Refactoring With a Splash Of Modern Tooling

Charles Guarino
Central Park Data Systems, Inc.

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 though 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

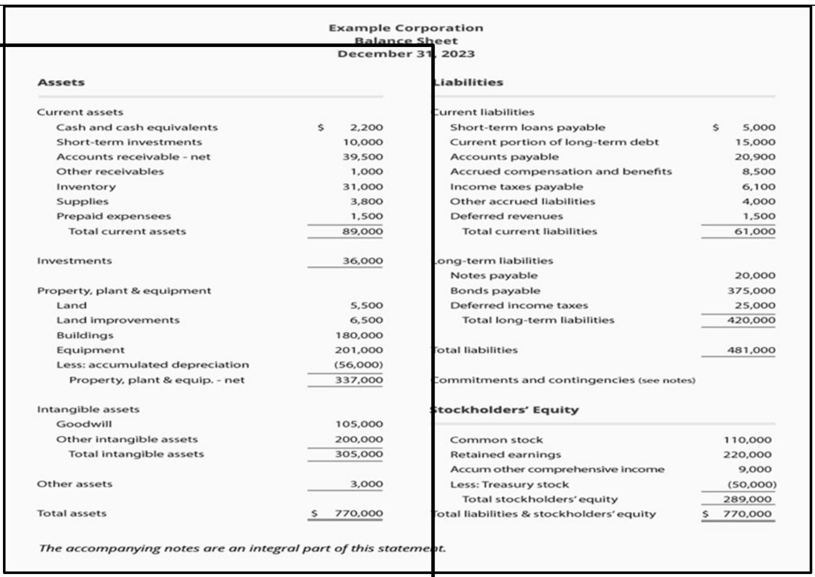
1

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

2

Sample balance sheet



Example Corporation
Balance Sheet
December 31, 2023

Assets		
Current assets		
Cash and cash equivalents	$	2,200
Short-term investments		10,000
Accounts receivable - net		39,500
Other receivables		1,000
Inventory		31,000
Supplies		3,800
Prepaid expensees		1,500
Total current assets		89,000
Investments		36,000
Property, plant & equipment		
Land		5,500
Land improvements		6,500
Buildings		180,000
Equipment		201,000
Less: accumulated depreciation		(56,000)
Property, plant & equip. - net		337,000
Intangible assets		
Goodwill		105,000
Other intangible assets		200,000
Total intangible assets		305,000
Other assets		3,000
Total assets	$	770,000

Liabilities		
Current liabilities		
Short-term loans payable	$	5,000
Current portion of long-term debt		15,000
Accounts payable		20,900
Accrued compensation and benefits		8,500
Income taxes payable		6,100
Other accrued liabilities		4,000
Deferred revenues		1,500
Total current liabilities		61,000
Long-term liabilities		
Notes payable		20,000
Bonds payable		375,000
Deferred income taxes		25,000
Total long-term liabilities		420,000
Total liabilities		481,000
Commitments and contingencies (see notes)		
Stockholders' Equity		
Common stock		110,000
Retained earnings		220,000
Accum other comprehensive income		9,000
Less: Treasury stock		(50,000)
Total stockholders' equity		289,000
Total liabilities & stockholders' equity	$	770,000

The accompanying notes are an integral part of this statement.

https://www.accountingcoach.com/wp-content/uploads/2013/10/balance-sheet-example.png

3

YOUR CODEBASE has value and is a very valuable strategic asset

4

Comparing Your Codebase to Physical Assets

Requires maintenance to stay competitive

Will become outdated if not kept current

Maintenance shortcuts need to be addressed

5

The “problem” that needs to be explained

To the untrained eye, software and your overall codebase does not show rust or signs of aging.

6

Plain and Simple

“As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.”

Meir Manny Lehman, 1980
Computer Scientist

https://en.wikipedia.org/wiki/Manny_Lehman_(computer_scientist)

7

Technical Debt

Technical debt refers to the future costs incurred when development teams prioritize quick delivery over optimal code quality.

