Implementing PER/ZIP4

For Programmers Only

WorksRight Software, Inc.
Phone 601-856-8337
Fax 601-856-9432
E-mail *software@worksright.com*

Disclaimer

Although the sample code provided with PER/ZIP4 has been thoroughly tested by the author, its use within your applications must be carefully controlled and tested. This material is provided as a courtesy to users of PER/ZIP4 on an "as is" basis without warranty, either expressed or implied.

Adding PER/ZIP4 Functions to User-Written Software

PER/ZIP4 contains many callable programs (some people prefer to call them API's) that allow you to add PER/ZIP4 functions to your own programs. The sample source code and descriptions for all these programs are in source file QSMPLSRC in library MLLIBR.

QSMPLSRC contains source code to aid you in the following areas:

Interactive postal coding
Browsing the postal database
Retrieving information about a particular ZIP Code.

We suggest you start PDM and use the print option to print a copy of all the members in QSMPLSRC in library MLLIBR. This will allow you to easily review all the possibilities provided by PER/ZIP4.

MULIB Mailing Utility Library

In addition to QSMPLSRC, PER/ZIP4 also includes another library of sample source code called MULIB. The source code for MULIB is stored in source file QMULIBSRC in library MLLIBR.

To create MULIB, compile CL program #CRTMULIB from the source member of the same name in QMULIBSRC in MLLIBR, then call the program using CALL MLLIBR/#CRTMULIB. This program will create library MULIB and compile all the programs needed for MULIB to function.

MULIB contains sample programs which use much of the code from QSMPLSRC to create actual working programs.

A mailing file named DATAFILE is used in MULIB for demonstration purposes. If you wish to use menu option five in MULIB to postal code DATAFILE, you must first enter a file definition using option four on the PER/ZIP4 main menu.

Use #FILEDEF source member in source file QMULIBSRC as an example of what the definition for DATAFILE should be.

Access to PER/ZIP4

We strongly suggest you add some sort of control record to your application software with a "switch" so that you can control access to PER/ZIP4. If for any reason you need to discontinue use of PER/ZIP4, simply changing the value of the switch should do the trick. Your user-written software can revert to its pre-PER/ZIP4 method of operation.

Library List

In order to call many of the PER/ZIP4 functions, it will be necessary to add MLCTRL and MLLIBR to the user's library list. These libraries may be added to the system library list or they can be added dynamically in the CL program leading to the call of the PER/ZIP4 function.

As PER/ZIP4 functions are called, a third library, ML*yymmv* (yy=year, mm=month, v=version), will be automatically added to the user's library list. This library's name will be unique for each month's PER/ZIP4 update. **Do not attempt to manually add this update specific library to your user's library list. PER/ZIP4 must manage this library dynamically.**

Using the STRPERZIP4 command manages the library list as well as other "housekeeping" chores. Use STRPERZIP4 for access to PER/ZIP4 except in those cases where you are using the callable programs with specific functions.

Important Note

CASS certification is provided only through the batch update program. Postal coding interactively simply codes one address record at a time. No postal form 3553 is produced and thus no CASS certification.

Address Verification

The number one feature most users want from PER/ZIP4 is interactive address verification. The logic behind this is fairly simple. If PER/ZIP4 can postal code an address, the odds are that it is a good address. If PER/ZIP4 cannot postal code an address, the odds are it isn't a good address. This is true even if the mail has been and is being delivered to that address.

The program to use for interactive address verification is ML219403. (Sample source code is found in member ML2194.RPG in source file QSMPLSRC in library MLLIBR.) This program uses no screen and is a yes/no situation. Either the address codes or it doesn't. If it codes, a standardized address, ZIP Code, ZIP+4, delivery point, and carrier route code are returned. The error code field will be blank.

If ML219403 can't code the address, the parameters will be returned unchanged, and the error code field will contain a three-character code. Also, there is an 80 character message field which will contain a description of the error code.

ML219403 uses a default length of 64 for the address line parameters. Parameter ADRL## allows you to optionally specify a desired length for the returned address lines.

Some standardized addresses may exceed the length of your original address line fields. You should evaluate the length of the returned information before you replace the original information.

If an address can't be coded, you should redisplay the address and the error message such as "Street name not found." There is a good likelihood that the user can review the information and make a change that will allow the address to be coded.

If the address can't be coded, you have two options. You can allow the user to override the error and update the record using a function key. If you do this, you may want to create an audit trail. Any address that is overridden should be written to a special file for later review. This will allow you to identify any "lazy" operator who is overriding the edit without making a good faith effort to get the correct addresses. Obviously, there are going to be some addresses that can't be coded. This audit file will allow detailed research at a later date of any "problem" addresses.

