

Unleash the Power of RDi and VS Code for IBM i



Charles Guarino
Central Park Data Systems, Inc.
@charlieguarino

0

Charles Guarino believes in the “power” of IBM Power Systems. His career reflects his dedication and interest in bringing the platform and its solutions to others and as a result has been recognized as an IBM Champion.

He is a member of COMMON’s Speaker Excellence Hall of Fame and a proud recipient of the Al Barsa Memorial Scholarship Award. Other professional endeavors have included the roles of President and monthly Q&A host for many years of the Long Island System User’s Group LISUG (www.lisug.org)

In the spring of 2020 Charles created the monthly virtual IBMi iChime community. Each meeting features a “No PowerPoint” discussion with an industry expert. To learn more visit www.ichime.io. You can also listen to podcasts of earlier meetings on your favorite podcast platform.

Today, along with the team at Central Park Data Systems, he is serving individuals and companies on a worldwide basis through his consulting work and award-winning speaking engagements. Charles is a true people person and can often be found at conferences sharing his expertise on RDi and other IBM i topics.

Charles can be reached at cguarino@centralparkdata.com.
LinkedIn - <http://www.linkedin.com/in/guarinocharles>
Twitter - @charlieguarino



1

What We'll Cover ...

Introduction

VS Code for IBM i

Source code editing

DB2 extension

Debugging

Rational Developer for IBM i

Workbench

Source code editing

Debugging

Wrap Up

2

Modernization – Digital Transformation

The Developer

Infrastructure

Application
Modernization

Digital
Transformation

3

Modernization – Digital Transformation

The Developer

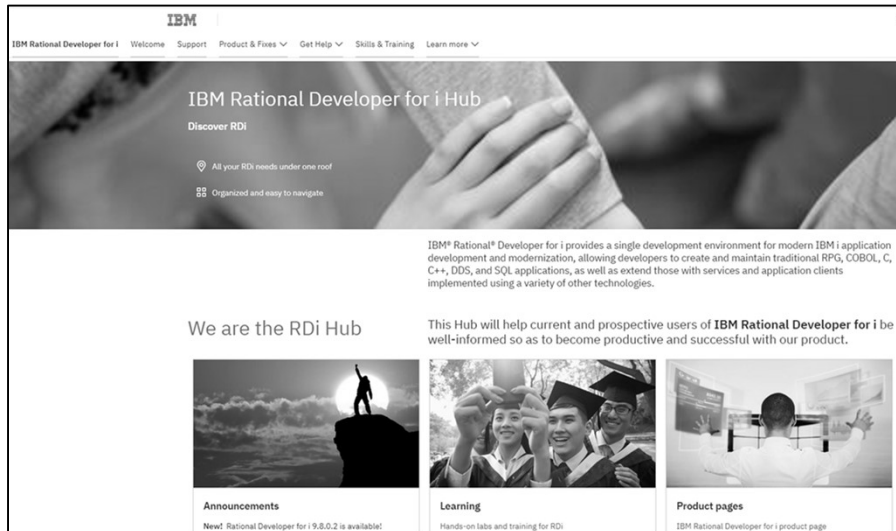
Infrastructure

Application
Modernization

Digital
Transformation

4

RD*i* Developer Hub



IBM Rational Developer for i | Welcome | Support | Product & Fixes | Get Help | Skills & Training | Learn more

IBM Rational Developer for i Hub

Discover RDi

- All your RDi needs under one roof
- Organized and easy to navigate

IBM® Rational® Developer for i provides a single development environment for modern IBM i application development and modernization, allowing developers to create and maintain traditional RPG, COBOL, C, C++, DDS, and SQL applications, as well as extend those with services and application clients implemented using a variety of other technologies.

We are the RDi Hub

This Hub will help current and prospective users of **IBM Rational Developer for i** be well-informed so as to become productive and successful with our product.

- Announcements**
New! Rational Developer for i 9.8.0.2 is available!
- Learning**
Hands-on labs and training for RDi
- Product pages**
IBM Rational Developer for i product page

<https://www.ibm.com/support/pages/ibm-rational-developer-i-hub>

5

Code for IBM i extension

Visual Studio | Marketplace

Visual Studio Code > Other > Code for IBM i

Code for IBM i
Code4i | 37,672 installs | ★★★★★ (19) | Free

Maintain your RPGLE, CL, COBOL, C/CPP on IBM i right from Visual Studio Code.

[Install](#) [Trouble Installing?](#)

Overview Version History Q & A Rating & Review

Code for IBM i

[GitHub star this repo](#)

Maintain your RPGLE, CL, COBOL, C/CPP on IBM i right from Visual Studio Code. Edit and compile all ILE languages, view errors inline, content assist for RPGLE and CL, source date support, and much more. Code for IBM i has thousands of daily users and a rapidly increasing number of downloads. We embrace open-source so that we can best support our community.

- [Install from Marketplace](#)
- [Install the extension pack](#) for RPGLE, CL and COBOL support
- [Watch some tutorials](#)
- [View our documentation](#)
- [See previous releases](#)
- [Build from source](#) (see below!)
- [Use our IBM i API](#) in your own extension

Categories
Other

Tags
as400 cobol keybindings rpgle

Works with
Universal

Resources
[Issues](#)
[Repository](#)
[Homepage](#)
[License](#)
[Download Extension](#)

<https://marketplace.visualstudio.com/items?itemName=HalcyonTechLtd.code-for-ibmi>

6

What We'll Cover ...