This often involves implementing expedient solutions that may require rework or maintenance later. If not addressed, technical debt can accumulate, leading to increased complexity and higher costs for future changes.

https://www.productplan.com/glossary/technical-debt/

8

Calculating Interest Paid When Making Minimum Credit Card Payments

$$I = \frac{APR}{12} \times B$$

$$N = (B + I) - P$$

I = interest

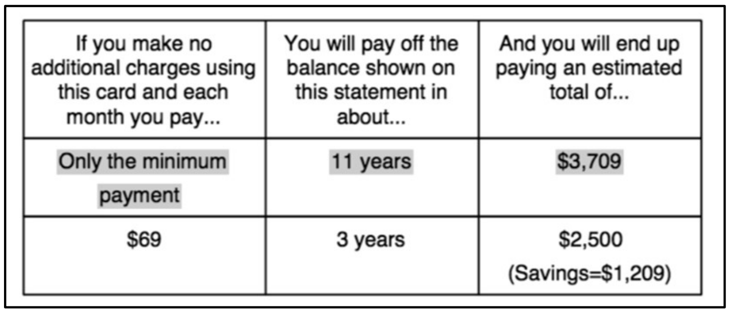
B = original balance

N = new balance

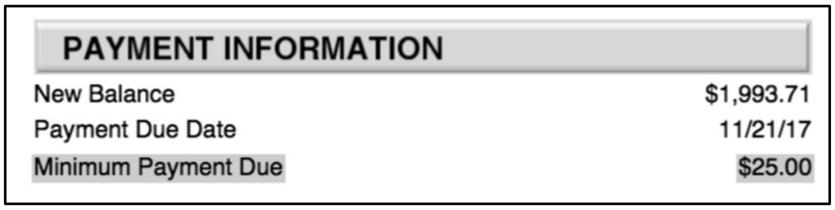
P = monthly payment

9

Paying Just the Minimum Payment can be Very Costly



If you make no additional charges using this card and each month you pay...	You will pay off the balance shown on this statement in about...	And you will end up paying an estimated total of...
Only the minimum payment	11 years	$3,709
$69	3 years	$2,500 (Savings=$1,209)



PAYMENT INFORMATION	
New Balance	$1,993.71
Payment Due Date	11/21/17
Minimum Payment Due	$25.00

https://www.aboveboardfinancial.com/blog/carry-a-credit-card-balance-learn-the-3-things-on-your-credit-card-statement-that-you-cant-afford-to

10

Calculating Technical Debt

$$I = \frac{APR}{12} \times B$$

$$N = (B + I) - P$$

Each new requirement is a "purchase"

I = quality of new code

B = original technical debt to maintain codebase

N = new technical debt to maintain codebase

P = cost and quality of programming enhancements

11

Beware Code Rot

As code is continually modified and maintained over time with possibly imperfect changes, more and more bugs get introduced.

Worse, it gets more and more “correct but ugly.”

This reduces the integrity of the code, “rotting” it until it eventually falls apart.

https://softwareengineering.stackexchange.com/questions/255866/what-is-meant-by-code-rot

12

Plain and Simple

An asset becomes a liability when it no longer generates economic benefits for its owner

13

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

14

Refactoring 101

Source code refactoring is the process of restructuring existing computer code without changing its external behavior.

The primary goal is to improve the design, structure, and implementation of the software, enhancing its readability, maintainability, and extensibility.

This is achieved by making small, incremental changes that simplify the code's internal structure while preserving its functionality.

