Silver Bay Software LLC

Maxicode Encoder

Version 2.1.3 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

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

Table of Contents

Silver Bay Software LLC Maxicode Encoder

Version 2.1.3 for V7R1M0

AS/400 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 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
`max213s41.savf`	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V4R1M0
`max213s52.savf`	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V5R2M0
`max213s61.savf`	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V6R1M0
`max213s71.savf`	AS/400 save file (SAVF) that contains the encoder compiled for OS/400 V7R1M0

In the “`docs`” subdirectory:

File	Contents
`maxi_as400.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.

In the “`fonts/hp`” subdirectory:

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

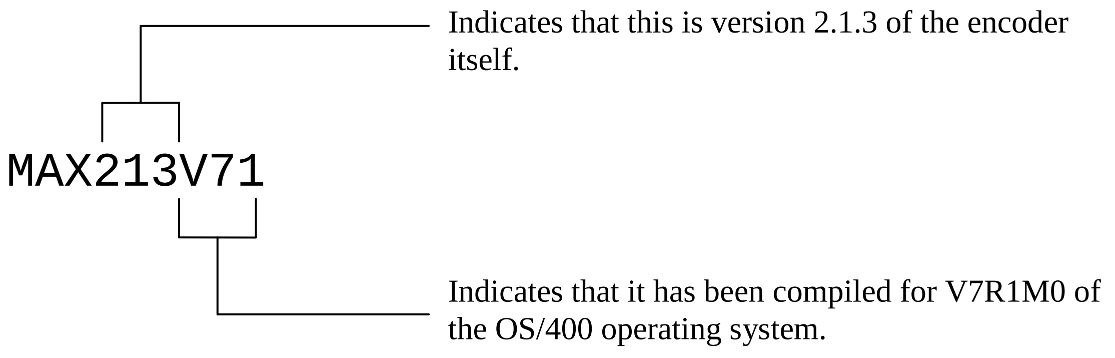
In the “`fonts/xerox`” subdirectory:

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

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 2.1.3, 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:



Indicates that this is version 2.1.3 of the encoder itself.

MAX213V71

Indicates that it has been compiled for V7R1M0 of the OS/400 operating system.

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 `MAX213V41`, while the variant compiled for V6R1M0 and later will have a library name of `MAX213V61`. Save files are named similarly, except that an “`S`” character replaces the “`V`” character.

In each case, the internal encoder version number is 2.1.3, 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. If you are upgrading from a prior Silver Bay Software version, or one of the NeoMedia Technologies 2.1 series of encoders (i.e. 2.1.0, 2.1.1 or 2.1.2):

   a. Install the new encoder library, using the steps listed in the next chapter.

   b. 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 Maxicode library with the newer library on your library list) however it is safest to explicitly rebind.

2. If you are upgrading from one of the NeoMedia Technologies 2.0 series of encoders (i.e. versions 2.0.0 through 2.0.9):

   a. Install the new encoder library, using the steps listed in the next chapter.

   b. NeoMedia Technologies made an API change between versions 2.0 and 2.1 of the encoder. *If you were using one of the 2.0 series of encoders, it will be necessary to update and recompile your source code to reflect this API change.* See the “Upgrade Notes” section of the `maxi_api.pdf` document for a description of the required change.

   c. Re-bind your applications to the service program that is part of the new library.

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 (particularly the 2.0 series), it is possible for both the new encoder and the older 2.0 encoder to co-exist on your system. Thus, if you have more than one application using the 2.0 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 Maxicode 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 Maxicode 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 Maxicode 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
`max213s41.savf`	V4R1M0 through V6R1M0.
`max213s52.savf`	V5R2M0 through V7R1M0
`max213s61.savf`	V6R1M0 and later.
`max213s71.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(SBSMAXI) TYPE(*TEST)
CRTSAVF FILE(SBSMAXI/MAX213S71)
```

This example assumes that you choose to upload the savefile into the SBSMAXI 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 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 SBSMAXI
binary
put max213s71.savf MAX213S71
```

The above assumes that your current directory is the savefiles directory on the CD, or that you have copied the save files from the CD into your current directory..

