



# **Maxicode Encoder**

**Version 2.1.3**

## **Linux 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 20110223

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 © 2011, 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
<b>2</b>	<b>INSTALLING THE ENCODER .....</b>	<b>3</b>
2.1	INSTALLING THE ENCODER LIBRARY .....	3
2.2	INSTALLING THE FONTS .....	3
2.2.1	<i>AFP Fonts</i> .....	3
2.2.2	<i>HP-PCL Fonts</i> .....	4
2.2.3	<i>Xerox Fonts</i> .....	4
<b>3</b>	<b>PRINTING.....</b>	<b>5</b>
3.1	PRINTING USING DDS .....	5
3.2	PRINTING USING HP-PCL FONTS .....	6
3.3	PRINTING USING XEROX FONTS .....	7
<b>4</b>	<b>SAMPLE PROGRAMS.....</b>	<b>8</b>

# Silver Bay Software LLC Maxicode Encoder

Version 2.1.3

## Linux Manual

### 1 Introduction

#### 1.1 Contents of this Manual

This manual accompanies the Silver Bay Software LLC Maxicode Encoder version 2.1.3. 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 contains the following:

Root directories:

Directory	Contents
linux_32bit	Installation files for the 32-bit version of the encoder
linux_64bit	Installation files for the 64-bit version of the encoder
docs	Documentation
fonts	Font files
samples	Source code for sample programs

In the "linux\_32bit" and "linux\_64bit" subdirectories:

File	Contents
libmaxicode.a	Library file containing the object files for the encoder
maxiapi.h	Primary C include file
maxidefs.h	A related C include file
maxiEncode	Command line program that will perform encoding

Note that the .h files in the two subdirectories are actually identical – they are included twice simply for convenience. The libraries and executable files differ only in the method via which they were compiled – 32-bit mode or 64-bit mode.

In the “docs” subdirectory:

File	Contents
maxi_linux.pdf	This manual.
maxi_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.
maxi_lowlevel.pdf	An Adobe Acrobat file that describes the low-level API into the encoder. This describes the ability of the encoder to produce non-standard output.

In the “fonts” subdirectory:

File	Contents
MAXHP3P.FLJ	HP-PCL Portrait-mode font
MAXHP3L.FLJ	HP-PCL Landscape-mode font
X5PMAX.FNT	Xerox 5-word Portrait mode font
X5LMAX.FNT	Xerox 5-word Landscape mode font
X9PMAX.FNT	Xerox 9700 Portrait mode font
X9LMAX.FNT	Xerox 9700 Landscape mode font
COMAXI2A, T1MAXI2A, XOMAXI2A	240 DPI AFP font files
COMAXI3A, T1MAXI3A, XOMAXI3A	300 DPI AFP font files

## 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 Maxicode 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.

### 2.1 Installing the Encoder Library

The encoder is delivered in two versions, one for 32-bit use and one for 64-bit use. Aside from the way they are compiled, the two versions are identical.

In order to install the encoder, copy the `libmaxicode.a` file from the appropriate distribution subdirectory into an appropriate “library” directory on the system that is used to compile your application. This is frequently either the `/usr/lib` directory or a subdirectory thereof, however your individual system layout, and the way your developers compile your application, will determine the appropriate location.

Similarly, the `maxiapi.h` and `maxidefs.h` files must be copied to an appropriate “include” directory on your development system.

Note that none of these three files are required in order to run an application built using the encoder. Thus, if you have separate development and production machines, these files will typically be installed only on the development machine.

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 `libmaxicode32.a` and `libmaxicode64.a`. The `maxiapi.h` and `maxidefs.h` files are identical between the 32-bit and 64-bit version, and thus only one copy of these needs to be installed.

### 2.2 Installing the Fonts

Unlike the encoder library, the font files that are included in the `fonts` directory in the distribution must be installed in your production environment, as they must be available during the printing process.

How you install the fonts depends a great deal on your particular print system design. A discussion of all the various possibilities is beyond the scope of this document.

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 AFP Fonts

Two AFP fonts have been supplied: one for 240 DPI printers and the other for 300, 600, and 1200 DPI printers. There is generally no harm in installing both sets of fonts, even if your environment only uses one resolution.

