

The Good, The Bad, The Ugly 2004

ADVERTISER INDEX

THE GOOD, THE BAD, THE UGLY

This month, we look at examples of clear SQL coding, the importance of using self-documenting code, and properly coding a date routine.

— Paul Morris

Neat Coding

The trouble with free-format code — whether it's CL, /Free, or embedded SQL — is that you're free to make the code totally unreadable (as I've seen many programmers do). So it's nice to see a bit of clear SQL code, as this example shows:

```
c/exec sql
c+ select count(*) into :w#count from cjp165
c+ where
c+     cono65 = :1#cono and locd65 = :locd#d
c+     and part65 = :part#d and pfhr65 <> :pfhr#d
c/end-exec
```

Would I have done the same? Possibly, although I tend to start each clause on a separate line like this:

```
c/exec sql
c+ select count(*) into :w#count
c+ from cjp165
c+ where
c+     cono65 = :1#cono and locd65 = :locd#d
c+     and part65 = :part#d and pfhr65 <> :pfhr#d
c/end-exec
```

Poor Field Names

Here's a snippet from a converted RPG program I worked on recently. The D-specs are from a definition of the LDA. Without going back to the calling program that set up the LDA, I have no clue what aaaaaa and bbbbbb were meant to be.

```
d item1#           513   527
d aaaaaa          519   535
d strc1#          528   529

d bbbbbb          s           Like(aaaaaa)

c           move      aaaaaa      bbbbbb
```

There's no excuse for such field names. Had the programmer used field names such as PART#L and PART#V, they would have been much clearer. Using self-documenting code is good coding practice.

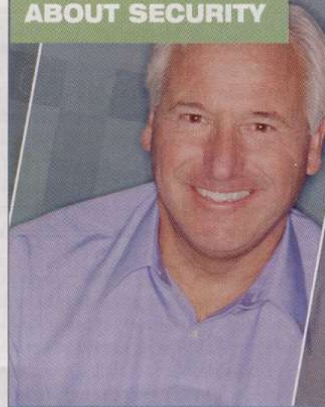
Date Processing

I'm currently working on an old implementation of JBA(Geac) System/21. Like many sites, its core has been hacked and bits have been bolted onto it. The core

FIGURE A
Example of bad coding for a date routine

```
c           Z-add      FromDate      @DTIEX
c           Exsr      @DTI
c           Z-add      @dtiin
FromCYMDdate
```

**OUR CUSTOMERS
DON'T WORRY
ABOUT SECURITY**



**ZMOD EXCHANGE
IS THE MOST
SECURE FTP
FOR THE I SERIES**

With more FTP installations than any competitor, we know a thing or two about providing powerful, secure connections between our customers and their trading partners. ZMOD Exchange FTP is a high-performing solution with features like:

- More encryption options than any software: PGP, S/MIME Implicit/Explicit SSL and TLS
- Firewall navigation tools
- Automation of any FTP session in 1 script
- Command line error checking
- Unmatched formatting options: DB2 ↔ XML, CSV, zip files, etc.

TRAILBLAZER
SYSTEMS
The Power to Exchange Any Transaction—Any Way

770-850-6966
www.trailblazersystems.com

system is written in good old RPG III, and many of the later programs are written in RPG IV — several of which were developed from converted RPG III code.

As with other packages, JBA provides date manipulation routines that can, for example, convert the internal seven-digit format to an external six-digit format in the form of YYMMDD, MMDDYY, or DDMMYY (a field in the LDA specifies which of these is used). To accomplish this date conversion, a subroutine is copied into the source code, and work fields are used to set and retrieve the dates when the subroutine is executed.

My customer converted the date routines to RPG IV without thought to using them properly (making them true ILE routines). Figure A shows an example of bad coding, which the customer had to perform each time he needed to convert a date. In Figure B, you see an example of good coding practice that improves the date routine. This example uses the existing converted copy member and, hence, preserves the existing logic.

FIGURE B
Example of good coding for a date routine

```

The prototype:
d ExToInDate    PR          7 0
d ExternalDate  6 0 value

The procedure:
P ExToInDate    b          7 0
d ExToInDate    pi          6 0 value
d ExtDate       Z-add      ExtDate  @DTIEX
c               Exsr       @DTI
c               Return     @dtiin
/ Copy oslcopysrc/qrpglesrc,@DTIAC
P               e

The function call (how we convert dates in the program):
C               Eval      FromCymDate =
ExToInDate(FromDate)
    
```

the remote IP device is always used. However, I want to use port 3002.

A You specify the port to use when you create the printer device rather than specifying it at the output-queue level. The Port parameter on command CrtDevPrt (Create Device Desc Printer) specifies the port.

— *iSeries NEWS technical staff*

DETERMINE THE VALUE OF BIFS

Q When debugging my RPG IV program, how can I see the value of fields %EOF, %Found, and so on?

A You can't display the value of %EOF, %Found, and such with the debugger because they're not fields. You can infer their value based on the execution path your program takes when you condition your code based on the value of one of these functions. You can also assign the value of these functions to a one-byte character field (using an Eval

operation) and then specify this character field as the target of a debug operation. After you've assigned a value, the char-

acter field will contain '0' or '1', reflecting the value of the assigned function.

— *iSeries NEWS technical staff*

If you think your tip or question is of interest to other readers, please e-mail it to us at techcorner@iSeriesNetwork.com

iSeries E-mail, Fax, PDF

KeyesFax

- Send selected spool file pages as E-mails
- Fax AFPDS, PCL or SCS spool files with overlays
- Fax Queue for tracking and control faxes.
- Auto report bursting

KeyesMail

- Burst spool files into PDF, RTF, TIFF attachments
- Access e-mail from any browser or terminal
- Spool file distribution as messages or attachments
- Connect to any mail server

KeyesPDF

- Convert spool files into images or PDF Docs
- Work with any type of spool file *SCS, *USERASCII, *AFPDS
- Auto indexing to IFS
- Create text based PDF documents from PCL print files

New!

KeyesOverlay

- Map spool files to full color PDF

Computer Keyes

25 Years!

1978-2003

www.ckeyes.com
info@keyes.com

Free 30 Day Trial!
USA: (800) 356-0203
Tel: (425) 776-6443

THE GOOD, THE BAD, THE UGLY

This month we look at the difficulty of reading files by record format name and how to tidy up code containing ELSE.

— Paul Morris

FILES AND FORMATS

I came across this snippet recently:

```
c   KeyInvoice   Setll   0er70
c   KeyInvoice   Reade   0er70
c                               Dow    Not(%EOF(0EP70L01))
```

I've never liked reading files by record format name — this dates back to the old days when you could have multiple format logical files (and who uses those now!). If you have the same physical file represented by different logical files in a single program, you need to refer back to the F-specs to see which is which. The above example with the %EOF makes no sense when you try to read by format name.