The second option is to allow the operator to search the postal database from a user program. If ML219403 can't code the address, give the operator a function key to call ML219303.

ML219303 is essentially the same program as the interactive postal coding program (menu option three on the PER/ZIP4 main menu). By giving parameter SCRN## the "S" (search) option, the program will immediately begin the database search.

Although this program allows function key 12 to return a selected address to the calling program, we do **not** recommend this action. Make the operator type the correct information and press ENTER again.

Special Considerations for Web Applications

The PER/ZIP4 callable program interface is designed to function as efficiently as possible. When a PER/ZIP4 callable program is called for the first time, program initialization is performed, the library list is set, and files are opened. To reduce overhead and speed up processing, these three steps are not performed on second and subsequent calls.
Since web application requests aren't always processed by the same OS/400 job, second and subsequent calls or requests may be processed by a different OS/400 job where the library list has not been properly set.

If you wish for the callable program to perform program/file initialization and library list management each time it is called, there is a way. After calling the callable program and passing in your address for processing, you can immediately call the callable program again with "EOJ" in the ECOD## field. This tells the PER/ZIP4 callable program to close the program and files and return control to the calling program. The next time your application calls the PER/ZIP4 callable program passing in an address for processing, program/file initialization and library list management will be performed.

An alternative would be for your application to call our library list management program MLLIBR/MLLIBL immediately before every call to the PER/ZIP4 callable program. The library list management program will set the library list every time it is called.

The call to the library list management program, MLLIBR/MLLIBL, must include two parameters. The first parameter is 10 characters in length and should always contain the value '***SETUP** '.

The second parameter is 5 characters in length and should indicate the installed version of PER/ZIP4 that you wish to access. To access the installed TEST version of PER/ZIP4, the second parameter should contain the value '***TEST**'. To access the installed PRODUCTION version of PER/ZIP4, the second parameter should contain the value '***PROD**'. To access the installed PREVIOUS version of PER/ZIP4, the second parameter should contain the value '***PREV**'.

For example, with MLCTRL and MLLIBR in your library list use the following call command to add the PRODUCTION dated library to your library list:

```
CALL PGM(MLLIBR/MLLIBL) PARM(*SETUP *PROD)
```

A Reminder

You should keep in mind that when calling any of the PER/ZIP4 programs, the action involved is nothing more than a call with a parameter list.

The manipulation of the parameters before and after the call is up to you. You must evaluate the results of the call and apply the results as indicated by your evaluation.

Interactive Postal Coding

Although you may want to give users access to interactive postal coding as provided by option three on the PER/ZIP4 main menu, you may not want to give them access to the PER/ZIP4 main menu. The interactive option can be called using the sample CL code found in ML2193.CL. This will allow you to call the interactive postal coding feature from some other user-written menu or program.

Batch Processing From A User Program

Another common situation is where you want to incorporate PER/ZIP4's batch processing option into a user-written routine. Program ML2190RN is the program to use. Sample CL code for calling ML2190RN is found in member ML2190.CL in source file QSMPLSRC in library MLLIBR.

The LDA (Local Data Area) is used to pass printer override information to program ML2190RN. You may need to make additional adjustments in your CL program to save the LDA before calling ML2190RN and restore the LDA after ML2190RN ends.

POSTNET Barcoding

PER/ZIP4 does not print the POSTNET barcode. What it does do is provide you with the information and tools you need to print the POSTNET barcode within your own applications.

We generally include a USPS Publications CD-ROM with each copy of PER/ZIP4 we ship. On the USPS Publications CD-ROM you will find a copy of publication 25 (Designing Business Mail). This is a publication from the US Postal Service which contains useful POSTNET barcode information. Please refer to publication 25 for additional information.

The POSTNET barcode consists of twelve characters. The five-digit ZIP Code, the ZIP+4, and the delivery point code make up the first eleven characters. The twelfth character is a correction digit. Most POSTNET barcode printers automatically add the correction digit.

When PER/ZIP4 postal codes an address, all these elements are provided as individual fields. You must combine them within your program into a single field to be fed into the barcoding routine in your software.

There are several methods of printing POSTNET barcodes. The method you chose must match your printer (hardware) and your application software.

The easiest method uses the DDS keyword BARCODE with the POSTNET parameter. This method requires a printer configured as IPDS with AFP enabled. If you can't use the DDS keyword, then you must use one of the other methods. MULIB contains extensive examples for all the common POSTNET methods.