Font files	Resolution
COMAXI2A T1MAXI2A XOMAXI2A	240 dpi
COMAXI3A T1MAXI3A XOMAXI3A	300 dpi

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

### 2.2.2 HP-PCL Fonts

There are two HP-PCL font files, one for portrait mode printing and one for landscape mode printing:

File	Contents
MAXHP3P . FLJ	HP-PCL Portrait-mode font
MAXHP3L . FLJ	HP-PCL Landscape-mode font

### 2.2.3 Xerox Fonts

Two types of Xerox 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
X5PMAX . FNT	Xerox 5-word Portrait mode font
X5LMAX . FNT	Xerox 5-word Landscape mode font
X9PMAX . FNT	Xerox 9700 Portrait mode font
X9LMAX . FNT	Xerox 9700 Landscape mode font

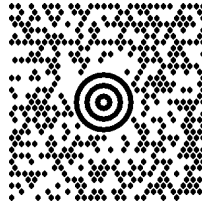
The appropriate fonts should be loaded onto your printer using standard Xerox procedures.

### 3 Printing

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

```
230303230331222231233222212222
331220321012103131311331133320
103223110030320012023121302010
222232322231222232313122132222
111230121331012200212131333030
013212320222000001013300013232
222222310000000000121202201012
120120230000000000021100222032
110020103500000000003023331130
303110011100000000101030021120
002012320020000000020110003020
222132110031322003111001301210
313310020202020202020100330130
130310213121202130311302121100
233332300112003211113112130220
111031202110230033010231310310
202002202020022020000020200200
```

Note there are 17 rows, 30 characters in each row. When printed using the supplied Maxicode font, the hexagons and bullseye of the Maxicode symbol are printed:



It is up to the application software and printing environment to cause the Maxicode 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 Maxicode 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 Maxicode symbol need to be printed at a nominal spacing of 16.5 lines per inch. However, most printing technologies will not let you specify 16.5 as a valid lines-per-inch setting.

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

#### 3.1 Printing using DDS

The `maxi_api.pdf` manual provides more detail on printing using the AFP fonts, however, the characters that make up the Maxicode symbol must be printed at a nominal spacing of 16.5 lines per inch in order for the symbol to be the correct size. Unfortunately, this is not a valid value for a DDS. Therefore, in order to print a Maxicode symbol using a DDS, the 17 lines of encoder output must be individual placed on the page using the `POSITION` function. NOTE: this requires that the printer file be created with `DEVTYPE(*AFPDS)`.

Consider the following DDS fragment:

```
A          R MAXIOUT          ENDPAGE
A          CDEFNT(X0MAXI3A)
A          MAXI01          30A  POSITION(1.700 1.35)
```



A	MAXI02	30A	POSITION(1.761 1.35)
A	MAXI03	30A	POSITION(1.821 1.35)
A	MAXI04	30A	POSITION(1.882 1.35)
A	MAXI05	30A	POSITION(1.942 1.35)
A	MAXI06	30A	POSITION(2.003 1.35)
A	MAXI07	30A	POSITION(2.064 1.35)
A	MAXI08	30A	POSITION(2.124 1.35)
A	MAXI09	30A	POSITION(2.185 1.35)
A	MAXI10	30A	POSITION(2.245 1.35)
A	MAXI11	30A	POSITION(2.306 1.35)
A	MAXI12	30A	POSITION(2.367 1.35)
A	MAXI13	30A	POSITION(2.427 1.35)
A	MAXI14	30A	POSITION(2.488 1.35)
A	MAXI15	30A	POSITION(2.549 1.35)
A	MAXI16	30A	POSITION(2.609 1.35)
A	MAXI17	30A	POSITION(2.670 1.35)

This DDS prints the Maxicode symbol with its upper left corner 1.70 inches down and 1.35 inches in on a page. Each of the successive 16 lines of output is manually positioned 0.0606 inches lower than the previous one (16.5 lines per inch, rounded to the nearest thousandth of an inch).

The CDEFNT function selects the font – in this case the X0MAXI3A font. The X0MAXI3A font is for use with 300, 600, and 1200 DPI page printers. If your printer is a 240 DPI printer, use the X0MAXI2A font.

### 3.2 Printing using HP-PCL Fonts

The design of Linux 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 Linux 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:

$^E_c *cnnD$  (where  $^E_c$  is the ASCII ESC character, and  $nm$  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 Maxicode font as the PCL secondary font

$^E_c )nnX$  (where  $^E_c$  is the ASCII ESC character, and  $nm$  is the font ID)

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

Step 4: Print the characters associated with the Maxicode symbol. Each individual line of the symbol must be positioned 0.060 inches below the previous line, thus achieving a spacing of 16.66 lines per inch (since HP-PCL is based on 300dpi units, this is as close to the nominal 16.5 LPI that can be achieved). 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 the  $yyy$  value will increase by 18 for each successive line.