In my early days as a Cobol programmer, the rule was *read a file, write a record*, and this approach works for RPG. I'd code the above snippet as

```
c   KeyInvoice   Setll   0EP70L01
c   KeyInvoice   Reade   0EP70L01
c                               Dow    Not %EOF(0EP70L01)
```

SETTING JAVA CLASSPATH IN QSHELL

Q We've been trying to compile the DirectoryList example program in QShell (see "Augmenting QShell Utilities," June 2000, www.iSeriesNetwork.com, article ID 7404), but we can't get the classpath to hold both the access package path and the default system package path. When we change the classpath for the access package, the compiler can't find the default system package. How can we overcome this?

A You can use the following command in QShell to compile the DirectoryList.java file:

```
javac -classpath /qibm/proddata
/HTTP/Public/jt400/lib/jt400.jar
/myjava/DirectoryList.java
```

www.iSeriesNetwork.com

UNNECESSARY ELSE CODE

You may wonder what is wrong with this code:

```
C   Key_INp35   Chain   Parts
C                               If    not(%found(Parts))
C                               eval  *in62 = *on
C                               eval  Pard#d = *blanks
C                               LeaveSr
C                               Else
C                               eval  Pard#d = pdes35
C * Get Extension
C   Key_INp35   Chain   INP035L1
C                               Endif
```

Well, nothing is logically wrong with it, but it's a little untidy. After LEAVESR, nothing else in the code that follows would get executed, so why not finish the IF statement with the ENDIF at that point? I would code the above snippet as follows:

```
C   Key_INp35   Chain   Parts
C                               If    not %found(Parts)
C                               eval  *in62 = *on
C                               eval  Pard#d = *blanks
C                               LeaveSr
C                               Endif
C                               eval  Pard#d = pdes35
C * Get Extension
C   Key_INp35   Chain   INP035L1
```

This approach makes for a smaller, tighter IF grouping. In a small piece of code such as this, it doesn't make much difference, but when you apply this method to more complex structures, it aids readability. In this example, I'd also code defensively and put a check on the second chain in case a record that should be there isn't.

The classpath option provides the path to the IBM Toolbox for Java (the jt400.jar file), and I've path-qualified the DirectoryList.java source file. If your source file is in a directory called something other than /myjava, change the command appropriately.

This approach works on my iSeries running OS/400 V5R2 and JDK 1.4, but compiler behavior can vary depending on JDK release levels. Make sure that you're using a JDK at least as current as 1.2 and that you've applied the latest Java Group PTF. The JDK 1.2 release introduced the

"bootstrap classpath," a separate classpath for system packages. Before this JDK release, it was a challenge to juggle both system and user classpath entries, and this may explain the difficulty you're experiencing.

For more information about compiling and running Java code on the iSeries, including the role of the user, bootstrap, and extension classpaths, see the second edition of *Java and the AS/400* (2002, 29th Street Press).

— Dan Darnell ■

AVAILABLE LINUX 5250 EMULATION PRODUCTS

Q I'm looking for a 5250 emulation program for Linux. Does IBM offer one? Are there others that are available on the Web? Ideally, it would be great if the product were free.

A IBM indeed offers a Linux-based 5250 emulation program as part of its new iSeries Access for Linux product.

You can download iSeries Access for Linux at www-1.ibm.com/servers/eserver/iseries/access/linux. iSeries Access for Linux is licensed exactly as the other members of the iSeries Access family, and you must buy a client license in order to connect the 5250 emulation to the iSeries. In addition, IBM's iSeries Access for Web product can provide 5250 emulation for a Linux system using the Netscape browser.

A couple of free 5250 emulation desk-

top products for Linux are also available as well as one reasonably priced shareware product. You'll find one open-source product at <http://tn5250.sourceforge.net>, and another open-source tn5250 emulator at <http://linux.tucows.com/preview/8731.html>. While I haven't used either of these, I have used the TN5250 product from MochaSoft (www.mochasoft.dk). This product isn't free (it costs \$25), but it's small, high quality, and works well.

— Michael Otey ■

THE GOOD, THE BAD, THE UGLY

by Paul Morris

Overrides in a Called Program

In a suite of programs I was reviewing before implementation, an RPGLE program called a CL program (not CLLE) with this code before opening a work file:

```
CRTDUPOBJ OBJ(XYP25) FROMLIB(*LIBL) OBJTYPE(*FILE) +
TOLIB(QTEMP) NEWOBJ(*SAME) DATA(*NO)
```

```
OVRDBF FILE(XYP25) TOFILE(QTEMP/XYP25)
```

The program happened to work because QTEMP was higher in the library list than the database library (but this could not be guaranteed).

If the CL program had instead called the RPGLE program after doing the override, the override would have worked. But OVRDBF in this form is "invocation sensitive," so the override is removed when the called CL program returns.

To fix this particular problem, I used the following code:

```
OVRDBF FILE(XYP25) TOFILE(QTEMP/XYP25) OVRSCOPE(*JOB)
```

The *JOB option on the override keeps the override active for the entire job, including all activation groups and invocations. I could have employed other approaches, such as doing the override in the RPGLE program or changing the activation groups of the programs, but this was a quick fix (the program was going live that day) that suited the development environment. The help text for OVRDBF provides more details about the OVRSCOPE options.

Making a Comment in Code

How many times has this happened to you? You're reading down an RPG program in fixed-format layout trying to understand what is happening, but it doesn't make sense. Then, you try debugging the program and find the debugger skips a line. Why? Because the programmer commented out the line with a single asterisk character (*) in column 7, and you missed it when you originally reviewed the code. A single * to comment out a line is poor coding practice that causes more problems later on.

I prefer either of the two following methods of commenting out code:

```
c*PM01 -----if----- fldttl03 <> *blanks
c                               eval      wrkdta1 = %trim(wrkdta1) + ',' + fldttl03

or

c*PM01                               if      fldttl03 <> *blanks
c                               eval      wrkdta1 = %trim(wrkdta1) + ',' + fldttl03
```

where PM01 is a change reference that matches a comment in the program header.

THE GOOD, THE BAD, THE UGLY

by Paul Morris

SETTING YOUR OWN PDM OPTIONS

Not everyone is aware that you can set up your own PDM options in a development environment. These options let you specify your own code to use against a PDM list in the same way that "2" invokes the editor. Simply use F16 from a PDM screen to get a list of existing options that you can change or add to. Figure A shows a simple command entered using the F6 option.

My favorite command to use is

```
RG  RGZPFM &L/&F &N SRCOPT(*DATE *SEQNBR)
```

where &L is the library of the list that PDM is showing, &F is the file name (if you're working with a member list), and &N is the name of the item, which is a member name in a member list or a file name in a file list. There are many other codes as well; the F1 help function details them all.