Introduction

VS Code for IBM i

Source code editing

DB2 extension

Debugging

Rational Developer for IBM i

Workbench

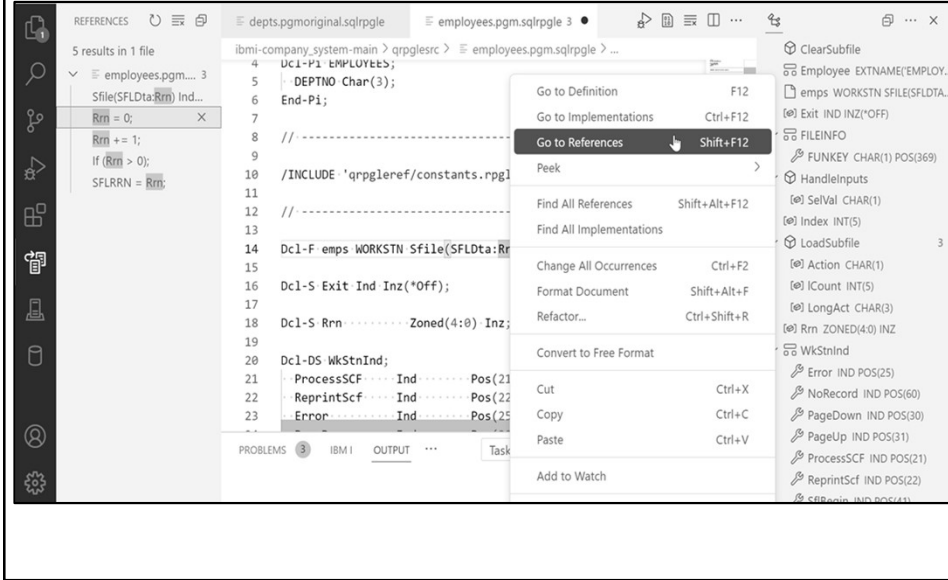
Source code editing

Debugging

Wrap Up

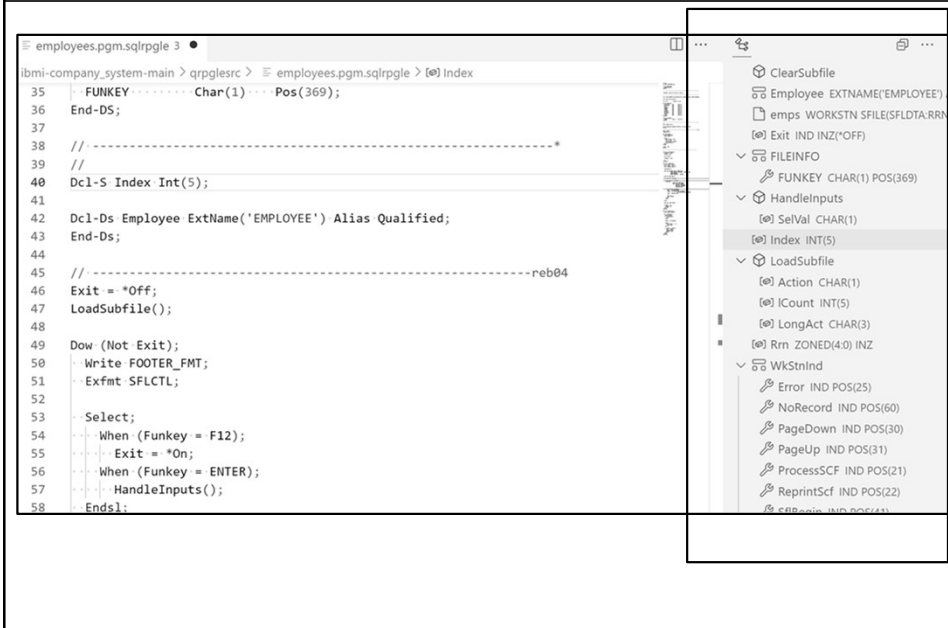
7

VS Code Basic editing features



8

VS Code Outline



9

VS Code Content assist

```
57     ······HandleInputs();
58     ······Endsl;
59     Enddo;
60
61     x = %sub
62
63     *INLR =
64     Return;
65
66     //-----
67
68     Dcl-Proc ClearSubfile;
69     ······Sf1DspCtl = *Off;
70     ······Sf1Dsp = *Off;
71
```

The image shows a content assist popup in VS Code. The popup is a light gray box with a dark header bar containing the text "%subarr". Below the header, there are four suggestions, each with a small icon to its left: a square with a dot for "%subdt", a square with a dot for "%subst", a cube for "LoadSubfile", and a cube for "ClearSubfile". The suggestions are aligned to the right of the popup.

10

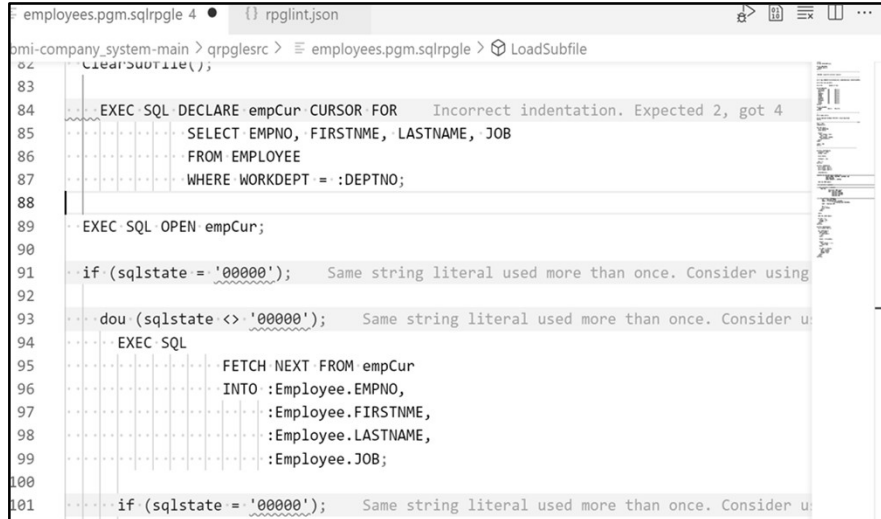
VS Code Hovering

```
employees.pgm.sqlrpgle 3
ibmi-company_system-main > qrpglesrc > employees.pgm.sqlrpgle > ClearSubfile
47     LoadSubfile();
48
49     Dow (Not Exit);
50     ······Write FOOTER_FMT;
51     ······Exfmt SFLCTL;
52
53     ······Select;
54     ······When (Funkey =
55     ······Exit = *On;
56     ······When (Funkey = ENTER);
57     ······HandleInputs();
58     ······Endsl;
59     Enddo;
60
61     x = %sub
62
63     *INLR = *ON;
64     Return;
65
66     //-----
```

The image shows a hovering tooltip in VS Code. The tooltip is a light gray box with a dark header bar containing the text "ENTER: X'F1'". Below the header, there is a single suggestion: "@file file:///c%3A/Users/cpdat/Downloads/ibmi-company_system-main/ibmi-company_system-main/qrpgle/refs/constants.rpgleinc:24". The tooltip is positioned over the "ENTER" text in the code.

11

VS Code RPGLE source code linter



```
employees.pgm.sqlrpgle 4 • {} rpglint.json
bmi-company_system-main > qrpplsrc > employees.pgm.sqlrpgle > LoadSubfile
04 ClearSubroutine();
83
84 ..EXEC SQL DECLARE empCur CURSOR FOR Incorrect indentation. Expected 2, got 4
85 ..SELECT EMPNO, FIRSTNAME, LASTNAME, JOB
86 ..FROM EMPLOYEE
87 ..WHERE WORKDEPT = :DEPTNO;
88
89 ..EXEC SQL OPEN empCur;
90
91 ..if (sqlstate = '00000'); Same string literal used more than once. Consider using
92
93 ..dou (sqlstate <> '00000'); Same string literal used more than once. Consider u
94 ..EXEC SQL
95 ..FETCH NEXT FROM empCur
96 ..INTO :Employee.EMPNO,
97 ..:Employee.FIRSTNAME,
98 ..:Employee.LASTNAME,
99 ..:Employee.JOB;
100
101 ..if (sqlstate = '00000'); Same string literal used more than once. Consider u
```