Step 5: 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.

### **3.3 Printing using Xerox Fonts**

The design of Linux 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 Maxicode symbol, and to position each individual line of the symbol 0.060 inches apart (16.66 lines per inch).

## 4 Sample Programs

Five sample C programs have been provided. Source for all five are located in the `samples` directory in the distribution media.

- `cups.c` illustrates the UPS interface, which is the most common
- `cscm.c` illustrates the Structured Carrier Message interface
- `cstr.c` illustrates the SCM “string” interface
- `ceec.c` illustrates EEC interface
- `maxiEncode.c` is the source for a command line program that will accept command line parameters and produce encoded output.

Each of these programs can be compiled using a command line such as:

```
xlc -q[32|64] -qlanglvl=ansi -qarch=com samplefile.c libmaxicode.a
```

The first four sample programs do not take any command line arguments, and write their output to a file with a name similar to the program name.

The output consists of 17 lines of text, comprised of the digits 0 through 5. Here’s an example of what the file would look like:

```
230303230331222231233222212222
331220321012103131311331133310
103223110030320012023121302000
222232322231222232313122132220
111330120331012200213130333012
010132120012000001012032013220
222202120000000000121320100020
103001210000000000023122202202
201100103500000000003020233322
311030122200000000102102112122
300200303020000000033010302030
232221322131322003013111013002
331002020202020202020110331320
131131303030312020213130200230
113101313130103313111210312230
022130232232213122213300313000
022022200220200002002200002022
```

When sent to the printer using one of the provided fonts, rather than the digits seen above, a Maxicode symbol comprised of hexagons and the centering bullseye will be printed.

The final sample program, `maxiEncode.c` is provided in both source and pre-compiled form. (The compiled version is located in the `linux_32bit` and `linux_64bit` directories.) This command line program may be used as a standalone encoder – it takes its inputs from the command line, and produces the encoded output on `stdout`.

Installing the pre-compiled version of this program only requires copying it from the distribution media and then making sure the appropriate execution permissions are set. (These may be lost depending on how the distribution media is prepared.) if necessary, the permissions can be restored using the command:

```
chmod 755 maxiEncode
```

`maxiEncode` takes the following command line parameters:

Parameter	Meaning
<code>-p POSTALCODE</code>	Required. Destination postal code. (5 or 9 digits in US)
<code>-s SERVICECLASS</code>	Required. UPS Service Class code

Parameter	Meaning
-t TRACKINGNUMBER	Required. 10-character UPS tracking number
-c COUNTRYCODE	Optional. 3-digit country code (default: 840 = US)
-m MODE	Optional. 0-3 (default: 1 = UPS auto-select algorithm)
-n SHIPPERNUMBER	Optional. 1-6 char alphanumeric
-j JULIANDATEOF PICKUP	Optional. 1-366
-i SHIPMENTID	Optional. 1-32 characters
-x PACKAGENUMBER	Optional. Numeric - 'X' in 'Package X of Y'
-y PACKAGECOUNT	Optional. Numeric - 'Y' in 'Package X of Y'
-w PACKAGEWEIGHT	Optional. Numeric, in tenths of a pound
-v [Y N]	Optional. Address validation flag
-A SHIP-TO-ADDRESS	Optional. 1-36 characters
-C SHIP-TO-CITY	Optional. 1-36 characters
-S SHIP-TO-STATE	Optional. 2 alpha

These command line parameters correspond directly to the various fields taken by the C API. Consult the API manual for more details on their meaning.

Thus, for example, a sample invocation might look like:

```
maxiEncode -p 08846 -s 1 -t 1Z12345678
```

This would result in the following being sent to `stdout`:

```
230303230331222231233222212232
321220321012103131311331133100
323223110030320231213020110122
22222231313131313131313100232
010101212131212020210203012110
101010100312000001012010101000
020202200020000000003202020212
21212123000000000011321212120
101010103500000000003010101010
02020211030000000000102020222
21212123120000000111321212110
101010100030020023123110101002
020202020202020202023302232212
202121212131202020313101020232
000101332233000312112230210202
001030203313322010031103300000
200202002200022200020202000202
```

This program can be used to generate test samples, or can be used in a production environment, if appropriate.