Once I've finished developing a program, I use the above command to reorganize the new program's source member to ensure that the program is "clean" before I initially promote it to production.

USING NULL PROTOTYPES

It's possible to have a prototype that simply sets a return value without having any parameters passed to it. The following code snippets show an example; the original code used global variables from a file being processed and date checking to determine the value to return:

The prototype definition:

```
d BackCat      pr          3
  * No parameters to pass!
```

Its use in the C-specs:

```
* NB BackCat is a procedure
c          Eval          OrdType =
BackCat
```

The procedure at the end of the source code:

```
p BackCat      b
d BackCat      pi          3
d w#ordtyp     s          3
:
* application coding
:
c          Return      w#ordtyp
p          e
```

You can even use this technique to provide a constant value to items such as pi (3.14159...) with the procedure held in a

```
Create User-Defined Option
Type option and command, press Enter.
Option . . . . . BY Option to create
Command . . . . . Signoff
```

FIGURE A
Entering a command with F6

Some options are predefined, such as

```
RQ  RQ  runqry *n &L/&n
RS  RS  RUNQRY QRYFILE((&L/&N))
RCDSL(*YES)
SD  STRDBG PGM(&L/&N) UPDPROD(*YES)
```

You can set your own options by changing the defaults on the F18 screen. Be careful if you move to another iSeries, as some of the predefined options may have been changed.

service program. But the benefits of employing this method over using /COPY to copy a constant value are dubious.

As this trivial example shows, you can have a procedure that does nothing to any variables:

```
h dftactgrp (*no) actgrp('qile')
h debug
d Dumpit      pr
  * No parameters to pass!

c          Callp      Dumpit
c          Eval      *inlr = *on
c          Return

p Dumpit      b
d Dumpit      pi
c          Dump
c          Return
p          E
```

In reality, the routine might fax a spool file, e-mail a user, or perform other tasks. The advantages of using this technique instead of a subroutine is that this method can have its own local variables and then be called from a service program or copied into the source code without fear of having conflicting data names.

Paul Morris is a freelance senior analyst/programmer in the U.K. who provides programming and systems support for the iSeries. You can e-mail him at Paulm@wssitd.demon.co.uk.

THE GOOD, THE BAD, THE UGLY

by Paul Morris

Manipulating Indicators as Arrays

One disagreement I've had with a colleague has been on the subject of manipulating indicators as arrays, as in the following code:

```
C          MOVEA      '00000000'  *IN(60)
```

The style guides warn people from using this technique, but some still use it, even in modern RPG IV programs. I recently had to change a display program to use another indicator to condition a field on the display; every indicator appeared to have been used until I checked through all the MoveA lines and found one indicator in the middle of a group that was set but never used.

Even worse is subscripting for an indicator using a variable such as *IN(Ind). My colleague firmly believed that he had to employ subscripting of indicator arrays to solve this particular issue: He had about 20 different value fields on a screen for a multicurrency system. Each field had to be validated for the number of decimal places and the minimum unit allowable (some currencies don't have decimal places represented by cents or pence, and others have such small value that even one unit is too small).

His solution was to use a subroutine to validate the value. For each screen field in turn, the value was placed in a work field and in an index set representing the first of two possible indicators for the error messages on the display file. These code snippets show what he did:

```
C          Eval      w#ValAmount =
C          Eval      CostFld1
C          Eval      Ind = 64
C          Exsr      ValAmount
:
:
C          ValAmount Begsr
:
```

The validation coding:

```
:
C          If        . . . .
C          Eval      *in(ind) = *on
C          Endif
C          If        . . . .
C          Eval      *in(ind+1) = *on
C          Endif
C          Endsr
```

My alternative was to employ a prototyped call to a procedure. Here are snippets from the solution he eventually used:

The procedure definition in the program D-specs:

```
D ValAmount      pr
D w#valamount    15 02 Value
D w#inde         n
D w#inma         n
```

The procedure call in the C-specs:

```
C          callp
C          ValAmount(CostFld1: *in64: *in65)
```

The procedure at the end of the source:

```
P ValAmount      b
D ValAmount      pi
D w#valamount    15 02 Value
D w#inde         n
D w#inma         n
:
```

The validation coding:

```
:
C          Select
C          When    . . . .
C          Eval    w#inde = *on
C          When    . . . .
C          Eval    w#inma = *on
C          Endsl
C          Return
P          e
```

I think you'll agree that the CallP line is much clearer and shows which indicators are being used.

Paul Morris is a freelance senior analyst/programmer in the U.K. who provides programming and systems support for the iSeries. You can e-mail him at Paul@wssltd.demon.co.uk.

Got a Hot Tip? Looking for an answer?

If you think your tip or question is
of interest to other readers, please
e-mail it to us at

techcorner@iSeriesNetwork.com

Tech Tips

You can name the PR and PI descriptions to match the name of the module, or you can assign a name of your choosing — perhaps a shop-standard name such as Main. If the name on the PR/PI descriptions doesn't match the module name, you must specify the module name with the ExtPgm keyword (e.g., ExtPgm('MyPgm')).

Figure 1 shows a sample *Entry Plist for a program named MyPgm and a pair of sample PR/PI combinations to replace it. You can call this program from any program, whether or not the calling program prototypes the call. The calling program can even be a CL program or an RPG III program.

There are several reasons you might want to switch to using PR/PI descriptions instead of an *Entry Plist. First, the coding is consistent with the coding required for subprocedures, which don't support *Entry Plist. Second, the free-format replacement specifica-

tion for C-specs doesn't support the Plist or Parm operation; free-format coding renders the *Entry Plist obsolete. Last, you can take advantage of prototyping to better control how parameters are passed by using prototyping keywords such as Const and Options(*NoPass).

Note that you cannot pass parameters by Value to a main procedure, nor does a main procedure support a return value.

ROUNDING INTERMEDIATE RESULTS

The Eval operation supports the (H) operation extender to provide a half-adjust, or rounding, capability. But the rounding occurs just once for an expression — after the entire expression is evaluated using RPG's internal sizing rules for intermediate values. Occasionally, you need to exercise more control over the precision and accuracy

of intermediate results than Eval and the (H) extender can provide. To control intermediate results in an expression, you can include the %Dec and %Dech decimal-conversion functions in the expression.

For example, the two statements in Figure 2 might yield different results. The first statement leaves the size and precision of any intermediate values up to RPG and rounds the end result only once. The second statement explicitly controls the size and precision of intermediate values (specifying 13 digits with two decimal places) and rounds each intermediate value. The second statement will yield a more accurate result for this sample calculation.

For information about Bryan Meyers' book *Power Tips for RPG IV*, go to iSeriesNetwork.com/str/books/uniquebook2.cfm?NextBook=218.