****Note – the linter is for **FREE format only**

12

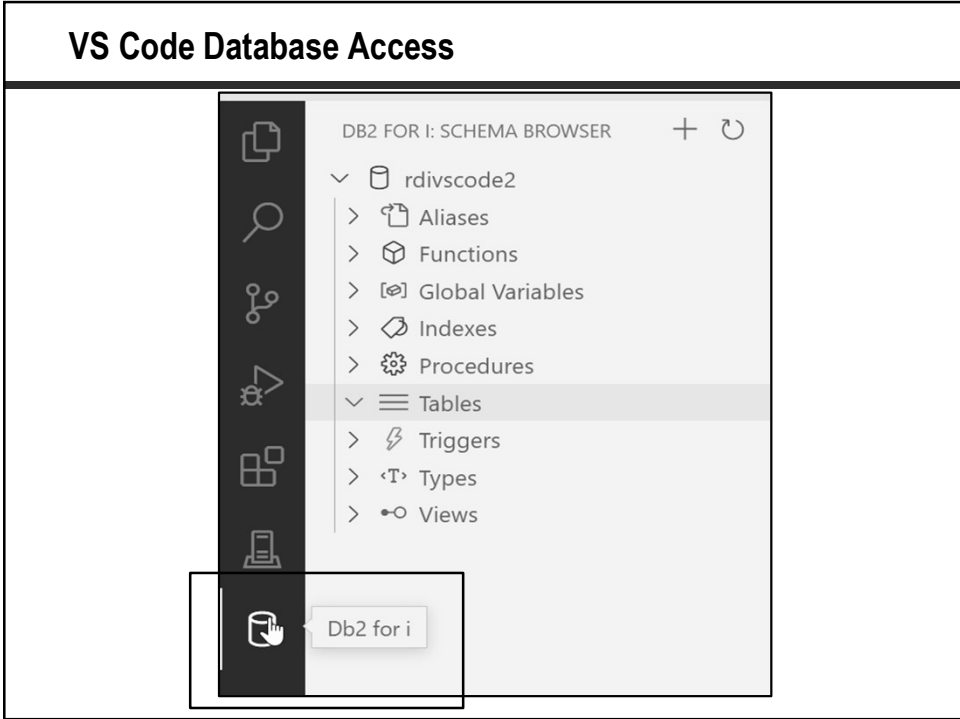
VS Code source code linter properties



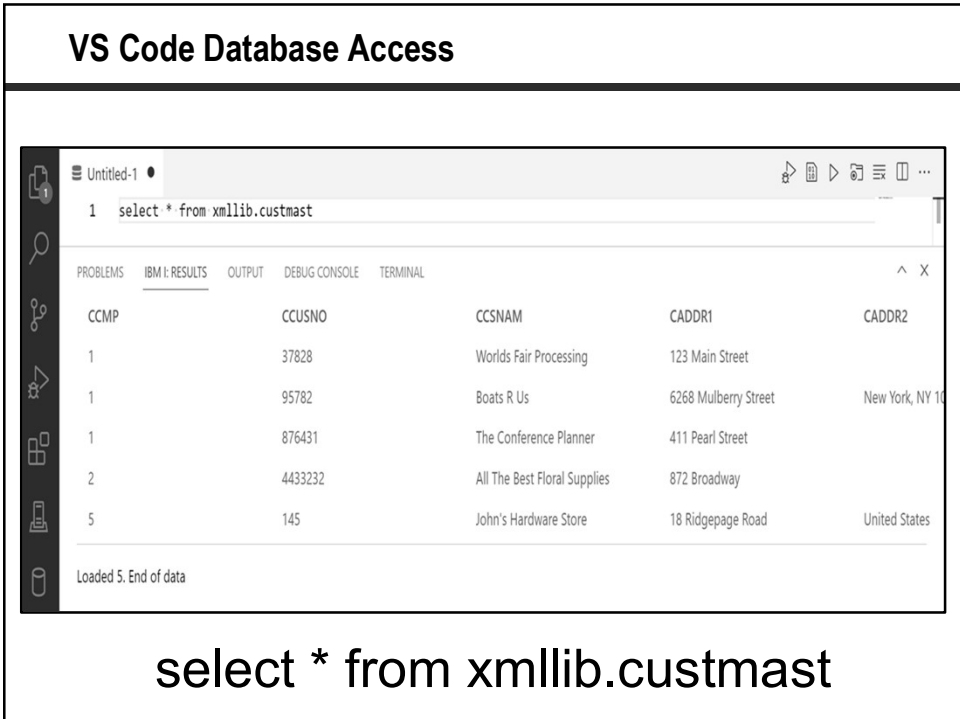
```
employees.pgm.sqlrpgle 4 • {} rpglint.json X
bmi-company_system-main > .vscode > {} rpglint.json > ...
1 {
2   "indent": 2,
3   "BlankStructNamesCheck": true,
4   "QualifiedCheck": false,
5   "PrototypeCheck": true,
6   "ForceOptionalParens": true,
7   "NoOCCURS": true,
8   "NoSELECTAll": true,
9   "UppercaseConstants": true,
10  "IncorrectVariableCase": true,
11  "StringLiteralDupe": true,
12  "NoSQLJoins": true,
13  "NoCTDATA": true,
14  "PrettyComments": true,
15  "NoGlobalSubroutines": true,
16  "NoLocalSubroutines": true,
17  "UppercaseDirectives": true,
18  "NoUnreferenced": false,
19  "NoExternalTo": ["qsygetph"],
20  "NoExtProgramVariable": true
21 }
```

rpglint.json

13



16



17

VS Code Database Access

The screenshot shows the Visual Studio Code interface with a query editor at the top and a Results pane at the bottom. The query editor contains a CSV query: `csv:select * from xmllib.custmast`. The Results pane displays a table with the following data:

CCMP	CCUSNO	CCSNAM	CADDR1	CADDR2
1	37828	Worlds Fair Processing	123 Main Street	
1	95782	Boats R Us	6268 Mulberry Street	New York, NY 1
1	876431	The Conference Planner	411 Pearl Street	
2	4433232	All The Best Floral Supplies	872 Broadway	
5	145	John's Hardware Store	18 Ridgepage Road	United States

`csv:select * from xmllib.custmast`

18

VS Code Database Access

The screenshot shows the Visual Studio Code interface with a query editor at the top and a Results pane at the bottom. The query editor contains a JSON query: `json:select * from xmllib.custmast`. The Results pane displays a table with the following data:

CCMP	CCUSNO	CCSNAM	CADDR1	CADDR2
1	37828	Worlds Fair Processing	123 Main Street	
1	95782	Boats R Us	6268 Mulberry Street	New York, NY
1	876431	The Conference Planner	411 Pearl Street	

