Objectives

• Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve.
• Seeing how RPGOA could help deliver a solution that is both elegant and efficient.
• Having a closer look at implementation details.
• Throwing us a few curveballs.
• Running a debugging session with RDi debugger.
• Some extras.
Challenge:
Have a few hundred CL/RPG programs utilizing OPNQRYF converted to use embedded SQL
This IBM white paper lays out these justifications for conversion:

- SQL is the industry standard.
- SQL is the strategic interface for DB2 for i.
- The SQL query engine (SQE) provides superior performance.
- The SQL interfaces offer advanced functions.
- DB2 performance tools are tailored for SQL interface.
- SQL offers simpler application development and maintenance.
- Using SQL reduces the number of programs and lines of code.

Enter RPG Open Access (RPGOA).

Per Jon Paris and Susan Gantner, RPGOA is a way of extending the RPG language using the native-language I/O operations such as READ, CHAIN, WRITE, EXFMT to perform functions way outside their traditional use with database, display and printer files.

http://ibmsystemsmag.com/ibmi/developer/rpg/big-changes-for-rpg-in-ibm-i-7-1/
How does it look from the end programmer’s perspective?

Before:

```sql
dcl-f order101 keyed usropn;
```

After:

```sql
dcl-f order101 keyed handler('OQ2SQLOA') usropn;
```

And that’s it! The Handler keyword parameter (in this case, OQ2SQLOA) is the name of the program that will be called by RPG every time it encounters an I/O operation against the file (including OPEN and CLOSE even when the file is opened/closed implicitly).
Before we begin!

• NOT a case for or against SQL.

• Empowering programmers with a tool for the right circumstance.

• The ultimate goal is to assist lazy programmers like myself.

• NOT a substitute for a good DBA work.

• Still work in progress.
Objectives

• Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve.
We would want to see SQL deliver identical report.

(Physical file PMAST is used by CNX Corp to test their Valence demos).

```
Here it is:

pgm
OVRDBF FILE(PMAST) SHARE(*YES)
OPNQRYF FILE((PMAST)) QRYSLT('ittyp *eq "M"') +
         KEYFLD((PRODONH))
call oqrtst01
CLOF OPNID(PMAST)
endpgm

<table>
<thead>
<tr>
<th>PRDNO</th>
<th>DESCP</th>
<th>ITTYP</th>
<th>UOM</th>
<th>PRODONH</th>
<th>PRODALLOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEXNUT375</td>
<td>Hex Nut 3/8 inch</td>
<td>B</td>
<td>EA</td>
<td>36</td>
<td>11</td>
</tr>
<tr>
<td>PCB6500X</td>
<td>Model 6500X PC Board</td>
<td>M</td>
<td>EA</td>
<td>145</td>
<td>10</td>
</tr>
<tr>
<td>PCB6500X-1</td>
<td>PC 6500X Subcomponent 1</td>
<td>M</td>
<td>EA</td>
<td>173</td>
<td>19</td>
</tr>
<tr>
<td>PCB6500X-2</td>
<td>PC 6500X Subcomponent 3</td>
<td>M</td>
<td>EA</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>PXDUSTIBM</td>
<td>IBM Magic Pixie Dust Solder</td>
<td>B</td>
<td>EA</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>PNTBLU286</td>
<td>Blue Paint RMS 28623</td>
<td>M</td>
<td>LB</td>
<td>66</td>
<td>55</td>
</tr>
<tr>
<td>SCRW375</td>
<td>Standard thread screw 3/8 inch</td>
<td>M</td>
<td>EA</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>S7800GALV</td>
<td>Galvanized Steel Plate .78G</td>
<td>B</td>
<td>SF</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>WASSY6500</td>
<td>Wire Bundle Assembly M#6500</td>
<td>B</td>
<td>EA</td>
<td>33</td>
<td>11</td>
</tr>
<tr>
<td>WCASE04X5</td>
<td>Case Assembly 4&quot;x5&quot;</td>
<td>M</td>
<td>EA</td>
<td>19</td>
<td>11</td>
</tr>
<tr>
<td>WIDGET6500</td>
<td>Acme Widget Model 6500</td>
<td>M</td>
<td>EA</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>XBAT216500</td>
<td>Battery unit for model 6500</td>
<td>B</td>
<td>EA</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```
Objectives

• Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve.

• Seeing how RPGOA could help deliver a solution that is both elegant and efficient.
The handler will prepare and execute a dynamic SQL and deliver the results back to the end program.

Who writes the handler? We do! RPG provides a wealth of info about the “file” and the most recent I/O operation against it.

That info is supplied through pre-defined data structure named QRNOPENACC. The layout of that data structure can be found in source member QRNOPENACC, file QOAR/QRPGLEACC.

As of May 2012, RPGOA is shipped free of charge as part of RPG ILE compiler.

https://www.itjungle.com/2013/08/21/fhg082113-story02/

So, without further ado, let’s start examining the handler code.
• The data structure is passed to the handler by RPG.

