bit PDF417 library files
pdf417v2.2.1/lib64/	64-bit PDF417 library files
pdf417v2.2.1/samples/	Miscellaneous sample source code

2 Installing the Encoder

It is highly recommended that a skilled System Administrator with intimate knowledge of your Linux environment perform the installation of the PDF417 encoder. The instructions that follow are written in a general sense and will work on most Linux 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.

The distribution includes an installation script which can optionally be used to install the libraries and C include files. However, execution of this script requires root permissions (i.e., sudo) as it utilizes `ldconfig` to link the shared library and update the run-time linker's cache.

2.1 Installing the encoder library

This step consists of installing the encoder's libraries and supporting C include files. This step is necessary on all Linux systems in your environment that will be linking with the encoder library (for example, developer machines, and even production machines if your applications are compiled and linked there, or if you will be using the shared object library version of the encoder).

2.1.1 Extract the distribution TAR

Included in your distribution media is a tape archive file (tar). This file needs to be placed on your Linux system (e.g., binary file transfer like SCP or FTP), and its contents extracted. After transferring the tar file to your Linux system, extract its contents:

```
tar xvf pdf417v2.2.1.tar
```

This will create a subdirectory named `pdf417v2.2.1`, which contains the files and installation script to complete the installation process.

2.1.2 Using the installation script

A BASH script has been included to simplify installation of the encoder product. The script can be used to install either the 32-bit libraries, the 64-bit libraries, or both. Note that the script also installs the shared object library version as well and thus requires root permission.

To run the installation script, on a Ubuntu system for example:

```
cd pdf417v2.2.1
sudo ./install.sh
```

You will be prompted for which architectures to install (32-bit, 64-bit, or both), as well as target directory names for the installed components. The script has the following default directories:

Installation option	Default directories (library/include)
If installing 32-bit libraries on a 32-bit OS	/usr/local/lib /usr/local/include
If installing 32-bit libraries on a 64 bit OS	/usr/local/lib32 /usr/local/include
If installing 64-bit libraries on a 64-bit OS	/usr/local/lib /usr/local/include

Before performing its installation, the script will present a confirmation screen displaying the pending actions and target directories. When prompted, you must enter ‘Y’ to confirm the actions and execute the installation.

Note that the installation script will run `ldconfig` to complete the installation of the shared object library of the encoder product. Generally this involves creating or updating the library’s shared object name symbolic link (as well as updating the run-time linker cache). The encoder library uses its major and minor version numbers in its shared object name/symbolic link name. For example, installation of version 2.2.1 of the shared object library will result in the physical installation of ‘libpdf417.so.2.2.1’, with the creation of ‘libpdf417.so.2.2’ as a symbolic link by `ldconfig`.

The installation script does not create a “linker name” for the shared library; that is, it will not create a symbolic link for `libpdf417.so` to the versioned library installed. Refer to the section titled *Linking with the shared library* later in this document for more details on the shared object library.

2.1.3 Manual installation

Manual installation of the encoder product follows these basic steps:

1. Choose the source libraries for your architecture (32-bit or 64-bit). The libraries are located in the following directories:

Architecture	Directory
32-bit libraries	pdf417v2.2.1/lib32
64-bit libraries	pdf417v2.2.1/lib64

2. Copy the library files to the appropriate library directory on your system (or to your developer’s project directories). The next section provides additional details on the versions of the library provided in the distribution (static library, PIC library, and shared object library).
3. Copy the C includes files from `pdf417v2.2.1/include` to the appropriate include directory on your system (or to your developer’s project directories).
4. Run `ldconfig` (as root) to reconfigure the dynamic linker for shared object libraries. This step is only necessary if you plan to use the encoder as a shared library.

Under normal circumstances, only one of the 32-bit or 64-bit libraries will be installed, based on the way that you normally build your applications. If you require simultaneous support for 32-bit and 64-bit applications, it will be necessary to install the two different libraries in different directories, as both versions share a common filename. Alternately, you could rename the libraries files, such as to `libpdf41732.a` and `libpdf41764.a`. The C

include files are identical between the 32-bit and 64-bit versions, and thus only one copy of these needs to be installed.

2.1.4 Versions of the Encoder Library

As mentioned above, there are three versions of the encoder library. Which version you use depends on how you are building your application.