`json:select * from xmllib.custmast`

19

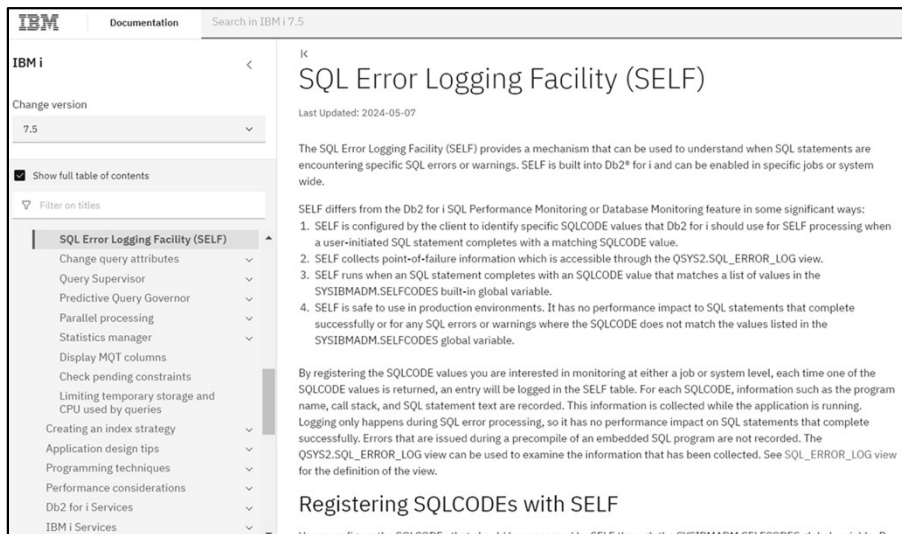
VS Code Database Access



SQL prompting

20

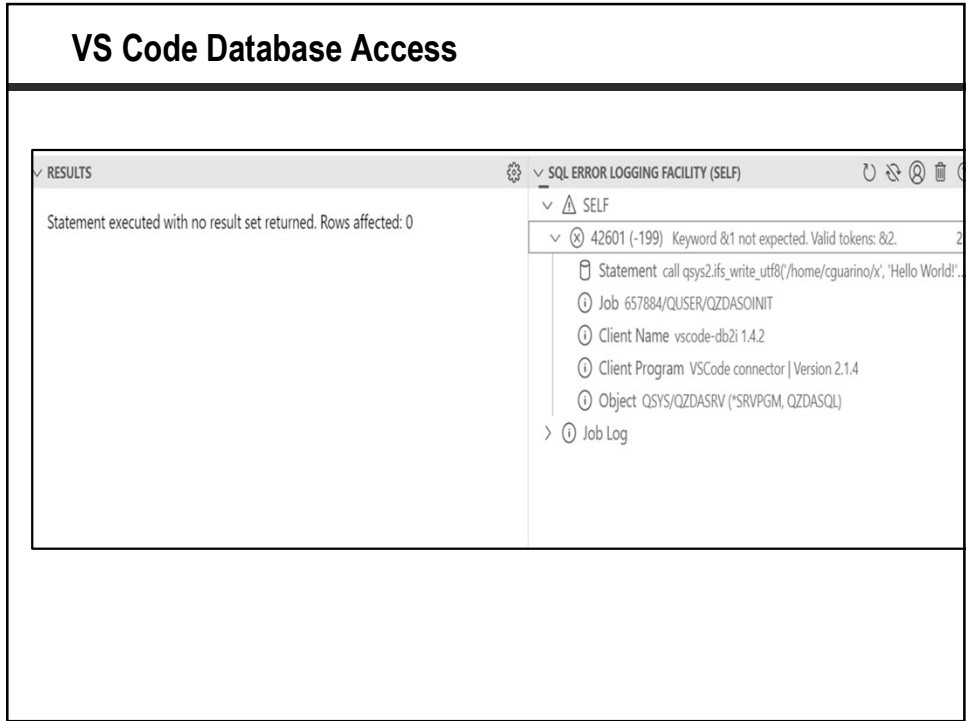
VS Code Database Access



<https://www.ibm.com/docs/en/i/7.5?topic=tools-sql-error-logging-facility-self>

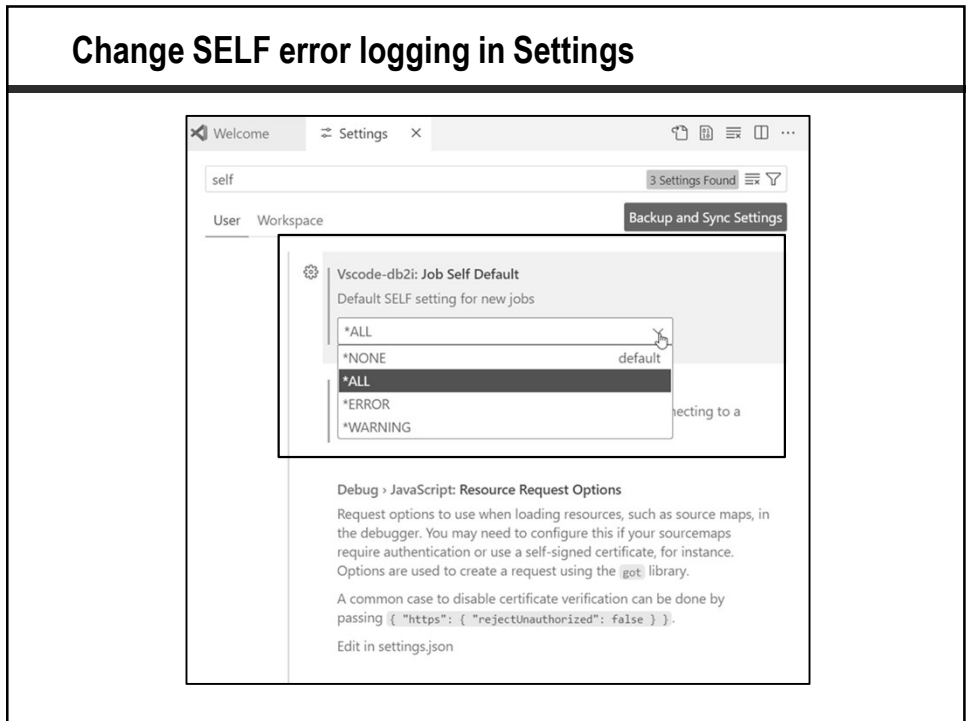
21

VS Code Database Access



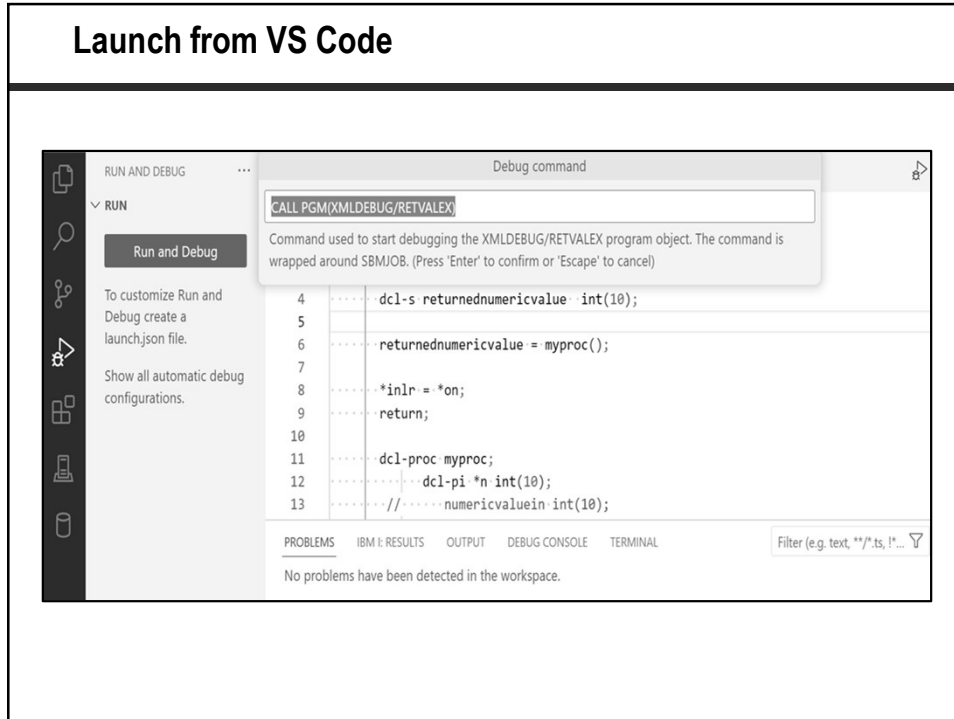
22

Change SELF error logging in Settings



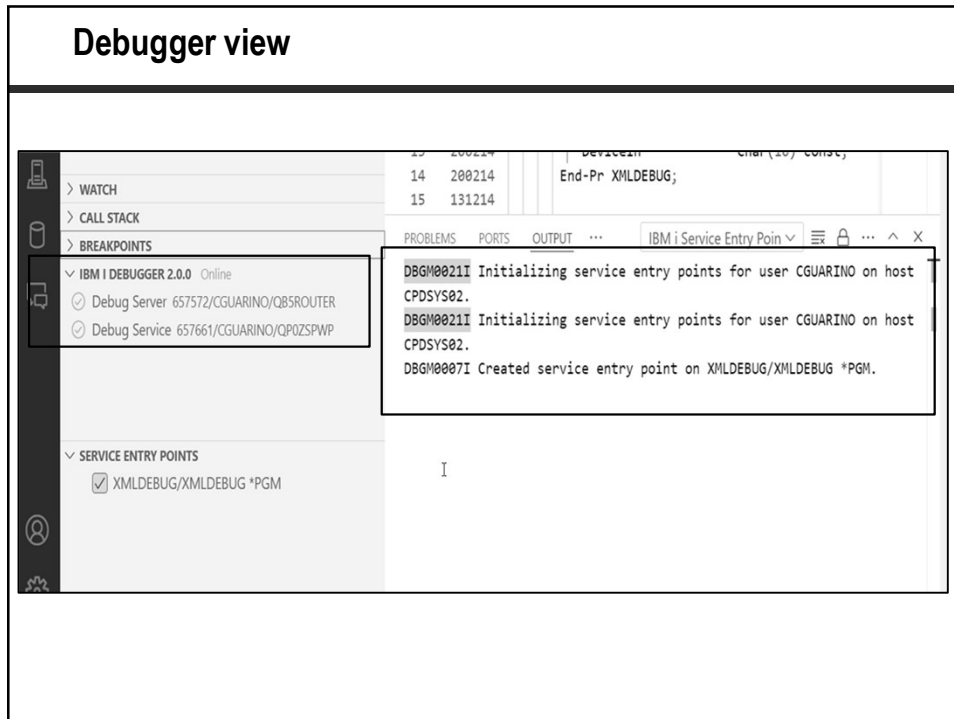
23

Launch from VS Code



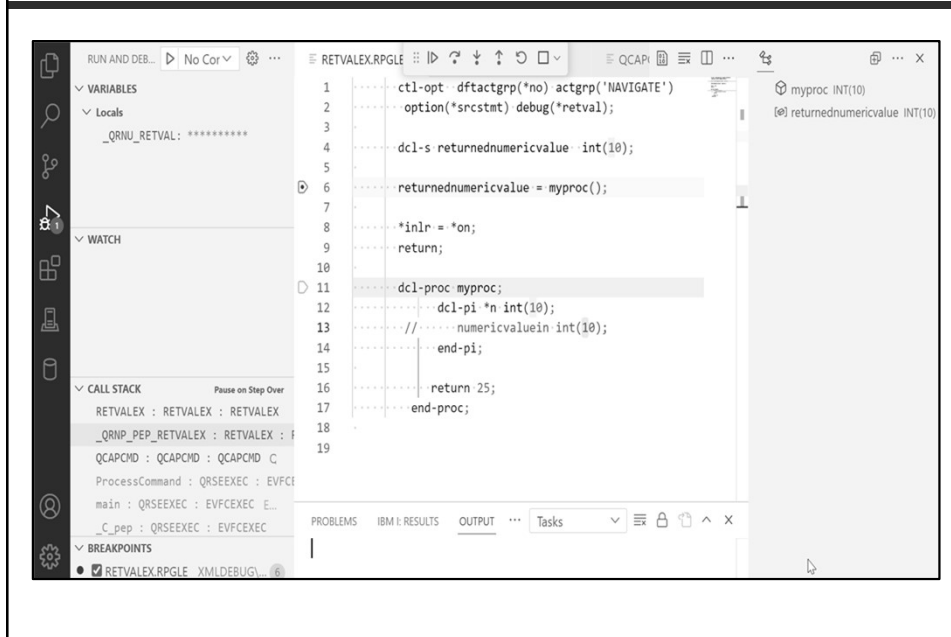
26

Debugger view



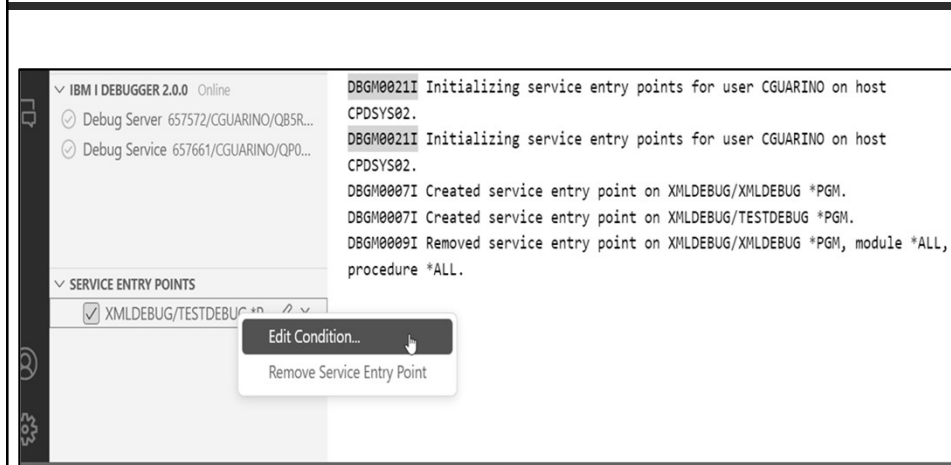
27

VS Code debugger



28

Conditional breakpoints



29

Conditional breakpoints

The screenshot shows a debugger interface. A dialog box titled "Service Entry Point Condition (Press 'Enter' to confirm or 'Escape' to cancel)" is open, with the text "parm1 = 'X'" entered in the input field. Below the dialog, a code editor displays the following code:

```
XMLDEBUG > QRPGLSRC > TESTDEBUG.RPGL > ...  
16 200216 Parm2 Char(1);  
17 200216 Parm3 Char(1);  
18 200216 Parm4 Char(1);  
19 200216 End-Pi TESTDEBUG;  
20 200214  
21 200214  
22 220203 if %parms = *zero;  
23 220203 *inlr = *on;  
24 220203 return;  
25 220203 endif;
```