```
dcl-proc og2sq1oa;

  dcl-pi *n extPgm;
  info_likeDs(QrnOpenAccess T);
END-PI;
```

• Which op-code brought about the handler invocation?
• This simple program only utilizes READ.
• OPEN and CLOSE will always be returned at least once each.

```
select;
when info.rpgOperation = QrnOperation_OPEN;
  exSr doOpen;
when info.rpgOperation = QrnOperation_READ;
  exSr doRead;
when info.rpgOperation = QrnOperation_CLOSE;
  exSr doClose;
```

• As simple as that.
• OPEN time: Declare and open an SQL cursor.
• Each READ: do an SQL FETCH.
• CLOSE time: bring all to an end (close the cursor).
We want to use the name/value pairs.

```c
// Use the name/value info rather than I/O buffers
info.useNamesValues = *on;
```

RPG will magically return the values we produce at handler to file record format.

How do we get there?

We’ll use SQL descriptor.

For technical details about SQL descriptors including some implementation scenarios, I’ll recommend the excellent series from Paul Tuohy.

https://www.itjungle.com/2016/07/19/fhg071916-story01/

At OPEN time:

```sql
exec sql
   allocate descriptor
      local :useDescriptor
      with max 999;

exec sql
   prepare fakeOpnQryFS1 from :sqlStatement;

exec sql
   describe fakeOpnQryFS1
      using SQL descriptor
      local :useDescriptor;
```
exec sql
  get descriptor :useDescriptor :numColumns = COUNT;
for index = 1 to numColumns;

  // Cast the column values to VARCHAR,
  // so we could use a single host variable
  // to retrieve any column value

  exec sql
    set descriptor :useDescriptor
    value :index
      TYPE = 12,
      LENGTH = 5000;

ENDFOR;

exec sql
  declare fakeOpnQryFC1 cursor for fakeOpnQryFS1;

exec sql
  open fakeOpnQryFC1;
At READ time:

1. Point to names/values area in OA data structure.

```c
// Set up access to names/value information
pNamesValues = info.namesValues;
```

2. Fetch next row from cursor.

```sql
exec sql
  fetch next
    from fakeOpnQryFC1
  into SQL descriptor :useDescriptor;
```

3. If FETCH was unsuccessful, set on EOF to return to end program so it could exit the loop!

```c
if SQLCOD < SQLSTATEOK
  or SQLCOD = SQLSTATENOROW;
  info.eof = *on; // no more records to process
```

4. ELSE, go over each column, point out their values one by one, return to work field.

```c
else;
  info.eof = *off;
  for index = 1 to numColumns;
      // Set up pointer to field buffer area
      pValue = namesValues.field(index).value;
```
Note that DATA will always be cast as VARCHAR (type 12) because we SET the descriptor earlier this way!

5. Finally, load the data into the buffer so that end program will get it.
So – let’s give it a try!

```
call ogrtst01#
```

Voila – the correct set of records is returned in the right sequence.