The Good, The Bad, The Ugly

by Paul Morris

Reset vs. Clear

An experienced programmer asked me about the difference between Clear and Reset (he had never seen Reset before and was changing one of my programs). So, on the theme of what is basic to some is new to others, I'll briefly explain the difference.

Clear will set, say, a data structure or record structure to its default values, such as blanks in character fields, zeros in numeric fields, or to the value set with an Inz.

Reset will set the data to the value it held when the *InzSR subroutine finished. So, if this subroutine reads a file and populates some fields with data, Reset in other subroutines will cause the data to be reset to these values. Obviously, you cannot use Reset in the *InzSR or any of its called subroutines.

Updating a Joined File

It's often assumed that you can't update a joined file, but as

with so many things, there are ways around it. The following code shows a quick update using interactive SQL. The order header file oep40 needed a flag setting in which there was a match on the corresponding file Xjp20. The Exists statement checks that a record matches in the joined file. The a.ordn40 uses a correlation variable (a) to bring into the Select statement the order number (ordn40) from the file being updated:

```
update oep40 a
set type40 = 'M'
where cono40 = 'DC' and exists
(select * from Xjp20 where cono20 = 'DC' and
pord20 = a.ordn40)
```

Paul Morris is a freelance senior programmer/analyst in the U.K. You can e-mail him at Paul@wssltd.demon.co.uk.

