The 5 Newest

Features in RPG

Presented by

Scott Klement

http://www.scottklement.com

How does a computer get drunk?
It takes screen shots.

Support in SEU

There is none.

SEU has not had updates since the release of IBM i 6.1
That was March 21 2008!

Use one of these alternatives:

- Rational Developer for I (RDi): https://tinyurl.com/ibmrdihome
- ILEditor 2: https://ileditor.dev/
- Original ILEditor: https://worksofbarry.com/ileditor/
- MiWorkplace: https://miworkplace.com/
- VS Code with the Code for IBM i Extensions: https://tinyurl.com/codefori

5. DEBUG(*RETVAL)

Released Fall 2020 (PTF for 7.4, 7.3)

- DEBUG(*RETVAL) control option and _QRNU_RETVAL

```
dcl-proc MyProc;

  dcl-s x int(10) inz(99);
  return (x * 43);

end-proc;
```

You can check the return value in the debugger. To do that:

- Set breakpoint on the end-proc
- View variable _QRNU_RETVAL

```
total = readNext(InvNo: MoreData);
dow MoreData;
  total += readNext(InvNo: MoreData);
enddo;

. . .

dcl-proc ReadNext;

  dcl-pi *n packed(9: 2);
    InvoiceNo packed(7: 0) const;
    MoreData ind;
  end-pi;

  reade (InvoiceNo) INVOICEP;
  If %EOF;
    MoreData = *off;
    return 0;
  else;
    ModeData = *on;
    return (QTY * PRICE);
  endif;

end-proc;
```

4. FOR-EACH

Released Fall 2020 (PTF for 7.4, 7.3)

- FOR-EACH loop opcode

Loops through all of the elements in an array. Can be used together with %SUBARR

Old way (FOR loop)

```
total = 0;
for x = 1 to %elem(invoices);
  total += invoices(x).amount;
endfor;
```

New way (FOR-EACH loop)

```
total = 0;
for-each invoice in invoices;
  total += invoice.amount;
endfor;
```

Old way (FOR loop)

```
total = 0;
for x = 1 to numLoaded;
  total += invoices(x).amount;
endfor;
```

New way (FOR-EACH loop)

```
total = 0;
for-each invoice in %subarr(invoices: 1: numLoaded);
  total += invoice.amount;
endfor;
```

3. IN, %RANGE, %LIST

Released Fall 2020 (PTF for 7.4, 7.3)

- IN operator used in comparisons with %RANGE or %LIST
- %RANGE used to compare data in a range
- %LIST returns a temporary array from a list of values

```
// OLD: if Qty >= 1 and Qty <= 999;

if Qty IN %range(1:999);
  msg = 'Quantity acceptable.';
endif;
```

```
// OLD: if shelf = 'TOP' or shelf = 'A' or shelf = 'B'
//           or shelf = 'C' or shelf = 'FLOOR';

if shelf in %list('TOP': 'A': 'B': 'C': 'FLOOR');
  msg = 'Shelf level is acceptable';
endif;
```

FOR-EACH with %LIST

Since %LIST returns a temporary array, it can also be used with the FOR-EACH loop I discussed earlier.

```
// Search shelves in warehouse location for ItemCode

for-each shelf in %list('TOP': 'A': 'B': 'C': 'FLOOR');
  chain (whsloc: shelf) INVENP;
  if %found and ItemCode = Item;
    leave;
  endif;
endfor;
```

2. %UPPER and %LOWER

Released Spring 2021 (PTF for 7.4, 7.3)

- %UPPER built-in function converts a string to uppercase
- %LOWER built-in function converts a string to lowercase
- Optional second parameter specifies the start position within the string
- Works with international characters, too. (So is better than %XLATE!)

```
myString = 'mIxEdcaSe';
myString = %lower(myString);

// myString is now 'mixedcase'

upperCase = %upper(myString);

// upperCase is now 'MIXEDCASE'

titleCase = %lower(upperCase:2);

// titleCase is now 'Mixedcase'
```

1. %SPLIT

Released Spring 2021 (PTF for 7.4, 7.3)

- %SPLIT built-in function splits a string when a given substring is found.
- The result is an array of strings

```
dcl-ds invoice qualified;
  name    varchar(30)  inz('unused');
  amount  packed(9: 2) inz(-1);
end-ds;

dcl-s record varchar(200) inz('Acme Foods Inc | 4502.60');
dcl-s array  varchar(50) dim(2);

array = %split(record:'|');

invoice.name = array(1);
invoice.amount = %dec(array(2):9:2);
```

In this example, two fields are stored in one string, separated by the pipe character. %Split is used to split them into two array elements, which can then be assigned to separate fields in a data structure.

FOR-EACH with %SPLIT

- Since %SPLIT returns an array, it can be used with a FOR-EACH loop
- In this example, "shelves" is a list of shelves separated by commas
- %SPLIT could be used in a similar manner to the way %LIST was used before.

```
dcl-s shelves varchar(20) inz('TOP,A,B,C,FLOOR');
dcl-s shelf char(5);

for-each shelf in %split(shelves:',');

 chain (whsloc: shelf) INVENP;
  if %found and ItemCode = Item;
    leave;
  endif

endfor;
```

Split is very useful when reading IFS files in CSV, Tab or Pipe-Delimited format, or when a database field or user input contains a delimited list of data.

That's Not All!

I only showed my favorite 5 features released in the past year. There are some others as well:

- (Spring 2021) EXPROPTS(*STRICTKEYS) keys in %KDS must match exactly.
- (Fall 2020) EXPROPTS(*ALWBLANKNUM and *USEDECEDIT) control how %INT, %DEC interpret strings to make numbers.
- (Spring 2020) Microseconds for %TIMESTAMP and %TIMESTAMP(*UNIQUE) timestamps are now returned with microseconds (6-digit fraction of a second) and optionally can have an additional 6-digits to be guaranteed unique.
- (Spring 2020) LIKEDS(Qualified.Name) LIKEDS now allows a qualified name
- (Spring 2020) %KDS(ds: number_of_keys) %KDS now allows a variable number of keys

This Presentation

You can download a PDF copy of this presentation:

http://www.scottklement.com/presentations/

Thank you!