```
Product: PCB6500X-2  On Hand: 2 Allocated: 1
```

```
Product: WIDGET6500  On Hand: 1000 Allocated: 1
```
• Objectives

• Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve
• Seeing how RPGOA could help deliver a solution that is both elegant and efficient.
• Having a closer look at implementation details.
It’s all well and good but how did the handler know of what OPNQRYF would have specified?

OA provides a spot in QrnOpenAccess data structure for end program passed on info (userArea). To use it would require some extra coding in both CL and RPG application programs thus contradicting the declared goal of assisting lazy programmers. We won’t use it at this time!

Enter new command: OPNSQLF
1. Create a temporary user space, name it after the file.
2. Take the “OPNQRYF” content (QRYSLT, KEYFLD) to that user space using a data structure template that is shared with handler.
3. At handler time, retrieve the “OPNQRYF” content via the shared data structure.
4. Compose a dynamic SQL statement for the cursor using the retrieved “OPNQRYF” content.

Open SQL File (OPNSQLF)

Type choices, press Enter.

File specifications:
- `pmast___` Name
  - `*LIBL___` Name, *LIBL, *CURLIB
  - `*FIRST___` Name, *FIRST, *LAST, *ALL
  - `*ONLY___` Name, *ONLY

Query selection expression . . . `ittyp *eq "M"`

Key field specifications:
- `ordonh___` Name, *NONE, *FILE
  - `*MAPFLD, 1, 2, 3, 4` Name, *ASCEND, *DESCEND
  - `*ABSVAL` Name, *ASCEND, *DESCEND
1. Note that we don’t now need OVRDBF in conjunction with the new command OPNSQLF.
2. CLOSQLF will simply delete the temporary user space object.

The user space content can be browsed as an IFS object.
The handler will parse that data prior to declaring the cursor.

```
when info.rpgOperation = QrnpOperation_OPEN;

    // Retrieve the name of user space containing the OPNQRYF info. 
    // The convention is that the user space will be named after 
    // the file with handler. 
    usrSpcQualifiedNam = info.externalFile.name+'QTEMP';

    // Point to user space
    getUserSpacePointer(usrSpcQualifiedNam
          :usrSpcAddress);

    // Get SQL statement by converting OPNQRYF to SQL
    sqlStatement = getSqlFromOpnQryF(opnQryFData);

dcl-proc getSqlFromOpnQryF;

    dcl-pi *n varchar(5000);
         peOpnQryFData likeDs(opnQryFData_T) value;
    END-PI;

    dcl-s sqlStatement varchar(5000) inz('select * from ');
    dcl-s whereClause varchar(2048);
    dcl-s index int(5);

    // Add primary table
    if peOpnQryFData.fileDsArray(1).qualifiedFileName.library <> '*LIBL' 
          and peOpnQryFData.fileDsArray(1).qualifiedFileName.library <> '*blanks;
          sqlStatement += %trim(peOpnQryFData.fileDsArray(1).qualifiedFileName.library)+'.';
    ENDIF;
    sqlStatement += %trim(peOpnQryFData.fileDsArray(1).qualifiedFileName.file);