30

Conditional breakpoints

The screenshot shows a debugger interface. On the left, a list titled "SERVICE ENTRY POINTS" contains one entry: "XMLDEBUG/TESTDEBUG/P...". A dialog box is open over this entry, showing the condition "parm1 = 'X'". On the right, a log window displays the following messages:

```
DBGM0007I Created service entry point on XMLDEBUG/TESTDEBUG/P...  
DBGM0009I Removed service entry point on XMLDEBUG/XML...  
procedure *ALL.  
DBGM0016I Set condition "parm1 = 'X'" on service entry point  
TESTDEBUG *PGM, module *ALL, procedure *ALL.
```

31

What We'll Cover ...

Introduction

VS Code for IBM i

Source code editing

DB2 extension

Debugging

Rational Developer for IBM i

Workbench

Source code editing

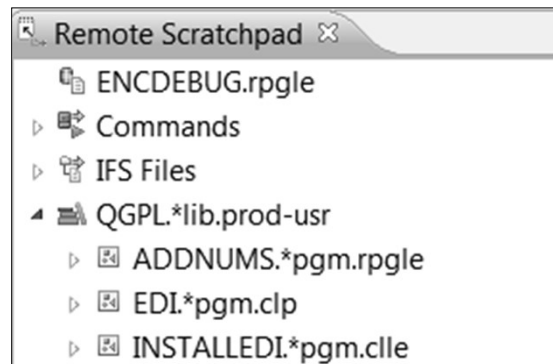
Debugging

Wrap Up

32

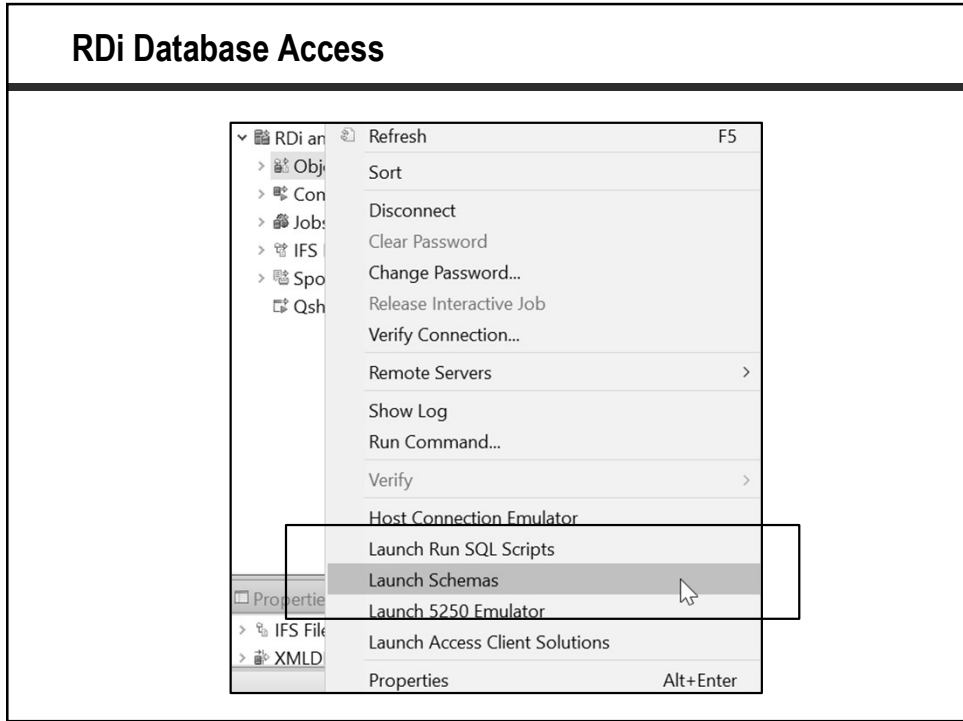
The Remote Scratchpad

- The graphical equivalent to a yellow sticky note
- Provides a convenient place to temporarily store objects
- Contents are removed from the scratchpad when you exit RDi