https://en.wikipedia.org/wiki/Code_refactoring

15

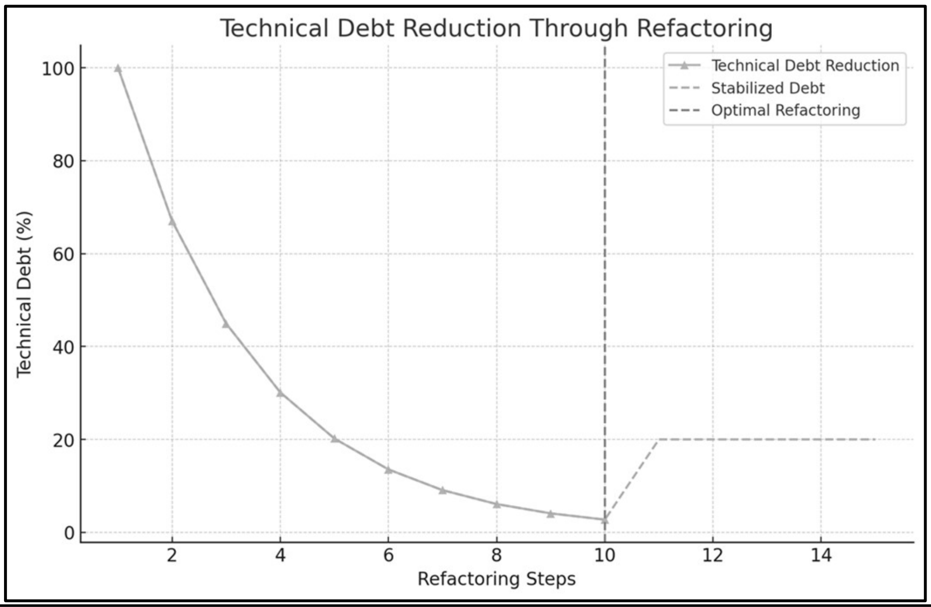
Plain and Simple

"Refactoring is a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior."

Martin Fowler
www.refactoring.com

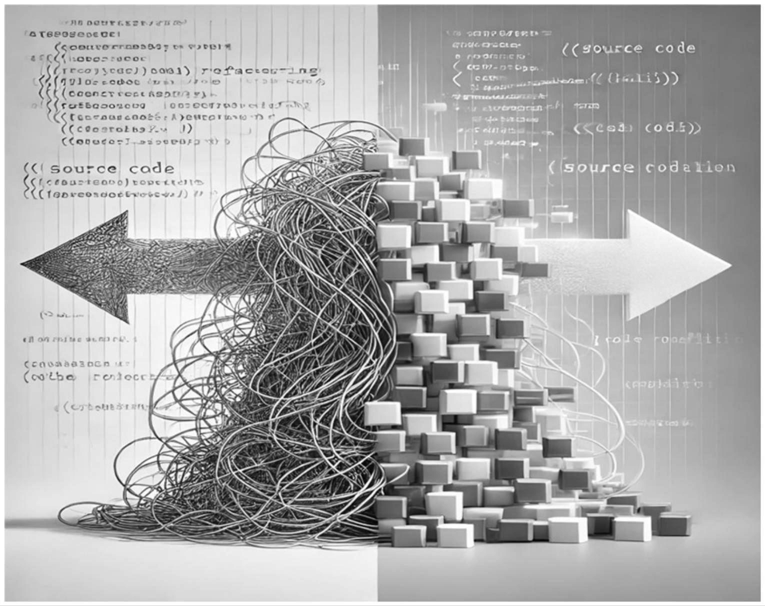
16

Goal – Refactor To Get to Stabilized Debt



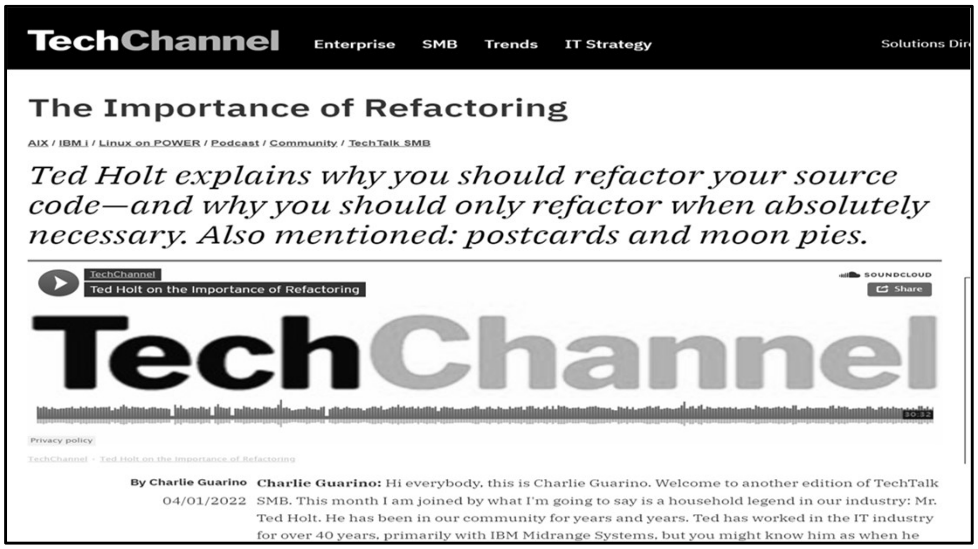
17

From Chaos to Structure



18

(Semi) Recent Podcast! – April 2022



TechChannel Enterprise SMB Trends IT Strategy Solutions Dir

The Importance of Refactoring

AIX / IBM i / Linux on POWER / Podcast / Community / TechTalk SMB

Ted Holt explains why you should refactor your source code—and why you should only refactor when absolutely necessary. Also mentioned: postcards and moon pies.