    // Add Where clause
    if peOpnQryFData.qrySlt <> '*ALL' 
          and peOpnQryFData.qrySlt <> '*blanks;
          whereClause = getWhereClause(peOpnQryFData.qrySlt);
    sqlStatement += ' where ' +whereClause;
    ENDIF;
```
Objectives

• Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve.
• Seeing how RPGOA could help deliver a solution that is both elegant and efficient.
• Having a closer look at implementation details.
• Throwing us a few curveballs.
Next Challenge:
What if RPG uses a logical file?
And what if CL uses an OPNID for OPNQRYF that is neither physical nor logical file name?

In “traditional” CL programs, they would override the logical to physical file name, then OPNQRYF on physical.

dcl-ds pmastDs likerec(pmastr);
read pmastr101 pmastDs;
doW not %eof;
   displayRecord = 'Product: '+pmastDs.prdno+ ' Description: '+pmastDs.dscp;
quillngtx(displayRecord:%len(displayRecord):
   messageid:messagefile:
      errorcode);
read pmastr101 pmastDs;
ENDDO;
How are we going to mimic that technique while still being on a mission of assisting lazy programmers (that is, introduce as little new code to the front end program as possible)?

It appears that we do need a file override command after all?

Well, sort of...

Enter new command: OVRSQLF

RPG program still uses the same handler: OQ2SQLOA.

dcl-f pmast101 rename(pmast:pmastr) handler('OQ2SQLOA');
All that the new command does is manipulate the name of the user space object. In fact, it creates two duplicate objects (in conjunction with OPNSQLF).

One object to face the handler (that is named after the file the handler is attached to), and another to provide an OPNID link to CLOSQLF file command (the latter is optional and can be utilized if a cleanup is needed: the removal of the user space objects).

<table>
<thead>
<tr>
<th>Opt</th>
<th>Object link</th>
<th>Type</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MYOPNID.USRSPC</td>
<td>USRSPC</td>
<td></td>
<td>User Space for OPNQRYF Han</td>
</tr>
<tr>
<td>1</td>
<td>PMASTL01.USRSPC</td>
<td>USRSPC</td>
<td></td>
<td>User Space for OPNQRYF Han</td>
</tr>
</tbody>
</table>
Note that OVRSQLF and OPNSQLF could be executed in any order and would still produce identical results when run one after another (as it would be in “old life” with OBRDBF and OPNQRYF).

Selection or command

```bash
==> OVRSQLF FILE(PIASTL01) TOFILE(PIAST)
```

Work with Object Links

Directory .......: /qsys.lib/qtemp.lib

Type options, press Enter.

2=Edit 3=Copy 4=Remove 5=Display 7=Rename 8=Display attributes
11=Change current directory ...

<table>
<thead>
<tr>
<th>Opt</th>
<th>Object link</th>
<th>Type</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PMAST.USRSPC</td>
<td>USRSPC</td>
<td></td>
<td>User Space for OPNQRYF Han</td>
</tr>
<tr>
<td>__</td>
<td>PMASTL01.USRSPC</td>
<td>USRSPC</td>
<td></td>
<td>User Space for OPNQRYF Han</td>
</tr>
</tbody>
</table>

Selection or command

```bash
==> OPNSQLF FILE((PIAST)) QRYSLT('ittyp *eq "M"') KEYFLD((DESCP)) OPNID(MYOPNID)
```

<table>
<thead>
<tr>
<th>Opt</th>
<th>Object link</th>
<th>Type</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MYOPNID.USRSPC</td>
<td>USRSPC</td>
<td></td>
<td>User Space for OPNQRYF Han</td>
</tr>
<tr>
<td>__</td>
<td>PMASTL01.USRSPC</td>
<td>USRSPC</td>
<td></td>
<td>User Space for OPNQRYF Han</td>
</tr>
</tbody>
</table>
### Work with Object Links

Directory . . . .: /qsys.lib/qtemp.lib

Type options, press Enter.
- 2=Edit  3=Copy  4=Remove  5=Display  7=Rename  8=Display attributes
- 11=Change current directory ...

<table>
<thead>
<tr>
<th>Opt</th>
<th>Object link</th>
<th>Type</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MYOPNID.USRSPC</td>
<td>USRSPC</td>
<td>USRSPC</td>
<td>User Space for OPNQRYF Han</td>
</tr>
<tr>
<td></td>
<td>PMAST.USRSPC</td>
<td>USRSPC</td>
<td>USRSPC</td>
<td>User Space for OPNQRYF Han</td>
</tr>
</tbody>
</table>

### Selection or command

```plaintext
===> OVRSQLF FILE(PMASTL01) TOFILE(PMAST)
```

<table>
<thead>
<tr>
<th>Opt</th>
<th>Object link</th>
<th>Type</th>
<th>Attribute</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MYOPNID.USRSPC</td>
<td>USRSPC</td>
<td>USRSPC</td>
<td>User Space for OPNQRYF Han</td>
</tr>
<tr>
<td></td>
<td>PMASTL01.USRSPC</td>
<td>USRSPC</td>
<td>USRSPC</td>
<td>User Space for OPNQRYF Han</td>
</tr>
</tbody>
</table>
Next Challenge:
Accommodate file random access in RPG program.
Namely, SETLL, CHAIN, READE.
Another physical file from CNX. We give this physical the name ORDER.

Logical ORDERL01 that is built over it.
### SQL Query 1