FAQ

Q I have a Windows 98, or Millennium Edition, or Windows XP, or Windows Vista, or Windows 7, or Windows 8, or Windows 10, or Windows 11, or Windows 12, or Windows 13, or Windows 14, or Windows 15, or Windows 16, or Windows 17, or Windows 18, or Windows 19, or Windows 20, or Windows 21, or Windows 22, or Windows 23, or Windows 24, or Windows 25, or Windows 26, or Windows 27, or Windows 28, or Windows 29, or Windows 30, or Windows 31, or Windows 32, or Windows 33, or Windows 34, or Windows 35, or Windows 36, or Windows 37, or Windows 38, or Windows 39, or Windows 40, or Windows 41, or Windows 42, or Windows 43, or Windows 44, or Windows 45, or Windows 46, or Windows 47, or Windows 48, or Windows 49, or Windows 50, or Windows 51, or Windows 52, or Windows 53, or Windows 54, or Windows 55, or Windows 56, or Windows 57, or Windows 58, or Windows 59, or Windows 60, or Windows 61, or Windows 62, or Windows 63, or Windows 64, or Windows 65, or Windows 66, or Windows 67, or Windows 68, or Windows 69, or Windows 70, or Windows 71, or Windows 72, or Windows 73, or Windows 74, or Windows 75, or Windows 76, or Windows 77, or Windows 78, or Windows 79, or Windows 80, or Windows 81, or Windows 82, or Windows 83, or Windows 84, or Windows 85, or Windows 86, or Windows 87, or Windows 88, or Windows 89, or Windows 90, or Windows 91, or Windows 92, or Windows 93, or Windows 94, or Windows 95, or Windows 96, or Windows 97, or Windows 98, or Windows 99, or Windows 100, or Windows 101, or Windows 102, or Windows 103, or Windows 104, or Windows 105, or Windows 106, or Windows 107, or Windows 108, or Windows 109, or Windows 110, or Windows 111, or Windows 112, or Windows 113, or Windows 114, or Windows 115, or Windows 116, or Windows 117, or Windows 118, or Windows 119, or Windows 120, or Windows 121, or Windows 122, or Windows 123, or Windows 124, or Windows 125, or Windows 126, or Windows 127, or Windows 128, or Windows 129, or Windows 130, or Windows 131, or Windows 132, or Windows 133, or Windows 134, or Windows 135, or Windows 136, or Windows 137, or Windows 138, or Windows 139, or Windows 140, or Windows 141, or Windows 142, or Windows 143, or Windows 144, or Windows 145, or Windows 146, or Windows 147, or Windows 148, or Windows 149, or Windows 150, or Windows 151, or Windows 152, or Windows 153, or Windows 154, or Windows 155, or Windows 156, or Windows 157, or Windows 158, or Windows 159, or Windows 160, or Windows 161, or Windows 162, or Windows 163, or Windows 164, or Windows 165, or Windows 166, or Windows 167, or Windows 168, or Windows 169, or Windows 170, or Windows 171, or Windows 172, or Windows 173, or Windows 174, or Windows 175, or Windows 176, or Windows 177, or Windows 178, or Windows 179, or Windows 180, or Windows 181, or Windows 182, or Windows 183, or Windows 184, or Windows 185, or Windows 186, or Windows 187, or Windows 188, or Windows 189, or Windows 190, or Windows 191, or Windows 192, or Windows 193, or Windows 194, or Windows 195, or Windows 196, or Windows 197, or Windows 198, or Windows 199, or Windows 200, or Windows 201, or Windows 202, or Windows 203, or Windows 204, or Windows 205, or Windows 206, or Windows 207, or Windows 208, or Windows 209, or Windows 210, or Windows 211, or Windows 212, or Windows 213, or Windows 214, or Windows 215, or Windows 216, or Windows 217, or Windows 218, or Windows 219, or Windows 220, or Windows 221, or Windows 222, or Windows 223, or Windows 224, or Windows 225, or Windows 226, or Windows 227, or Windows 228, or Windows 229, or Windows 230, or Windows 231, or Windows 232, or Windows 233, or Windows 234, or Windows 235, or Windows 236, or Windows 237, or Windows 238, or Windows 239, or Windows 240, or Windows 241, or Windows 242, or Windows 243, or Windows 244, or Windows 245, or Windows 246, or Windows 247, or Windows 248, or Windows 249, or Windows 250, or Windows 251, or Windows 252, or Windows 253, or Windows 254, or Windows 255, or Windows 256, or Windows 257, or Windows 258, or Windows 259, or Windows 260, or Windows 261, or Windows 262, or Windows 263, or Windows 264, or Windows 265, or Windows 266, or Windows 267, or Windows 268, or Windows 269, or Windows 270, or Windows 271, or Windows 272, or Windows 273, or Windows 274, or Windows 275, or Windows 276, or Windows 277, or Windows 278, or Windows 279, or Windows 280, or Windows 281, or Windows 282, or Windows 283, or Windows 284, or Windows 285, or Windows 286, or Windows 287, or Windows 288, or Windows 289, or Windows 290, or Windows 291, or Windows 292, or Windows 293, or Windows 294, or Windows 295, or Windows 296, or Windows 297, or Windows 298, or Windows 299, or Windows 300, or Windows 301, or Windows 302, or Windows 303, or Windows 304, or Windows 305, or Windows 306, or Windows 307, or Windows 308, or Windows 309, or Windows 310, or Windows 311, or Windows 312, or Windows 313, or Windows 314, or Windows 315, or Windows 316, or Windows 317, or Windows 318, or Windows 319, or Windows 320, or Windows 321, or Windows 322, or Windows 323, or Windows 324, or Windows 325, or Windows 326, or Windows 327, or Windows 328, or Windows 329, or Windows 330, or Windows 331, or Windows 332, or Windows 333, or Windows 334, or Windows 335, or Windows 336, or Windows 337, or Windows 338, or Windows 339, or Windows 340, or Windows 341, or Windows 342, or Windows 343, or Windows 344, or Windows 345, or Windows 346, or Windows 347, or Windows 348, or Windows 349, or Windows 350, or Windows 351, or Windows 352, or Windows 353, or Windows 354, or Windows 355, or Windows 356, or Windows 357, or Windows 358, or Windows 359, or Windows 360, or Windows 361, or Windows 362, or Windows 363, or Windows 364, or Windows 365, or Windows 366, or Windows 367, or Windows 368, or Windows 369, or Windows 370, or Windows 371, or Windows 372, or Windows 373, or Windows 374, or Windows 375, or Windows 376, or Windows 377, or Windows 378, or Windows 379, or Windows 380, or Windows 381, or Windows 382, or Windows 383, or Windows 384, or Windows 385, or Windows 386, or Windows 387, or Windows 388, or Windows 389, or Windows 390, or Windows 391, or Windows 392, or Windows 393, or Windows 394, or Windows 395, or Windows 396, or Windows 397, or Windows 398, or Windows 399, or Windows 400, or Windows 401, or Windows 402, or Windows 403, or Windows 404, or Windows 405, or Windows 406, or Windows 407, or Windows 408, or Windows 409, or Windows 410, or Windows 411, or Windows 412, or Windows 413, or Windows 414, or Windows 415, or Windows 416, or Windows 417, or Windows 418, or Windows 419, or Windows 420, or Windows 421, or Windows 422, or Windows 423, or Windows 424, or Windows 425, or Windows 426, or Windows 427, or Windows 428, or Windows 429, or Windows 430, or Windows 431, or Windows 432, or Windows 433, or Windows 434, or Windows 435, or Windows 436, or Windows 437, or Windows 438, or Windows 439, or Windows 440, or Windows 441, or Windows 442, or Windows 443, or Windows 444, or Windows 445, or Windows 446, or Windows 447, or Windows 448, or Windows 449, or Windows 450, or Windows 451, or Windows 452, or Windows 453, or Windows 454, or Windows 455, or Windows 456, or Windows 457, or Windows 458, or Windows 459, or Windows 460, or Windows 461, or Windows 462, or Windows 463, or Windows 464, or Windows 465, or Windows 466, or Windows 467, or Windows 468, or Windows 469, or Windows 470, or Windows 471, or Windows 472, or Windows 473, or Windows 474, or Windows 475, or Windows 476, or Windows 477, or Windows 478, or Windows 479, or Windows 480, or Windows 481, or Windows 482, or Windows 483, or Windows 484, or Windows 485, or Windows 486, or Windows 487, or Windows 488, or Windows 489, or Windows 490, or Windows 491, or Windows 492, or Windows 493, or Windows 494, or Windows 495, or Windows 496, or Windows 497, or Windows 498, or Windows 