TechChannel
Ted Holt on the Importance of Refactoring

SOUNDCLOUD
Share

TechChannel

30:32

Privacy policy

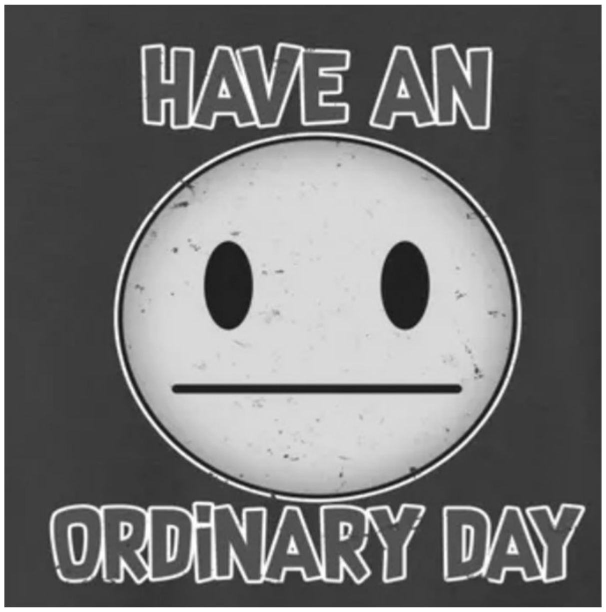
TechChannel · Ted Holt on the Importance of Refactoring

By Charlie Guarino
04/01/2022

Charlie Guarino: Hi everybody, this is Charlie Guarino. Welcome to another edition of TechTalk SMB. This month I am joined by what I'm going to say is a household legend in our industry: Mr. Ted Holt. He has been in our community for years and years. Ted has worked in the IT industry for over 40 years, primarily with IBM Midrange Systems, but you might know him as when he

https://techchannel.com/Trends/04/2022/ted-holt-refactoring

19



20

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

21

Software is an expensive asset to maintain

22

Enhancements in functionality is the bulk of software cost

23

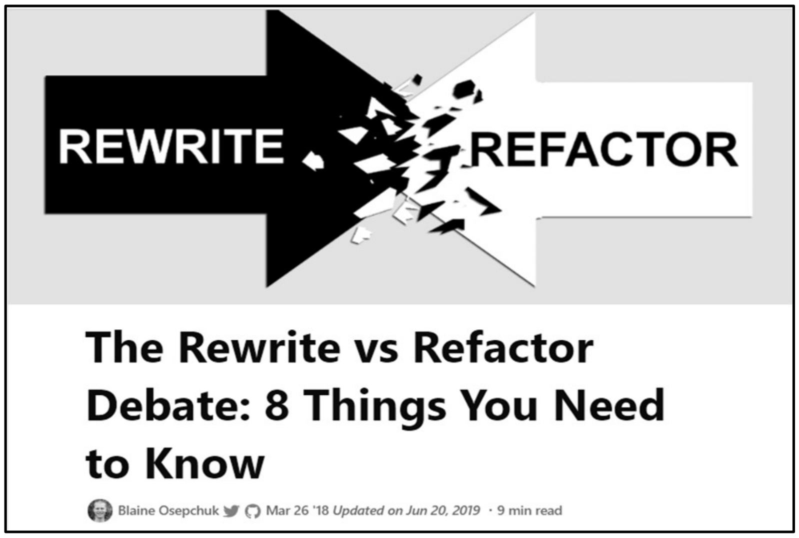
New functions and processes take too long to deploy

24

Business requirements are always changing

25

Rewrite vs refactor – the controversy continues….