File	Used for
libpdf417.a	This is a static library, whose contents will be compiled into your application. You would typically use this version of the library if your own code was not being compiled to be position-independent (i.e. without the <code>-fPIC</code> flag, if you are using the gnu compiler.)
libpdf417pic.a	Like the previous file, this is a static library that will be compiled into your application. It differs from the previous file in that the code in this library was all compiled with the <code>-fPIC</code> gcc compiler flag, to produce position-independent code. Use this version of the library if you are compiling your own code with the <code>-fPIC</code> flag, as might be the case if you are building your own shared object, particularly on a 64-bit platform.
libpdf417.so.2.2.1	This file contains the same modules as the previous two files. It differs in that it is a shared object, which will be dynamically loaded at runtime by the application linked to it. As such, unlike the previous two, this file must be available on the production machine. In general, there is little advantage to use the library as a shared object as opposed to statically linking it into your application, however some customers prefer shared objects to static linking.

2.2 Installing the Fonts

The font files that are included in the `fonts` directory of the distribution must be installed on those systems that will actively print PDF417 barcodes created by the encoder. How you install the fonts depends a great deal on your particular print system design. Some configurations include:

- The application program integrates the font as a resource directly within the print stream. This is a common approach used with HP PCL fonts.
- The font may be installed directly on the printer, then invoked by the application program when needed. Again, this approach is common with HP PCL fonts, but also with Xerox fonts.
- The font may be installed on a print server or print management systems which in turn will deliver it to the printer when needed. This approach is common with Xerox and AFP print technologies.

A discussion of all the various possibilities is beyond the scope of this document, as are the specific steps for configuring and deploying fonts in them. The remainder of this section provides information on the fonts provided in the distribution set.

For a more detailed discussion on font printing, font selection, and other printing topics refer to the *PDF417 Encoder Programmer's Manual*, in the section titled *Font and Printing Related Information*.

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.

2.2.1 HP-PCL Fonts

Both portrait and landscape mode fonts have been provided; for each orientation there are two fonts (for generating different size PDF417 barcodes). The following fonts are included in the `fonts/hp` folder of the distribution media:

File	Orientation	Description
HPPP3309.FLJ,	Portrait	Generates a PDF417 symbol with a module size of 3x9 pels.
HPPP3412.FLJ	Portrait	Generates a PDF417 symbol with a module size of 4x12 pels.
HPPL3309.FLJ	Landscape	Generates a PDF417 symbol with a module size of 3x9 pels.
HPPL3412.FLJ	Landscape	Generates a PDF417 symbol with a module size of 4x12 pels.

2.2.2 Xerox Fonts

Two families of Xerox fonts are provided – Xerox “5-word” fonts, and older “9700” fonts. For each font family, both portrait and landscape fonts are available; for each family/orientation there are three fonts (for generating different size PDF417 barcodes). The following fonts are included in the `fonts/Xerox` folder of the distribution media:

File	Family	Orientation	Description
X5P309.FNT	5-word	Portrait	Generates a PDF417 symbol with a module size of 3x9 pels.
X5P310.FNT	5-word	Portrait	Generates a PDF417 symbol with a module size of 3x10 pels.
X5P412.FNT	5-word	Portrait	Generates a PDF417 symbol with a module size of 4x12 pels.
X5L309.FNT	5-word	Landscape	Generates a PDF417 symbol with a module size of 3x9 pels.
X5L310.FNT	5-word	Landscape	Generates a PDF417 symbol with a module size of 3x10 pels.
X5L412.FNT	5-word	Landscape	Generates a PDF417 symbol with a module size of 4x12 pels.
X9P309.FNT	9700	Portrait	Generates a PDF417 symbol with a module size of 3x9 pels.
X9P310.FNT	9700	Portrait	Generates a PDF417 symbol with a module size of 3x10 pels.
X9P412.FNT	9700	Portrait	Generates a PDF417 symbol with a module size of 4x12 pels.
X9L309.FNT	9700	Landscape	Generates a PDF417 symbol with a module size of 3x9 pels.
X9L310.FNT	9700	Landscape	Generates a PDF417 symbol with a module size of 3x10 pels.
X9L412.FNT	9700	Landscape	Generates a PDF417 symbol with a module size of 4x12 pels.