NOTE: 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 uploaded save file:

```
RSTLIB DEV(*SAVF) SAVF(SBSMAXI/MAX213S71)
    SAVLIB(MAX213V71)
```

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	max213s41.savf	MAX213S41	MAX213V41
V5R2M0	max213s52.savf	MAX213S52	MAX213V52
V6R1M0	max213s61.savf	MAX213S61	MAX213V61
V7R1M0	max213s71.savf	MAX213S71	MAX213V71

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
MAX213SRV	*SRVPGM	CLE	Service program containing the encoder API's. ILE programs using the encoder will bind to this service program.
MAXINIT	*PGM	CLE	Program implementing the MAXINIT API call for use by OPM programs.
MAXUPSN	*PGM	CLE	Program implementing the MAXUPSN API call for use by OPM programs.
H	*FILE	PF-SRC	Source physical file containing the "C" include files for the encoder API's.
QCBLLESRC	*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 Maxicode 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 your compile and install both fonts.

Font files	Resolution
C0MAXI2A T1MAXI2A X0MAXI2A	240 dpi
C0MAXI3A T1MAXI3A X0MAXI3A	300 dpi

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 Maxicode 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., C0MAXI2A and T1MAXI2A must be compiled before X0MAXI2A).

```
CRTFNTRSC FNTRSC(QFNT01/C0MAXI2A) FILE(MAX213V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/T1MAXI2A) FILE(MAX213V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/X0MAXI2A) FILE(MAX213V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/C0MAXI3A) FILE(MAX213V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/T1MAXI3A) FILE(MAX213V71/QFNTRSC)
CRTFNTRSC FNTRSC(QFNT01/X0MAXI3A) FILE(MAX213V71/QFNTRSC)
```

These fonts use the standard AFP font naming conventions. Thus, those are zeros in the file names, not capital 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
`MAXHP3P.FLJ`	HP-PCL Portrait-mode font
`MAXHP3L.FLJ`	HP-PCL Landscape-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
`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.

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 Maxicode 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(MAX213V71/MAX213SRV)
```

If the `MAX213V71` library has been added to the library list, you may instead wish to use the following:

```
CRTPGM PGM(MYLIB/MYPROGRAM) BNDSRVPGM(*LIBL/MAX213SRV)
```

4.1.2 OPM Programming Environment

Using the Maxicode encoder with an OPM language is a little different. An OPM program cannot directly call an ILE service program. Therefore, an interface program (`MAX213V71/MAXUPSN`) has been provided. This is a “stub” C program that, when invoked, in turn calls the `MAX213SRV` service program, since. The `MAXUPSN` program was linked using the following command:

```
CRTPGM PGM(MAX213V71/MAXUPSN) MODULE(MAXUPSN)
    BNDSRVPGM(*LIBL/MAX213SRV) ACTGROUP(*CALLER)
```

There is a subtle implication here; the `MAX213SRV` service program was linked from the library list. Thus, if you are using an OPM language, the `MAX213SRV` service program **must** be in a library that appears in the job’s library list when it executes. Thus, you may either add the `MAX213V71` library to the job’s library list, or you may copy the `MAX213SRV` service program into a library that is in the job’s library list.

A second OPM program (`MAX213V71/MAXINIT`) is provided to handle the optional initialization call. The same comments apply to this program as to the `MAXUPSN` program.

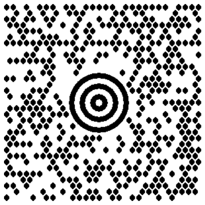
Note for advanced programmers: The Maxicode API provides a number of advanced function calls other than the `MAXINIT` and `MAXUPSN` calls. Only these two calls are available to OPM programs, however. If you have a need for other Maxicode API functions in an OPM environment, contact Silver Bay Software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

5 Printing

The encoder software returns a block of characters to the caller. If you were to examine the output of the 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.

5.1 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 that 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.

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:

$^{E}{}_{C}$`*c`*nn*`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 Maxicode font as the PCL secondary font

$^{E}{}_{C}$`)`*nn*`X` (where $^{E}{}_{C}$ is the ASCII ESC character, and *nn* 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:

$^{E}{}_{C}$*p*xxx*X$^{E}{}_{C}$*p*yyy*Y (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.

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

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 MAX213V71 library.

6.1 Sample Printer File

The three sample programs use the same printer file, MAX213V71/MAXICODE. Before you can compile any of the sample programs, you must create the printer file.

The source for the printer file is MAX213V71/QDDSSRC(MAXICODE). This is a very simple DDS that prints a Maxicode symbol using the supplied font. Note that this DDS is set up to use the 300 DPI font. The following line appears in the DDS:

```
CDEFNT(X0MAX3A)
```

This will work for 300, 600, and 1200 DPI IPDS printers alike. However, if you have a 240 DPI printer, you must change the font to X0MAXI2A:

```
CDEFNT(X0MAX2A)
```

To create the printer file, you must use a device type of *AFPDS:

```
CRTPRTF FILE(MAX213V71/MAXICODE)
    SRCFILE(MAX213V71/QDDSSRC) SRCMBR(MAXICODE)
    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 MAX213V71/MAXUPSN program. The source for the sample program is in MAX213V71/QLBLSRC(CBLUPS). It can be compiled using the following command:

```
CRTCBLPGM PGM(MAX213V71/CBLUPS) SRCFILE(MAX213V71/QLBLSRC)
    SRCMBR(CBLUPS) REPLACE(*YES)
```

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

```
CALL CBLUPS
```

Remember that, because of the way that the MAXUPSN program is linked, the MAX213V71 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 MAX213V71/MAX213SRV service program. The source for the sample program is in MAX213V71/QCBLLESRC(CBLUPS). You must first compile the COBOL module:

```
CRTCBLMOD MODULE(MAX213V71/CBLUPS)
    SRCFILE(MAX213V71/QCBLLESRC)
    SRCMBR(CBLUPS)REPLACE(*YES)
```

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

```
CRTPGM PGM(MAX213V71/ILECBLUPS) MODULE(MAX213V71/CBLUPS)
    BNDSRVPGM(MAX213V71/MAX213SRV) REPLACE(*YES)
```

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

```
CALL ILECBLUPS
```

6.3 RPG Samples

6.3.1 Compiling and Running the Sample RPG/400 Program

The sample RPG/400 program calls the MAX213V71/MAXUPSN program. The source for the sample program is in MAX213V71/QRPGSRC(RPGUPS). It can be compiled using the following command:

```
CRTRPGPGM PGM(MAX213V71/RPGUPS) SRCFILE(MAX213V71/QRPGSRC)
    SRCMBR(RPGUPS) REPLACE(*YES)
```

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

```
CALL RPGUPS
```

Remember that, because of the way that the MAXUPSN program is linked, the MAX213V71 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 MAX213V71/MAX213SRV service program. The source for the sample program is in MAX213V71/QRPGLESRC(RPGUPS). You must first compile the RPG module:

```
CRTRPGMOD MODULE(MAX213V71/RPGUPS)
    SRCFILE(MAX213V71/QRPGLESRC) SRCMBR(RPGUPS)
    REPLACE(*YES)
```

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

```
CRTPGM PGM(MAX213V71/ILERPGUPS) MODULE(MAX213V71/RPGUPS)
    BNDSRVPGM(MAX213V71/MAX213SRV) REPLACE(*YES)
```

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

```
CALL ILERPGUPS
```

6.4 ILE C Sample

The sample ILE C program links with the `MAX213V71/MAX213SRV` service program. The source for the sample program is in `MAX213V71/QCSRC(CUPS)`. You must first compile the C module:

```
CRTCMOD MODULE(MAX213V71/CUPS) SRCFILE(MAX213V71/QCSRC)
    SRCMBR(CUPS) REPLACE(*YES)
```

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

```
CRTPGM PGM(MAX213V71/CUPS) MODULE(MAX213V71/CUPS)
    BNDSRVPGM(MAX213V71/MAX213SRV) REPLACE(*YES)
```

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

```
CALL CUPS
```

6.5 Output of the Sample Programs

All of the sample programs generate a single Maxicode symbol to the output spooler. If you view the spooler, you should see 17 lines of text, comprised of the digits 0 through 5. Here’s an example of what the spool file would look like:

```
230303230331222231233222212222
331220321012103131311331133310
103223110030320012023121302000
222232322231222232313122132220
111330120331012200213130333012
010132120012000001012032013220
222202120000000000121320100020
103001210000000000023122202202
201100103500000000003020233322
311030122200000000102102112122
300200303020000000033010302030
232221322131322003013111013002
331002020202020202020110331320
131113130303031202021313020023O
113101313130103313111210312230
022130232232221312221330031300O
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.

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