The Rewrite vs Refactor Debate: 8 Things You Need to Know

Blaine Osepchuk Mar 26 '18 *Updated on Jun 20, 2019* · 9 min read

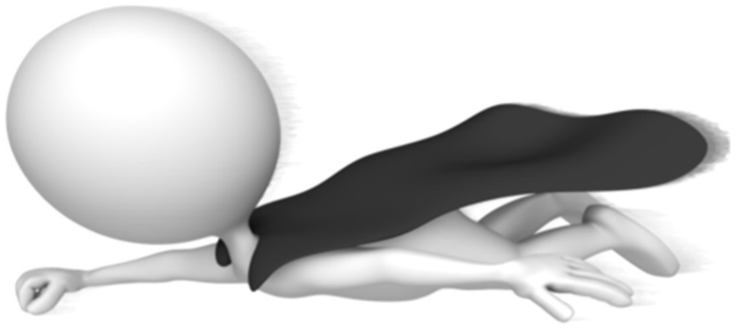
https://dev.to/bosepchuk/the-rewrite-vs-refactor-debate-8-things-you-need-to-know-2hi4

26

Does the developer have the right skills to rewrite or refactor?

27

Does faster hardware make up for poorly written code?



28

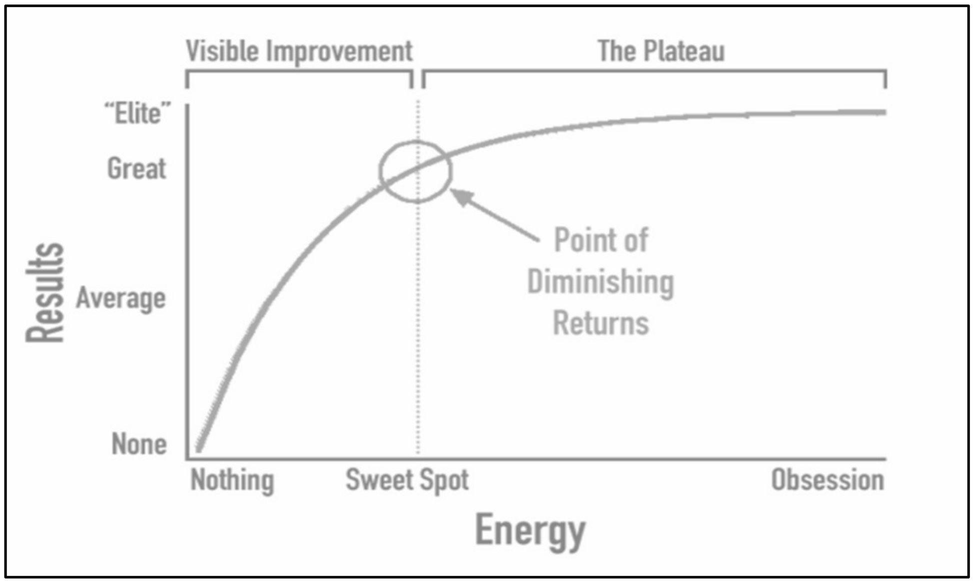
New Developers Cannot or Will Not Work on Older Code

29

When is “good enough” “good enough?”