Barcoding is too extensive a subject to be covered in detail in this discussion. However, we are confident that following the examples we provide will allow you to add POSTNET to your mailing pieces with a reasonable amount of effort.

Intelligent Mail Barcode (IMB) Barcoding

Just like the POSTNET barcode, PER/ZIP4 does not print the Intelligent Mail Barcode (IMB). The actual printing of the barcode (POSTNET and/or IMB) is a function of your print program.

The IMB is comprised of a Tracking portion and a Routing portion.

The Tracking portion of the IMB is made up of four fields:

1.) A 2-digit Barcode Identifier
2.) A 3-digit Special Services Code
3.) A 6-digit Customer Identifier
4.) A 9-digit Mailpiece Identifier

The Routing portion of the IMB is made up of three fields:

1.) The 5-digit Zip Code for the delivery address
2.) The 4-digit +4 for the delivery address
3.) The 2-digit Delivery Point Code for the delivery address

As you can see, the routing portion of the IMB consists of the same data used to print the POSTNET barcode. This information is provided by PER/ZIP4.

The tracking portion of the IMB, on the other hand, consists of new, mailing specific information. This information cannot be provided by PER/ZIP4. It must be provided by a mail presorting solution like our HyPER/Sort product.

HyPER/Sort does support the IMB and is capable of outputting all of the information needed for the tracking portion of the IMB.

Once PER/ZIP4 and HyPER/Sort have provided the data necessary to print the IMB, your print application will need to print the IMB on your label/mail piece the same way it would have printed the POSTNET barcode.

There are two methods of printing the Intelligent Mail Barcode(IMB). The easiest method uses the DDS keyword BARCODE with the parameters (34 X'03' *NOHRI). Example: **BARCODE(34 X'03' *NOHRI).** This method requires a printer configured as IPDS with AFP enabled. It also requires that your printer's IPDS firmware be up to date as the IMB barcode is relatively new.

If you can't use the DDS keyword, then you can use a special encryption module and downloadable font provided by the USPS to print the IMB barcode. Of course, to utilize this technique for printing the IMB, your iSeries printer must support downloadable fonts.

The USPS encryption module, USPS4CB, and the special USPS Fonts for the iSeries are included with PER/ZIP4. The encryption module can be used to encode or convert the 31 digit IMB data to a 65 character string. The special USPS Fonts can then be used to print the 65 character string in the form of the IMB.

To access the USPS encryption module and special fonts, you will need to unpack the IMB sample objects using the following steps:

1. If you haven't already created the Mailing Utility library, MULIB, use the following steps to create the library:

   A. Compile the program #CRTMULIB. The source code for this CL program can be found in MLLIBR/QMULIBSRC.

   B. Call the program #CRTMULIB. This program will create the Mailing Utility library, MULIB.

2. Next, you will need to unpack the IMB sample objects (sample program, fonts, and encoding module). To unpack the IMB sample objects, use the following steps:

A. Compile the program #CRTIMBOBJ. The source code for this CL program can be found in MULIB/QMULIBSRC.

B. Call the program MULIB/#CRTIMBOBJ. This program will unpack the IMB sample program, fonts, and encoding module.

After unpacking the IMB sample objects, you can study the sample program and its use of the encoding module. The sample program source code can be found in source file MULIB/QRPGLESRC.

NOTE: the source file MULIB/QRPGLESRC does not exist until you unpack the IMB sample objects.

Barcode Quality

In addition to containing the correct information, the Intelligent Mail Barcode (IMB) and POSTNET barcodes must be of a high print quality. You should consult with your local post office about submitting a sample for barcode evaluation. There can be a severe postage penalty for barcoded mail that does not achieve the required readability percentage.

PER/RDI

PER/RDI is an optional add-on module for PER/ZIP4. Utilizing USPS® RDI data, PER/RDI provides PER/ZIP4 the ability to return a Residential Delivery Indicator (RDI™) while processing your addresses. If PER/RDI is not installed on your system, the fields related to it will simply be blank.

When you purchase a PER/ZIP4 license, PER/RDI is available to you at no additional charge. However, to activate PER/RDI, you must obtain a license for the RDI data directly from the U.S. Postal Service®. Forms and instructions are available on our website to assist you in the process of licensing the RDI data.

Although the USPS uses the term Residential Delivery Indicator (RDI), the software and data are used to identify businesses. Therefore, any address not identified as a business by the USPS, by default, is identified as a residence.

Due to restrictions on the type of information that can be released regarding Post Office Boxes, RDI data does not distinguish between residential and business Post Office Boxes. Consequently, all Post Office Boxes are identified as residential.