```sql
select 
cusno, orddate, orderno, ordscts, orddaddr, ordcity, ordqty, orditem, ordamt
from alexk.order;
```

<table>
<thead>
<tr>
<th>CUSNO</th>
<th>ORDATE</th>
<th>ORDERNO</th>
<th>ORDSCTS</th>
<th>ORDADDR</th>
<th>ORDCITY</th>
<th>ORDQTY</th>
<th>ORDITEM</th>
<th>ORDAMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1033</td>
<td>2016-07-15</td>
<td>5001 A</td>
<td>372-9476 Aareaen St.</td>
<td>Valdez</td>
<td>100.00</td>
<td>8XBATZ26500</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>1118</td>
<td>2016-05-21</td>
<td>5002 C</td>
<td>562-6135 Maleuada St.</td>
<td>Everett</td>
<td>24.00</td>
<td>8XBATZ26500</td>
<td>82.00</td>
<td></td>
</tr>
<tr>
<td>1095</td>
<td>2016-04-04</td>
<td>5003 C</td>
<td>P.O. Box 215, 1966 Sodailes St.</td>
<td>Mission Vejo</td>
<td>4.00</td>
<td>HEVENT375</td>
<td>178.00</td>
<td></td>
</tr>
<tr>
<td>1013</td>
<td>2016-07-27</td>
<td>5004 A</td>
<td>790-468 1951 Ave.</td>
<td>Waukegan</td>
<td>2.000</td>
<td>PC66500X</td>
<td>60.00</td>
<td></td>
</tr>
<tr>
<td>1037</td>
<td>2016-07-10</td>
<td>5005 C</td>
<td>729-5164 Nic St.</td>
<td>Duquesne</td>
<td>25.000</td>
<td>578003GALV</td>
<td>94.00</td>
<td></td>
</tr>
<tr>
<td>1096</td>
<td>2016-06-18</td>
<td>5006 X</td>
<td>1445 Mollie Road.</td>
<td>Oswego</td>
<td>1.000</td>
<td>578003GALV</td>
<td>127.00</td>
<td></td>
</tr>
<tr>
<td>207563</td>
<td>2015-05-31</td>
<td>5007 A</td>
<td>3470 Quisque St.</td>
<td>Suffolk</td>
<td>6.000</td>
<td>PNTLBU286</td>
<td>138.00</td>
<td></td>
</tr>
<tr>
<td>1088</td>
<td>2016-01-20</td>
<td>5008 C</td>
<td>8590 Congue Rd.</td>
<td>Covington</td>
<td>24.000</td>
<td>WASSY6500</td>
<td>92.00</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>2016-04-04</td>
<td>5009 X</td>
<td>2472 Turpie Rd.</td>
<td>Waycross</td>
<td>25.000</td>
<td>PC66500X</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>432009</td>
<td>2015-05-27</td>
<td>5010 C</td>
<td>992-8517 Preum Road</td>
<td>Berlin</td>
<td>27.000</td>
<td>WASSY6500</td>
<td>192.00</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>2016-05-09</td>
<td>5011 X</td>
<td>2541 Quix Av.</td>
<td>Frederic Kurg</td>
<td>6.000</td>
<td>PX6500X</td>
<td>103.00</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>2016-01-20</td>
<td>5012 C</td>
<td>629-4908 Metus St.</td>
<td>Morrison</td>
<td>27.000</td>
<td>SCR00052</td>
<td>172.00</td>
<td></td>
</tr>
<tr>
<td>1094</td>
<td>2016-08-28</td>
<td>5013 X</td>
<td>472-610 Ante St.</td>
<td>Corona</td>
<td>7.000</td>
<td>PC66500X</td>
<td>171.00</td>
<td></td>
</tr>
<tr>
<td>1023</td>
<td>2016-04-04</td>
<td>5014 A</td>
<td>872 Laboris Ave</td>
<td>Brooklyn Park</td>
<td>6.000</td>
<td>PC66500X</td>
<td>125.00</td>
<td></td>
</tr>
<tr>
<td>1109</td>
<td>2016-09-20</td>
<td>5015 X</td>
<td>2078 Osis Street</td>
<td>Shamonin</td>
<td>20.000</td>
<td>PC66500X</td>
<td>62.00</td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>2016-10-20</td>
<td>5016 C</td>
<td>2553 Odis Av.</td>
<td>Jackson</td>
<td>22.000</td>
<td>PC66500X</td>
<td>46.00</td>
<td></td>
</tr>
<tr>
<td>732</td>
<td>2016-08-02</td>
<td>5017 C</td>
<td>666-943 Ave.</td>
<td>York Falls</td>
<td>4.000</td>
<td>PC66500X</td>
<td>114.00</td>
<td></td>
</tr>
</tbody>
</table>

### SQL Query 2