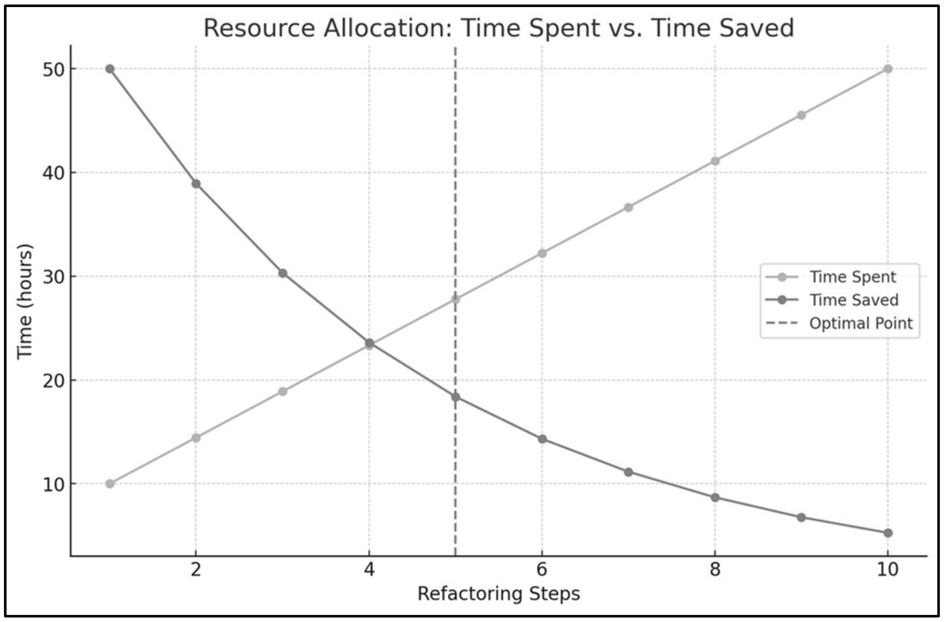
30

Where is the sweet spot?



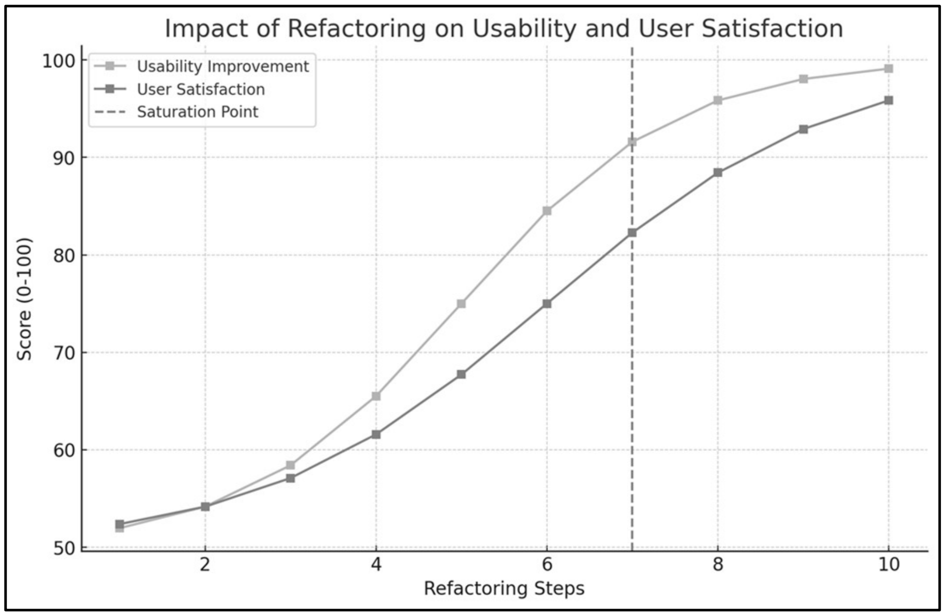
31

Point of Diminishing Returns



32

Point of Diminishing Returns



33

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

34

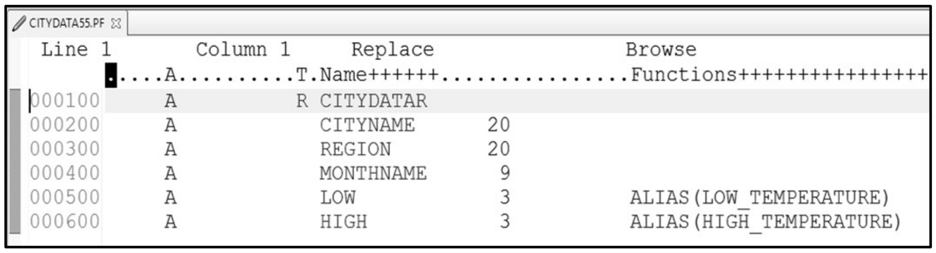
Why "refactor" DDS? Benefits of DDL over DDS

- DDS is IBM i specific, SQL is universal
- SQL and DDL are widely taught, DDS is not
- DDS is stabilized, last enhancement was in V5R3
- DDL supports CLOB, DBCLOB, BLOB, XML, ROWID
- Identity columns
- Data is validated on INSERT, not on the READ
- Longer column names support, up to 128 characters
  - See new ALIAS keyword for RPG!