3 Compiling and linking with the encoder

3.1 Compiling

Unless you have installed the encoder's C include files in a directory that is not in your compiler's default path, no special steps are required to compile a C program that includes one of the encoder's header files. In general, you can use either the double quote or angle bracket syntax for the encoder's header files. Thus, either of the following will work:

```
#include "pdfenc.h"
```

Or

```
#include <pdfenc.h>
```

In the case where the encoder's header files have not been installed in a directory that is part of your compiler's default search path, use the `-I` option (with `gcc`) to indicate the directory name where the include files are installed. For example, if the encoder's header files were installed in `/opt/pdf417/include`, the following would be necessary to compile the sample C program:

```
gcc -c -I/opt/pdf417/include samplec.c
```

3.2 Linking with the static library

The PDF417 encoder static library name is `libpdf418.a`. To link with the encoder then, the following option must be added to your link command:

```
-lpdf417
```

Assuming the encoder library has been installed in a directory that is part of the linker's default path, no additional parameters are necessary. For example, the following command could be used (with `gcc`) to compile and link the sample program provided:

```
gcc samplec.c -lpdf417
```

In the case where the encoder library is not in the linker's default search path, use the `-L` option (with `gcc`) to indicate the directory name where the library files are installed. For example, if the encoder's library files were installed in `/opt/pdf417/lib`, the following would be necessary to compile and link the sample C program:

```
gcc -I/opt/pdf417/include samplec.c -L/opt/pdf417/lib -lpdf417
```

3.3 Linking with the shared library

When the installation script provided with the distribution is used to install the encoder library, the dynamic linker configuration utility is run to create the encoder library's shared object name (SONAME). This symbolic link is based on the library's major and minor version number (e.g., 2.2). Thus, when installing version 2.2.1 of the PDF417 encoder, the following libraries are installed (with the first being a symbolic link created by `ldconfig`):

```
libpdf417.so.2.2
```

```
libpdf417.so.2.2.1
```

Silver Bay Technologies assures API compatibility of its encoders for the same major/minor version number. This means that all version of 2.2.x will be compatible, but there is no guarantee that the API in version 2.3.0 would be compatible with any version of 2.2.x.

When linking with the shared object library, it is recommended to link with the shared object name (the symbolic link) created by `ldconfig`. Thus, when version 2.2.1 of the encoder is installed, it is recommended to link with the 2.2 shared object lib (`libpdf417.so.2.2`). To do this, you must provide the full library name to the linker (e.g `libpdf417.so.2.2`) using the `-l:` syntax. The fo

Assuming the shared object library has been installed in a directory that is part of the linker's default path, the following command could be used (with `gcc`) to compile and link the sample program provided:

```
gcc samplec.c -l:libpdf417.so.2.2
```

In the case where the shared object library is not in the linker's default search path, use the `-L` option (with `gcc`) to indicate the directory name where the library files are installed. For example, if the encoder's library files were installed in `/opt/pdf417/lib`, the following would be necessary to compile and link the sample C program:

```
gcc -I/opt/pdf417/include samplec.c -L/opt/pdf417/lib -l:libpdf417.so.2.2
```

4 Sample Program

A sample C program has been provided in the distribution tar file in the `pdfv2.2.1/samples` directory.

File	Contents
<code>samplec.c</code>	<p>This program uses the high-level APIs to encode a simple message. It then outputs the PDF417 symbol to files in a number of formats:</p> <ul style="list-style-type: none">• Font rendering• TIFF image rendering• Bitmap image rendering• Compact PDF417 generation (as a bitmap image)

Assuming the PDF417 encoder library and include files were installed on your system and are available to the C compiler and linker through their default paths, the sample program can be compiled and executed as follows:

```
cd pdf417v2.2.1/samples
gcc samplec.c -o samplec -lpdf417
./samplec
```

The program will create four files in the current directory:

- `pdfmnt` – this file contains the output of the font rendered; that is, the characters that when printed with one of the PDF417 fonts provided with result in the generation of the PDF417 barcode. The actual printing of this file, using the correct font is beyond the scope of this example program and document.
- `pdftif.tif` – this file contains a TIFF image (Tagged Image File Format) of the PDF417 barcode.
- `pdfdib.bmp` – this file contains a DIB image (Device Independent Bitmap), or just bitmap image of the PDF417 barcode.
- `pdftrc.bmp` – this file contains a DIB image of the same PDF417 barcode, but as a Compact (Truncated) PDF417. Compact PDF417 is discussed in more detail in the *PDF417 Encoder Programmer's Manual*.

The `pdfmnt` file can be viewed using any text editor (e.g., nano or vim), while the TIFF and BMP images require a compatible image viewer.