```sql
select 
cusno, orddate, orderno, ordscts, orddaddr, ordcity, ordqty, orditem, ordamt
from alexk.order
where ordscts = 'C' and ordamt > 50
order by cusno, orddate, orderno;
```

<table>
<thead>
<tr>
<th>CUSNO</th>
<th>ORDATE</th>
<th>ORDERNO</th>
<th>ORDSCTS</th>
<th>ORDADDR</th>
<th>ORDCITY</th>
<th>ORDQTY</th>
<th>ORDITEM</th>
<th>ORDAMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>2016-03-31</td>
<td>5825 C</td>
<td>655-8493 Ullamcorper, Rd.</td>
<td>Scarborough</td>
<td>17.00</td>
<td>WCASEG9K5</td>
<td>80.00</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>2016-05-27</td>
<td>5289 C</td>
<td>P.O. Box 763, 4193 Pron Ave</td>
<td>Siddles</td>
<td>6.000</td>
<td>57900GALV</td>
<td>61.00</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>2016-06-06</td>
<td>5984 C</td>
<td>1546 Non Street</td>
<td>Trenton</td>
<td>10.000</td>
<td>57900GALV</td>
<td>136.00</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>2016-06-12</td>
<td>5703 C</td>
<td>P.O. Box 753, 626 Ac Avenue</td>
<td>Indio</td>
<td>1.000</td>
<td>57900GALV</td>
<td>154.00</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>2016-05-06</td>
<td>5152 C</td>
<td>8264 Nulam St.</td>
<td>Kankakee</td>
<td>10.000</td>
<td>PCB65000X</td>
<td>141.00</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>2016-06-03</td>
<td>5428 C</td>
<td>Ap #188-9039 Vet., Road</td>
<td>Yalome</td>
<td>15.000</td>
<td>PC66500X</td>
<td>52.00</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-01-27</td>
<td>5069 C</td>
<td>1650 Cursus Rd.</td>
<td>Hoover</td>
<td>325.000</td>
<td>PC66500X</td>
<td>83.34</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-02-18</td>
<td>6114 C</td>
<td>P.O. Box 778, 4368 Nic St.</td>
<td>Brunswick</td>
<td>817.000</td>
<td>PX8500X</td>
<td>65.71</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-03-16</td>
<td>6051 C</td>
<td>655-8493 Ullamcorper, Rd.</td>
<td>Cedarr Rapids</td>
<td>680.000</td>
<td>HHR00373</td>
<td>61.67</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-03-18</td>
<td>6091 C</td>
<td>3768 Ac. Avenue</td>
<td>Haverhill</td>
<td>941.000</td>
<td>57800GALV</td>
<td>81.26</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-03-19</td>
<td>6073 C</td>
<td>P.O. Box 478, 4597 Curabitur Rd.</td>
<td>Pittsfield</td>
<td>11.000</td>
<td>WASSY6500</td>
<td>58.49</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-04-24</td>
<td>6109 C</td>
<td>2472 Turpia St.</td>
<td>Cape Girardeau</td>
<td>792.000</td>
<td>PNTLBU286</td>
<td>90.61</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-05-01</td>
<td>6096 C</td>
<td>P.O. Box 396, 7122 Ante Rd.</td>
<td>Detroit</td>
<td>579.000</td>
<td>PCB6500X</td>
<td>82.82</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-07-27</td>
<td>6087 C</td>
<td>702-7580 Ipsum Rd.</td>
<td>Grass Valley</td>
<td>597.000</td>
<td>PX8500X</td>
<td>85.41</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-10-25</td>
<td>6078 C</td>
<td>5494 Nic Ave</td>
<td>Newark</td>
<td>684.000</td>
<td>PX8500X</td>
<td>89.41</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-11-13</td>
<td>6118 C</td>
<td>500-6679 Suspender Ave.</td>
<td>Waukegan</td>
<td>117.000</td>
<td>WASSY6500</td>
<td>64.26</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>2016-12-04</td>
<td>6060 C</td>
<td>2886 Dictum Av.</td>
<td>Minneapolis</td>
<td>752.000</td>
<td>SCR00037</td>
<td>64.80</td>
<td></td>
</tr>
<tr>
<td>1003</td>
<td>2016-07-29</td>
<td>5882 C</td>
<td>2350 Sapien St.</td>
<td>Pittsfield</td>
<td>9.000</td>
<td>57800GALV</td>
<td>123.00</td>
<td></td>
</tr>
<tr>
<td>1004</td>
<td>2017-07-25</td>
<td>6078 C</td>
<td>5494 Nic Ave</td>
<td>Newark</td>
<td>7.000</td>
<td>SCR00037</td>
<td>76.56</td>
<td></td>
</tr>
</tbody>
</table>
Traditional programs.

call oqrtst058
CLOF OPNID(ORDER)
endpgm

open orderl01;

dcl-s peCusNo zoned(6:0) inz(300);
dcl-s peOrdDate date inz(d'2017-10-25');
dcl-s peOrderNo zoned(6:0) inz(6078);

orderKeys.cusNo = peCusNo;
orderKeys.ordDate = peOrdDate;
orderKeys.ordoNo = peOrderNo;

// Test file positioning followed by reads by key
setLL %kds(orderKeys:3) orderl01;
readE %kds(orderKeys:1) orderl01 orderDs;
doW not %eof;

displayRecord = 'Customer: '+'%char(orderDs.cusNo)+' Date: '+'%char(orderDs.o) +' Order: '+'%char(orderDs.ordoNo);
quilngtx(displayRecord: %len(displayRecord):
mESSAGEID:messagefile:
errorcode);

ENDDO;

*inlr=*on;
return;
Open Access programs.

The handler program needs to *INHERIT *TERASPACE storage model

... and the rest is identical
The two sets of programs produce identical results.
How do we do it in Handler?

1. Defer the building of SQL statement until the first I/O operation is encountered.

2. Incorporate the KLIST “selection” into the combined QRYSLT clause