499, or Windows 500, or Windows 501, or Windows 502, or Windows 503, or Windows 504, or Windows 505, or Windows 506, or Windows 507, or Windows 508, or Windows 509, or Windows 510, or Windows 511, or Windows 512, or Windows 513, or Windows 514, or Windows 515, or Windows 516, or Windows 517, or Windows 518, or Windows 519, or Windows 520, or Windows 521, or Windows 522, or Windows 523, or Windows 524, or Windows 525, or Windows 526, or Windows 527, or Windows 528, or Windows 529, or Windows 530, or Windows 531, or Windows 532, or Windows 533, or Windows 534, or Windows 535, or Windows 536, or Windows 537, or Windows 538, or Windows 539, or Windows 540, or Windows 541, or Windows 542, or Windows 543, or Windows 544, or Windows 545, or Windows 546, or Windows 547, or Windows 548, or Windows 549, or Windows 550, or Windows 551, or Windows 552, or Windows 553, or Windows 554, or Windows 555, or Windows 556, or Windows 557, or Windows 558, or Windows 559, or Windows 560, or Windows 561, or Windows 562, or Windows 563, or Windows 564, or Windows 565, or Windows 566, or Windows 567, or Windows 568, or Windows 569, or Windows 570, or Windows 571, or Windows 572, or Windows 573, or Windows 574, or Windows 575, or Windows 576, or Windows 577, or Windows 578, or Windows 579, or Windows 580, or Windows 581, or Windows 582, or Windows 583, or Windows 584, or Windows 585, or Windows 586, or Windows 587, or Windows 588, or Windows 589, or Windows 590, or Windows 591, or Windows 592, or Windows 593, or Windows 594, or Windows 595, or Windows 596, or Windows 597, or Windows 598, or Windows 599, or Windows 600, or Windows 601, or Windows 602, or Windows 603, or Windows 604, or Windows 605, or Windows 606, or Windows 607, or Windows 608, or Windows 609, or Windows 610, or Windows 611, or Windows 612, or Windows 613, or Windows 614, or Windows 615, or Windows 616, or Windows 617, or Windows 618, or Windows 619, or Windows 620, or Windows 621, or Windows 622, or Windows 623, or Windows 624, or Windows 625, or Windows 626, or Windows 627, or Windows 628, or Windows 629, or Windows 630, or Windows 631, or Windows 632, or Windows 633, or Windows 634, or Windows 635, or Windows 636, or Windows 637, or Windows 638, or Windows 639, or Windows 640, or Windows 641, or Windows 642, or Windows 643, or Windows 644, or Windows 645, or Windows 646, or Windows 647, or Windows 648, or Windows 649, or Windows 650, or Windows 651, or Windows 652, or Windows 653, or Windows 654, or Windows 655, or Windows 656, or Windows 657, or Windows 658, or Windows 659, or Windows 660, or Windows 661, or Windows 662, or Windows 663, or Windows 664, or Windows 665, or Windows 666, or Windows 667, or Windows 668, or Windows 669, or Windows 670, or Windows 671, or Windows 672, or Windows 673, or Windows 674, or Windows 675, or Windows 676, or Windows 677, or Windows 678, or Windows 679, or Windows 680, or Windows 681, or Windows 682, or Windows 683, or Windows 684, or Windows 685, or Windows 686, or Windows 687, or Windows 688, or Windows 689, or Windows 690, or Windows 691, or Windows 692, or Windows 693, or Windows 694, or Windows 695, or Windows 696, or Windows 697, or Windows 698, or Windows 699, or Windows 700, or Windows 701, or Windows 702, or Windows 703, or Windows 704, or Windows 705, or Windows 706, or Windows 707, or Windows 708, or Windows 709, or Windows 710, or Windows 711, or Windows 712, or Windows 713, or Windows 714, or Windows 715, or Windows 716, or Windows 717, or Windows 718, or Windows 719, or Windows 720, or Windows 721, or Windows 722, or Windows 723, or Windows 724, or Windows 725, or Windows 726, or Windows 727, or Windows 728, or Windows 729, or Windows 730, or Windows 731, or Windows 732, or Windows 733, or Windows 734, or Windows 735, or Windows 736, or Windows 737, or Windows 738, or Windows 739, or Windows 740, or Windows 741, or Windows 742, or Windows 743, or Windows 744, or Windows 745, or Windows 746, or Windows 747, or Windows 748, or Windows 749, or Windows 750, or Windows 751, or Windows 752, or Windows 753, or Windows 754, or Windows 755, or Windows 756, or Windows 757, or Windows 758, or Windows 759, or Windows 760, or Windows 761, or Windows 762, or Windows 763, or Windows 764, or Windows 765, or Windows 766, or Windows 767, or Windows 768, or Windows 769, or Windows 770, or Windows 771, or Windows 772, or Windows 773, or Windows 774, or Windows 775, or Windows 776, or Windows 777, or Windows 778, or Windows 779, or Windows 780, or Windows 781, or Windows 782, or Windows 783, or Windows 784, or Windows 785, or Windows 786, or Windows 787, or Windows 788, or Windows 789, or Windows 790, or Windows 791, or Windows 792, or Windows 793, or Windows 794, or Windows 795, or Windows 796, or Windows 797, or Windows 798, or Windows 799, or Windows 800, or Windows 801, or Windows 802, or Windows 803, or Windows 804, or Windows 805, or Windows 806, or Windows 807, or Windows 808, or Windows 809, or Windows 810, or Windows 811, or Windows 812, or Windows 813, or Windows 814, or Windows 815, or Windows 816, or Windows 817, or Windows 818, or Windows 819, or Windows 820, or Windows 821, or Windows 822, or Windows 823, or Windows 824, or Windows 825, or Windows 826, or Windows 827, or Windows 828, or Windows 829, or Windows 830, or Windows 831, or Windows 832, or Windows 833, or Windows 834, or Windows 835, or Windows 836, or Windows 837, or Windows 838, or Windows 839, or Windows 840, or Windows 841, or Windows 842, or Windows 843, or Windows 844, or Windows 845, or Windows 846, or Windows 847, or Windows 848, or Windows 849, or Windows 850, or Windows 851, or Windows 852, or Windows 853, or Windows 854, or Windows 855, or Windows 856, or Windows 857, or Windows 858, or Windows 859, or Windows 860, or Windows 861, or Windows 862, or Windows 863, or Windows 864, or Windows 865, or Windows 866, or Windows 867, or Windows 868, or Windows 869, or Windows 870, or Windows 871, or Windows 872, or Windows 873, or Windows 874, or Windows 875, or Windows 876, or Windows 877, or Windows 878, or Windows 879, or Windows 880, or Windows 881, or Windows 882, or Windows 883, or Windows 884, or Windows 885, or Windows 886, or Windows 887, or Windows 888, or Windows 889, or Windows 890, or Windows 891, or Windows 892, or Windows 893, or Windows 894, or Windows 895, or Windows 896, or Windows 897, or Windows 898, or Windows 899, or Windows 900, or Windows 901, or Windows 902, or Windows 903, or Windows 904, or Windows 905, or Windows 906, or Windows 907, or Windows 908, or Windows 909, or Windows 910, or Windows 911, or Windows 912, or Windows 913, or Windows 914, or Windows 915, or Windows 916, or Windows 917, or Windows 918, or Windows 919, or Windows 920, or Windows 921, or Windows 922, or Windows 923, or Windows 924, or Windows 925, or Windows 926, or Windows 927, or Windows 928, or Windows 929, or Windows 930, or Windows 931, or Windows 932, or Windows 933, or Windows 934, or Windows 935, or Windows 936, or Windows 937, or Windows 938, or Windows 939, or Windows 940, or Windows 941, or Windows 942, or Windows 943, or Windows 944, or Windows 945, or Windows 946, or Windows 947, or Windows 948, or Windows 949, or Windows 950, or Windows 951, or Windows 952, or Windows 953, or Windows 954, or Windows 955, or Windows 956, or Windows 957, or Windows 958, or Windows 959, or Windows 960, or Windows 961, or Windows 962, or Windows 963, or Windows 964, or Windows 965, or Windows 966, or Windows 967, or Windows 968, or Windows 969, or Windows 970, or Windows 971, or Windows 972, or Windows 973, or Windows 974, or Windows 975, or Windows 976, or Windows 977, or Windows 978, or Windows 979, or Windows 980, or Windows 981, or Windows 982, or Windows 983, or Windows 984, or Windows 985, or Windows 986, or Windows 987, or Windows 988, or Windows 989, or Windows 990, or