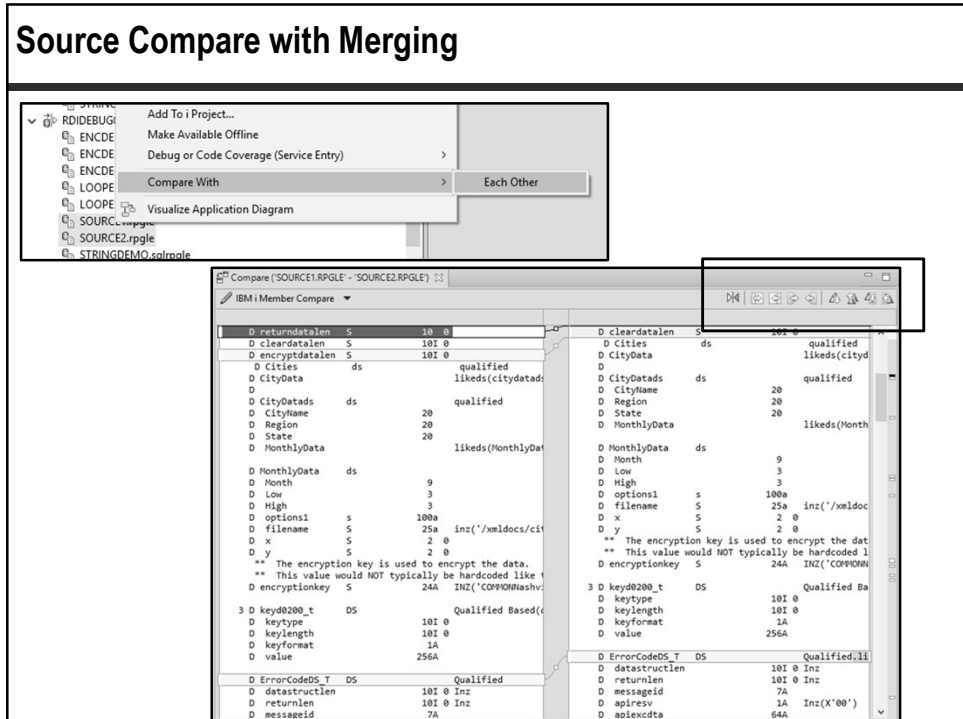
33

RDi Database Access



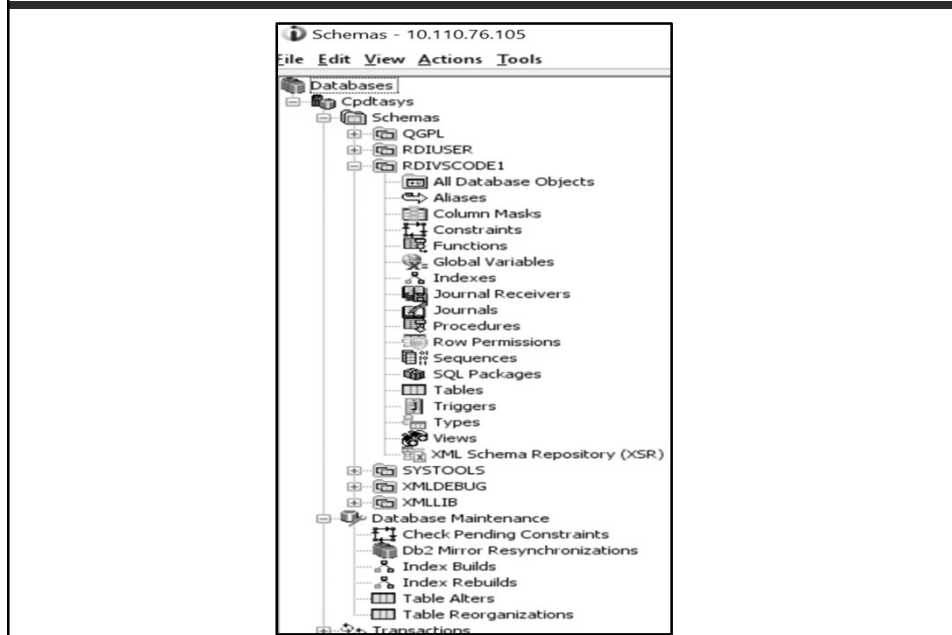
34

Source Compare with Merging



35

RDi Launches ACS



36

What We'll Cover ...

Introduction

VS Code for IBM i

Source code editing

DB2 extension

Debugging

Rational Developer for IBM i

Workbench

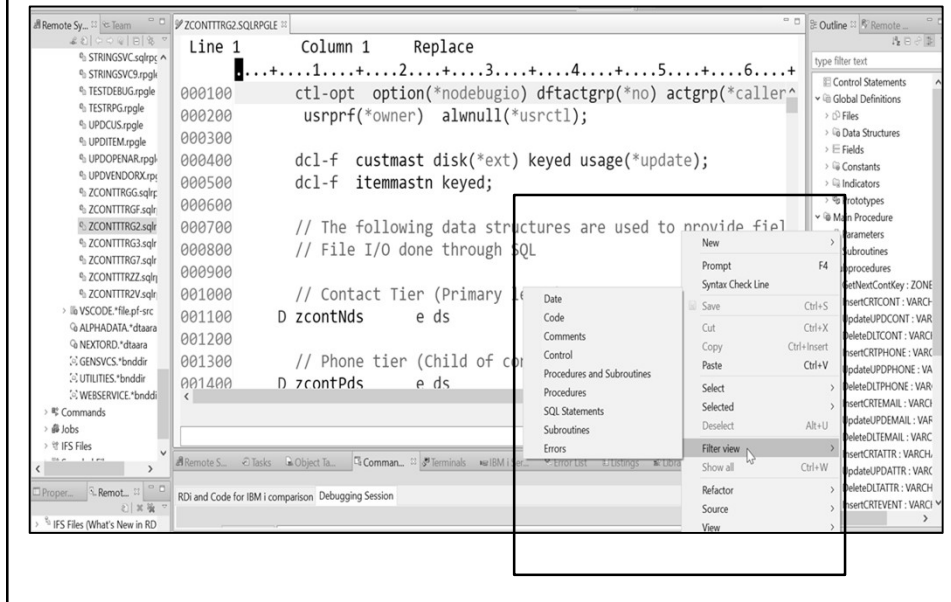
Source code editing

Debugging

Wrap Up

37

RDi Basic editing features



38

Contest Assist with BIFs

x = %sub

- %subarr(array:start-index|numeric-field)
- %subarr(array:start-index|numeric-field:number-of-e
- %subdt(date|time|timestamp:"ms"|"s"|"mn"|"h"|"d"|"m")
- %subst(string:start-index)
- %subst(string:start-index:length)

%SUBARR(array:start-index | numeric-field)

%SUBARR (Set/Get Portion of an Array) returns a section of the specified array starting at start-index. The number of elements returned is specified by the optional number-of-elements parameter. If not specified, the number-of-elements defaults to the remainder of the array.

```

%SUBARR
(
  array
  start-index | numeric-field
  number-of-elements | numeric-field
)
x = %subarr (myarray : : )
    
```

39

RDi Content assist with templates

```
sqlcode
SQLCODE - Process the SQLCODE return code from e // Check SQLCODE:
Select;
When SqlCode 0;
//execution of SQL statement was not successful
${cursor}
When SqlCode = 100;
//execution of SQL statement resulted in no data
When SqlCode 0;
//execution of SQL statement was successful, with a warning
When SqlWn0 = 'W';
//execution of SQL statement was successful, with a warning
Other;
//execution of SQL statement was successful
*****Endsl;
```

40

RDi Content assist with templates

```
// Check SQLCODE:
Select;
  When SqlCode < 0;
    //execution of SQL statement was not successful

  When SqlCode = 100;
    //execution of SQL statement resulted in no data
  When SqlCode > 0;
    //execution of SQL statement was successful, with a warning
  When SqlWn0 = 'W';
    //execution of SQL statement was successful, with a warning
  Other;
    //execution of SQL statement was successful
Endsl;
```

41

RDi Hovering

```
Line 195      Column 34      Replace
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....
018200      endif;
018300      else;
018400      a = 112233;
018500      EndIf;
018600
018601      // to copy from a source physical file
018700      /copy qcpysrc,encdebug01
018701
018702      *LIBL/qcpysrc,encdebug01
018703      direction = 'E'; // Value of 'E' tells procedure to ENCRYPT
018704      // t      cleardata = cclrdata;
018800      /copy      encrypteddata = secretdata(cleardata:direction);
018801
018900
018900
019000
019100      inputstring = data;
019200
019300
<
(Main Procedure)
```

42

Show Block Nesting

```
if a = 6;
if b = 5;
select;
when customer = 456;
b = 4;
when customer = 123;
b = 7;
other;
b = 10;
endsl;
endif;
else;
a = 112233;
EndIf;
```

```
0 → if a = 6;
0 → if b = 5;
0 → select;
0 → when customer = 456;
0 → b = 4;
0 → when customer = 123;
0 → b = 7;
0 → other;
0 → b = 10;
0 → ends;
0 → endif;
0 → else;
0 → a = 112233;
0 → EndIf;
```

43

Formatting a Nested Block

```
if a = 6;
  if b = 5;
    select;
    when customer = 456;
      b = 4;
    when customer = 123;
      b = 7;
    other;
      b = 10;
    ends1;
  endif;
else;
  a = 112233;
EndIf;
```

```
if a = 6;
  if b = 5;
    select;
    when customer = 456;
      b = 4;
    when customer = 123;
      b = 7;
    other;
      b = 10;
    ends1;
  endif;
else;
  a = 112233;
EndIf;
```

44

What We'll Cover ...