```plaintext
// Here is the pseudocode for the logic.
// Suppose we have
//   SETL (key1,key2,key3) FILE
// followed by one of the following:
//   READE (key1,key2,key3) FILE
// then field1=key1 and field2=key2 and field3=key3
//   READE (key1,key2) FILE
// then <as above>
// or field1=key1 and field2=key2 and field3=key3
//   READE (key1) FILE
// then <as above>
// or field1=key and field2=key2
//   READ FILE
// then <as above>
// or field1=key1
```
Objectives.

- Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve.
- Seeing how RPGOA could help deliver a solution that is both elegant and efficient.
- Having a closer look at implementation details.
- Throwing us a few curveballs.
- Running a debugging session with RDi debugger.
EXTRAWHERECLAUSE = (CUSNO=300 and ORDDATE='2017-10-25' and ORDERNO=5078 or CUSNO=300 and ORDDATE='2017-10-25' and ORDERNO>6078 or CUSNO=300 and ORDDATE> '2017-10-25')

SQLSTATEMENT = select * from ORDER where ordsts = 'C' AND ordamt > 50 ;

and (CUSNO=300 and ORDDATE='2017-10-25' and ORDERNO=5078)

or CUSNO=300 and ORDDATE='2017-10-25' and ORDERNO>6078

or CUSNO=300 and ORDDATE> '2017-10-25')
What’s next?

• Read backwards (READP, READPE)
• Output opcodes (WRITE, UPDATE)
• Sort in descending order
• Sort by absolute value
• BIFs (%VALUES, %RANGE, %WLDCRD)
• Field renames (internal, external)
• Joins
• GRPFLD
• MAPFLD
• Enhanced error handling....
Objectives

• Getting to see a technical challenge that, at the first glance, requires a traditional “piecemeal” approach to solve.
• Seeing how RPGOA could help deliver a solution that is both elegant and efficient.
• Having a closer look at implementation details.
• Throwing us a few curveballs.
• Running a debugging session with RDi debugger.
• Some extras.
Bonus Topic: Table Function to Query a CSV File and Flexible DB Upload Utility

(Yes, I am a hockey fan!)

[Table showing NHL 2016-17 Player Data]

www.hockeyabstract.com/testimonials

NHL 2016-17 Player Data
posted May 13, 2017, 8:25 PM by Robert Vollman
Function written as an extension to Scott Klement’s service program CSVR4.

Any rule that is applicable to SQL Select stays effective for this table function (including joins!)