Windows 991, or Windows 992, or Windows 993, or Windows 994, or Windows 995, or Windows 996, or Windows 997, or Windows 998, or Windows 999, or Windows 1000, or Windows 1001, or Windows 1002, or Windows 1003, or Windows 1004, or Windows 1005, or Windows 1006, or Windows 1007, or Windows 1008, or Windows 1009, or Windows 1010, or Windows 1011, or Windows 1012, or Windows 1013, or Windows 1014, or Windows 1015, or Windows 1016, or Windows 1017, or Windows 1018, or Windows 1019, or Windows 1020, or Windows 1021, or Windows 1022, or Windows 1023, or Windows 1024, or Windows 1025, or Windows 1026, or Windows 1027, or Windows 1028, or Windows 1029, or Windows 1030, or Windows 1031, or Windows 1032, or Windows 1033, or Windows 1034, or Windows 1035, or Windows 1036, or Windows 1037, or Windows 1038, or Windows 1039, or Windows 1040, or Windows 1041, or Windows 1042, or Windows 1043, or Windows 1044, or Windows 1045, or Windows 1046, or Windows 1047, or Windows 1048, or Windows 1049, or Windows 1050, or Windows 1051, or Windows 1052, or Windows 1053, or Windows 1054, or Windows 1055, or Windows 1056, or Windows 1057, or Windows 1058, or Windows 1059, or Windows 1060, or Windows 1061, or Windows 1062, or Windows 1063, or Windows 1064, or Windows 1065, or Windows 1066, or Windows 1067, or Windows 1068, or Windows 1069, or Windows 1070, or Windows 1071, or Windows 1072, or Windows 1073, or Windows 1074, or Windows 1075, or Windows 1076, or Windows 1077, or Windows 1078, or Windows 1079, or Windows 1080, or Windows 1081, or Windows 1082, or Windows 1083, or Windows 1084, or Windows 1085, or Windows 1086, or Windows 1087, or Windows 1088, or Windows 1089, or Windows 1090, or Windows 1091, or Windows 1092, or Windows 1093, or Windows 1094, or Windows 1095, or Windows 1096, or Windows 1097, or Windows 1098, or Windows 1099, or Windows 1100, or Windows 1101, or Windows 1102, or Windows 1103, or Windows 1104, or Windows 1105, or Windows 1106, or Windows 1107, or Windows 1108, or Windows 1109, or Windows 1110, or Windows 1111, or Windows 1112, or Windows 1113, or Windows 1114, or Windows 1115, or Windows 1116, or Windows 1117, or Windows 1118, or Windows 1119, or Windows 1120, or Windows 1121, or Windows 1122, or Windows 1123, or Windows 1124, or Windows 1125, or Windows 1126, or Windows 1127, or Windows 1128, or Windows 1129, or Windows 1130, or Windows 1131, or Windows 1132, or Windows 1133, or Windows 1134, or Windows 1135, or Windows 1136, or Windows 1137, or Windows 1138, or Windows 1139, or Windows 1140, or Windows 1141, or Windows 1142, or Windows 1143, or Windows 1144, or Windows 1145, or Windows 1146, or Windows 1147, or Windows 1148, or Windows 1149, or Windows 1150, or Windows 1151, or Windows 1152, or Windows 1153, or Windows 1154, or Windows 1155, or Windows 1156, or Windows 1157, or Windows 1158, or Windows 1159, or Windows 1160, or Windows 1161, or Windows 1162, or Windows 1163, or Windows 1164, or Windows 1165, or Windows 1166, or Windows 1167, or Windows 1168, or Windows 1169, or Windows 1170, or Windows 1171, or Windows 1172, or Windows 1173, or Windows 1174, or Windows 1175, or Windows 1176, or Windows 1177, or Windows 1178, or Windows 1179, or Windows 1180, or Windows 1181, or Windows 1182, or Windows 1183, or Windows 1184, or Windows 1185, or Windows 1186, or Windows 1187, or Windows 1188, or Windows 1189, or Windows 1190, or Windows 1191, or Windows 1192, or Windows 1193, or Windows 1194, or Windows 1195, or Windows 1196, or Windows 1197, or Windows 1198, or Windows 1199, or Windows 1200, or Windows 1201, or Windows 1202, or Windows 1203, or Windows 1204, or Windows 1205, or Windows 1206, or Windows 1207, or Windows 1208, or Windows 1209, or Windows 1210, or Windows 1211, or Windows 1212, or Windows 1213, or Windows 1214, or Windows 1215, or Windows 1216, or Windows 1217, or Windows 1218, or Windows 1219, or Windows 1220, or Windows 1221, or Windows 1222, or Windows 1223, or Windows 1224, or Windows 1225, or Windows 1226, or Windows 1227, or Windows 1228, or Windows 1229, or Windows 1230, or Windows 1231, or Windows 1232, or Windows 1233, or Windows 1234, or Windows 1235, or Windows 1236, or Windows 1237, or Windows 1238, or Windows 1239, or Windows 1240, or Windows 1241, or Windows 1242, or Windows 1243, or Windows 1244, or Windows 1245, or Windows 1246, or Windows 1247, or Windows 1248, or Windows 1249, or Windows 1250, or Windows 1251, or Windows 1252, or Windows 1253, or Windows 1254, or Windows 1255, or Windows 1256, or Windows 1257, or Windows 1258, or Windows 1259, or Windows 1260, or Windows 1261, or Windows 1262, or Windows 1263, or Windows 1264, or Windows 1265, or Windows 1266, or Windows 1267, or Windows 1268, or Windows 1269, or Windows 1270, or Windows 1271, or Windows 1272, or Windows 1273, or Windows 1274, or Windows 1275, or Windows 1276, or Windows 1277, or Windows 1278, or Windows 1279, or Windows 1280, or Windows 1281, or Windows 1282, or Windows 1283, or Windows 1284, or Windows 1285, or Windows 1286, or Windows 1287, or Windows 1288, or Windows 1289, or Windows 1290, or Windows 1291, or Windows 1292, or Windows 1293, or Windows 1294, or Windows 1295, or Windows 1296, or Windows 1297, or Windows 1298, or Windows 1299, or Windows 1300, or Windows 1301, or Windows 1302, or Windows 1303, or Windows 1304, or Windows 1305, or Windows 1306, or Windows 1307, or Windows 1308, or Windows 1309, or Windows 1310, or Windows 1311, or Windows 1312, or Windows 1313, or Windows 1314, or Windows 1315, or Windows 1316, or Windows 1317, or Windows 1318, or Windows 1319, or Windows 1320, or Windows 1321, or Windows 1322, or Windows 1323, or Windows 1324, or Windows 1325, or Windows 1326, or Windows 1327, or Windows 1328, or Windows 1329, or Windows 1330, or Windows 1331, or Windows 1332, or Windows 1333, or Windows 1334, or Windows 1335, or Windows 1336, or Windows 1337, or Windows 1338, or Windows 1339, or Windows 1340, or Windows 1341, or Windows 1342, or Windows 1343, or Windows 1344, or Windows 1345, or Windows 1346, or Windows 1347, or Windows 1348, or Windows 1349, or Windows 1350, or Windows 1351, or Windows 1352, or Windows 1353, or Windows 1354, or Windows 1355, or Windows 1356, or Windows 1357, or Windows 1358, or Windows 1359, or Windows 1360, or Windows 1361, or Windows 1362, or Windows 1363, or Windows 1364, or Windows 1365, or Windows 1366, or Windows 1367, or Windows 1368, or Windows 1369, or Windows 1370, or Windows 1371, or Windows 1372, or Windows 1373, or Windows 1374, or Windows 1375, or Windows 1376, or Windows 1377, or Windows 1378, or Windows 1379, or Windows 1380, or Windows 1381, or Windows 1382, or Windows 1383, or Windows 1384, or Windows 1385, or Windows 1386, or Windows 1387, or Windows 1388, or Windows 1389, or Windows 1390, or Windows 1391, or Windows 1392, or Windows 1393, or Windows 1394, or Windows 1395, or Windows 1396, or Windows 1397, or Windows 1398, or Windows 1399, or Windows 1400, or Windows 1401, or Windows 1402, or Windows 1403, or Windows 1404, or Windows 1405, or Windows 1406, or Windows 1407, or Windows 1408, or Windows 1409, or Windows 1410, or Windows 1411, or Windows 1412, or Windows 1413, or Windows 1414, or Windows 1415, or Windows 1416, or Windows 1417, or Windows 1418, or Windows 1419, or Windows 1420, or Windows 1421, or Windows 1422, or Windows 1423, or Windows 1424, or Windows 1425, or Windows 1426, or Windows 1427, or Windows 1428, or Windows 1429, or Windows 1430, or Windows 1431, or Windows 1432, or Windows 1433, or Windows 1434, or Windows 1435, or Windows 1436, or Windows 1437, or Windows 1438, or Windows 1439, or Windows 1440, or Windows 1441, or Windows 1442, or Windows 1443, or Windows 1444, or Windows 1445, or Windows 1446, or Windows 1447, or Windows 1448, or Windows 1449, or Windows 1450, or Windows 1451, or Windows 1452, or Windows 1453