Introduction

VS Code for IBM I

Source code editing

DB2 extension

Debugging

Rational Developer for IBM i

Workbench

Source code editing

Debugging

Wrap Up

45

Calling program TESTDEBUG

```
TESTDEBUG.RPGLE
Line 28      Column 3      Replace
..+.....1.....2.....3.....4.....5.....+..
000700      ctl-opt dftactgrp(*no) actgrp('TESTDEBUG')
000800              option(*srcstmt : *nodebugio) debug(*input);
000900
001200      Dcl-Pr TESTDEBUG ExtPgm;
001300          Parm1          Char(1);
001301          Parm2          Char(1);
001302          Parm3          Char(1);
001303          Parm4          Char(1);
001400      End-Pr TESTDEBUG;
001500
002700      // Main procedure interface
002800      Dcl-Pi TESTDEBUG;
002801          Parm1          Char(1);
002802          Parm2          Char(1);
002803          Parm3          Char(1);
002804          Parm4          Char(1);
003000      End-Pi TESTDEBUG;
003100
003200
003501          parm1 = parm1;
003503          parm2 = parm2;
003505          parm3 = parm3;
003507          parm4 = parm4;
005900
006000      *inlr = *on;
006100      return;
```

46

Calling program TESTDEBUG

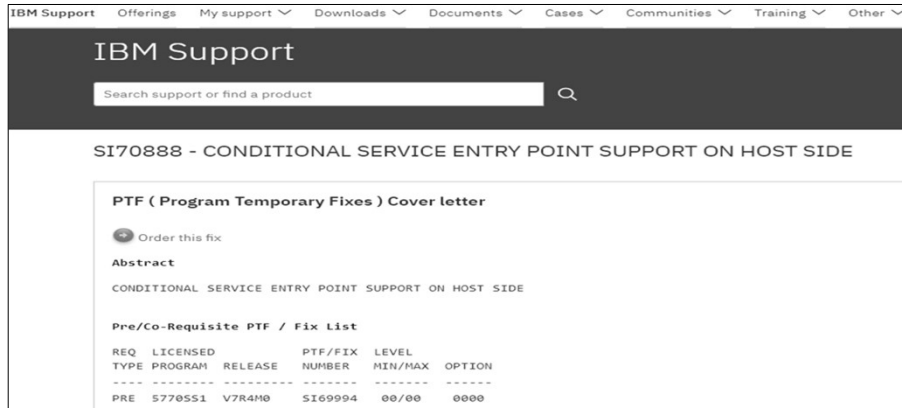
```
A - 10.110.76.105
File Edit View Communication Actions Window Help
MAIN                               IBM i Main Menu                               System: CPDSYS02
Select one of the following:
1. User tasks
2. Office tasks
3. General system tasks
4. Files, libraries, and folders
5. Programming
6. Communications
7. Define or change the system
8. Problem handling
9. Display a menu
10. Information Assistant options
11. IBM i Access tasks
90. Sign off
Selection or command
==> CALL PGM(TESTDEBUG) PARM('A' 'B' 'C' 'D')
F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant
F23=Set initial menu
```

47

SEP with conditions test cases

SEPs now allow conditional entries for breakpoints

- PTF SI70888 for IBM i 7.4
- PTF SI70844 for IBM i 7.3



IBM Support

Search support or find a product

SI70888 - CONDITIONAL SERVICE ENTRY POINT SUPPORT ON HOST SIDE

PTF (Program Temporary Fixes) Cover letter

Order this fix

Abstract

CONDITIONAL SERVICE ENTRY POINT SUPPORT ON HOST SIDE

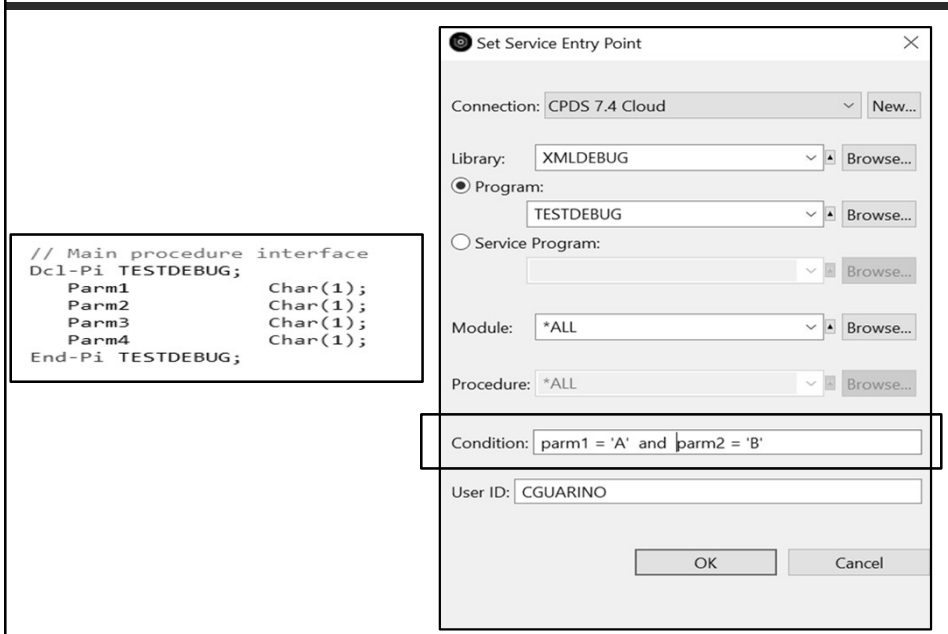
Pre/Co-Requisite PTF / Fix List

REQ	LICENSED	PTF/FIX	LEVEL		
TYPE	PROGRAM	NUMBER	MIN/MAX	OPTION	
PRE	5770551	V7R4M0	SI69994	00/00	0000

<https://www-01.ibm.com/support/docview.wss?uid=nas3SI70888>

48

SEP with conditions test cases



// Main procedure interface
Dcl-Pi TESTDEBUG;
 Parm1 Char(1);
 Parm2 Char(1);
 Parm3 Char(1);
 Parm4 Char(1);
End-Pi TESTDEBUG;

Set Service Entry Point

Connection: CPDS 7.4 Cloud

Library: XMLDEBUG

Program: TESTDEBUG

Service Program:

Module: *ALL

Procedure: *ALL

Condition: parm1 = 'A' and parm2 = 'B'

User ID: CGUARINO

OK Cancel

49

What We'll Cover ...

Introduction

VS Code for IBM i

Source code editing

DB2 extension

Debugging

Rational Developer for IBM i

Workbench

Source code editing

Debugging

Wrap Up

50

Unleash the Power of RDi and VS Code for IBM i



**Charles Guarino
THANK YOU!!!!**

51