```
select trim(m) || ', ' || trim(n) player
from table(alexk.csv2sql(char('/alexk/uploads/nhl 2016-17.csv',''))) as t
where d = 'USA';
```
create table alexk.nhlplayers2017 (
lastname char(25),
firstname char(25),
team char(10),
country char(3),
city char(25),
gamesplayed integer,
goals integer,
assists integer,
plusminus integer
);

Dynamic Table Load from CSV (LOADCSV)

Type choices, press Enter.

IFS Path to CSV File: ./alexk/uploads/nhl 2016-17.csv

DB2 Table Name: nhlplayers2017

Schema: Name, *LIBL
Load Permanent Table (Y/N): *YES
Clear Table B4 Load? (Y/N): *yes
```sql
select *
from alexk.nhlplayers2017;
```

<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>FIRSTNAME</th>
<th>TEAM</th>
<th>COUNTRY</th>
<th>CITY</th>
<th>GAMESPLAYED</th>
<th>GOALS</th>
<th>ASSISTS</th>
<th>PLUSMINUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott</td>
<td>Spencer</td>
<td>CHI</td>
<td>CAN</td>
<td>Hamilton</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Abdelkader</td>
<td>Justin</td>
<td>DET</td>
<td>USA</td>
<td>Muskegon</td>
<td>64</td>
<td>7</td>
<td>14</td>
<td>-20</td>
</tr>
<tr>
<td>Aberg</td>
<td>Pontus</td>
<td>NSH</td>
<td>SWE</td>
<td>Stockholm</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>-2</td>
</tr>
<tr>
<td>Acciai</td>
<td>Noel</td>
<td>BOS</td>
<td>USA</td>
<td>Johnston</td>
<td>29</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Agostino</td>
<td>Kenny</td>
<td>STL</td>
<td>USA</td>
<td>Morristown</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Aho</td>
<td>Sebastian</td>
<td>CAR</td>
<td>FIN</td>
<td>Rauma</td>
<td>82</td>
<td>24</td>
<td>25</td>
<td>-1</td>
</tr>
<tr>
<td>Alzner</td>
<td>Karl</td>
<td>WSH</td>
<td>CAN</td>
<td>Burnaby</td>
<td>82</td>
<td>3</td>
<td>10</td>
<td>23</td>
</tr>
<tr>
<td>Anderson</td>
<td>Josh</td>
<td>CBJ</td>
<td>CAN</td>
<td>Burlington</td>
<td>78</td>
<td>17</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Andersson</td>
<td>Rasmus</td>
<td>CGY</td>
<td>SWE</td>
<td>Malmö</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>Andreoff</td>
<td>Andy</td>
<td>L.A</td>
<td>CAN</td>
<td>Pickering</td>
<td>36</td>
<td>0</td>
<td>2</td>
<td>-2</td>
</tr>
</tbody>
</table>
Thank you!