TECH TIPS

RclStg Select(*DbXref)

— Paul Conte

For information about Paul Conte's book *Power Tips for iSeries Database and SQL*, go to iSeriesNetwork.com/istr/books/uniquebook2.cfm?NextBook=217.

ISERIES ACCESS ODBC DRIVER KEYWORDS

Q I'm trying to automate an ODBC connection to our iSeries. We want to access J.D. Edwards data that's on the iSeries and output it for custom reporting. However, we don't want the application to be prompted for sign-on information, and we want to make sure it

connects to the right database. What keywords do I need to use in the connection string?

A You can control the user ID and password information, as well as the database information that is used in your ODBC application's connection string. The table in Figure 1 summarizes the most important keywords you can employ with the iSeries Access ODBC driver. The following code snippet shows how to use these keyword pairs in a VB/ADO application:

```
Dim cn As New ADODB.Connection
cn.ConnectionString = _
"DRIVER={Client Access ODBC Driver (32-
bit)};SYSTEM=S1030438;DBQ=ODBCDEMO;CMT=0"
cn.Open
```

— Michael Otey ■

The Good, The Bad, The Ugly

by Paul Morris

SQL and RRN

It's not always appreciated that SQL can process using the relative record numbers (RRNs) of a file. This is not a common technique, but in some cases, it does provide a solution. In the example below, I had to reprocess records imported from a Web application where the data was sitting on a flat file. I needed to get the latest version of a record. This is one case where the application was ugly, forcing me to use these code snippets:

```
d IndNull          s          4b00
d Rrn259a          s          9 00
... ..
c/exec sql
c+ Declare Xj259a Cursor for
c+ Select
c+       max(rrn(Xjp259a))
c+ From
c+       Xjp259a
c+ Where
c+       substr(Dat259a,21,12) = :a_PartNumber
c+ For fetch only
c/end-exec
... ..
c/exec sql
c+ Fetch Next
c+ From
c+       Xj259a
c+ Into
c+       :Rrn259a :indnull
c/end-exec
```

Here, IndNull is a binary field used to indicate that no data has been found — it's used as a null indicator — and Rrn259a is the numeric field that holds the returned RRN. In the SQL Declare statement, I get the highest RRN, where the part number exists in

a certain position within the record. In the Fetch statement, I retrieve both the RRN and the null indicator, which is checked to ensure a result is returned (a negative value means no data was found).

I then used returned RRN to chain to a record to get the data:

```
c       Rrn259a          chain      Xjp259a
```

The following SQL statement shows an alternative way to retrieve the data:

```
c/exec sql
c+ Declare Xj259a Cursor for
c+ Select
c+       Dat259a
c+ From
c+       Xjp259a a
c+ Where
c+       rrn(a) =
c+       (Select max(rrn(b))
c+       From
c+       Xjp259a b
c+       Where
c+       substr(b.Dat259a,31,15) = :a_PartNumber)
c+ For fetch only
```

In this example, I use SQL to retrieve the flat file record by RRN, using an inner Select in the Where clause to determine the largest RRN. From either method, the result is the latest record that satisfies the selection criteria. ■

Paul Morris is a freelance senior programmer/analyst in the U.K. who provides programming and systems support for the iSeries. You can e-mail him at Paul@wssltd.demon.co.uk.

Tech Tips

ODBC PERFORMANCE PROBLEMS WITH OS/400 UPGRADE

Q We're using ODBC to exchange data with our shipping system. The vendor has told us that when we install the next upgrade to our iSeries, ODBC queries will take 15-30 seconds per transaction; these queries take less than a second now. The vendor suggests that we replace the IBM ODBC driver on the shipping system with a third-party driver, as IBM has failed to update its ODBC driver. I'd rather keep the iSeries Access ODBC driver, if possible.

A In specific situations, various third-party ODBC drivers may have an advantage over the IBM iSeries Access ODBC driver, but I strongly doubt that's the case here — especially considering that the vendor's application is currently using the IBM iSeries ODBC driver with very good results. IBM does continue to update the driver as needed to utilize relevant changes that it makes in OS/400.

Considering that your third-party application isn't changing, and that you're currently getting excellent performance using the application with an earlier version of the iSeries Access ODBC driver, you should get the same

level of performance that you're experiencing now using the newer version of the IBM ODBC driver.

In V5R1, IBM changed the way in which the default library and the library list values worked to make them more compatible with typical ODBC usage, and this change did create some problems. If you used the *SQL naming convention, V5R1 and later ODBC drivers used only the value in the SQL default library field for unqualified file names and ceased using the entries in the library list field. But this change doesn't affect performance, and at worst, it requires you to update the DSN that the application is using. In most cases, no changes are required.

— Michael Otey ■

The Good, The Bad, The Ugly

by Paul Morris

Long Data Names

I originally started my career in Cobol programming, where the use of long, descriptive data names was second nature. Coming to RPG III with the limit of six characters was a difficult hurdle for me to overcome. Most programmers whom I've seen using RPG IV will use up to 10 characters, but not many are aware that you can use more than 10 characters. To prove this point to a colleague, I created this somewhat extreme example:

```
d ThisFlagIsToCheckIftheAuditTrailHasBeenRequested
d ForThisProgram...
d
d TheValuesCanBeEither1ForErrorsOnlyOr2forAllSteps...
d                                     s                1

c                                     If
c ThisFlagIsToCheckIftheAuditTrail...
c
c HasBeenRequestedForThisProgram...
c
c TheValuesCanBeEither1For...
```

```
c
ErrorsOnlyOr2forAllSteps...
c                                     = '1'
```

Okay, this is over the top, but it does show that 10 characters is not a limitation. The use of descriptive data names looks better with the /FREE coding style. ■

Paul Morris is a freelance senior programmer/analyst in the U.K. who provides programming and systems support for the iSeries. You can e-mail him at Paul@wssitd.demon.co.uk.

This month's Tech Corner sponsor >>>



Advanced Systems Concepts, Inc.
www.asc-iserics.com