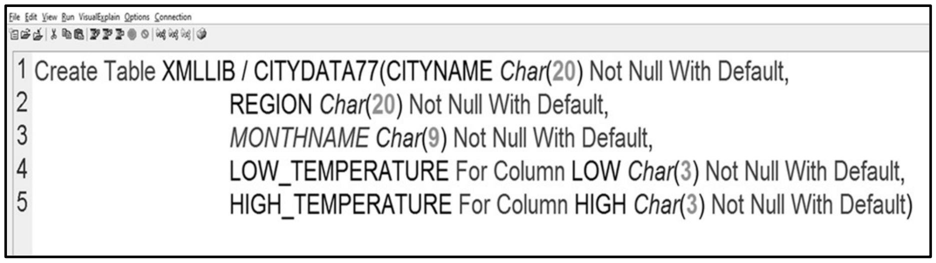
- And other performance and I/O benefits

35

Fun Fact! Alias keyword allows you to read long field names



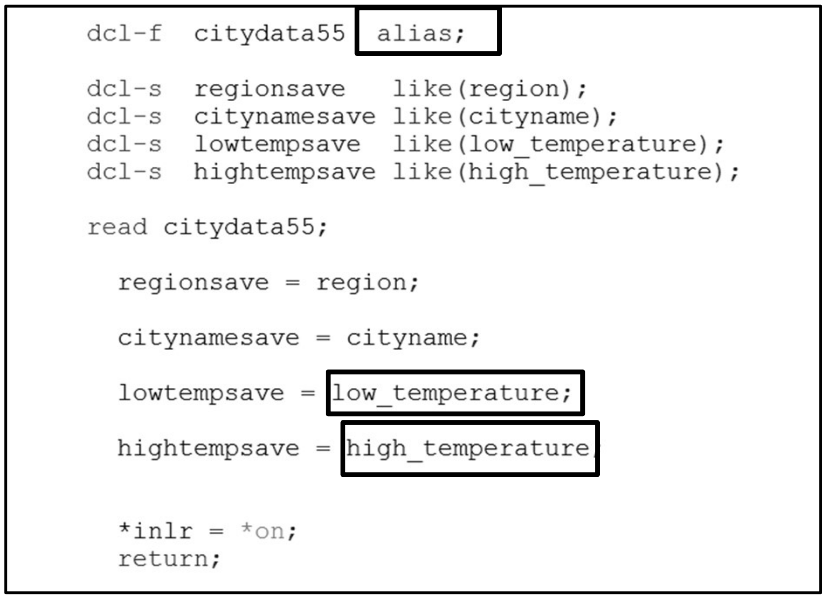
```
 Line 1       Column 1     Replace               Browse
       ....A..........T.Name++++++................Functions++++++++++++++++
000100     A          R CITYDATAR
000200     A            CITYNAME      20
000300     A            REGION        20
000400     A            MONTHNAME      9
000500     A            LOW            3          ALIAS(LOW_TEMPERATURE)
000600     A            HIGH           3          ALIAS(HIGH_TEMPERATURE)
```



```
Create Table XMLLIB / CITYDATA77(CITYNAME Char(20) Not Null With Default,
                      REGION Char(20) Not Null With Default,
                      MONTHNAME Char(9) Not Null With Default,
                      LOW_TEMPERATURE For Column LOW Char(3) Not Null With Default,
                      HIGH_TEMPERATURE For Column HIGH Char(3) Not Null With Default)
```

36

Alias keyword is placed on the F spec or dcl-f statement



```
dcl-f  citydata55  alias;

dcl-s  regionsave   like(region);
dcl-s  citynamesave like(cityname);
dcl-s  lowtempsave  like(low_temperature);
dcl-s  hightempsave like(high_temperature);

read citydata55;

  regionsave = region;

  citynamesave = cityname;

  lowtempsave = low_temperature;

  hightempsave = high_temperature;


  *inlr = *on;
  return;
```

37

How do I start – some ideas

RPG FREE

38

Sample source code changes

Eliminate record level access where appropriate using set processing with SQL

```
 // Delete any records in output file with same path and document name
 // If any are found, they are from a previous run
 dou not %found(saxdata);
   delete(e) (docpath:docname) saxdata;
 enddo;
```

```
// Delete any records in output file with same path and document name
// If any are found, they are from a previous run

 exec sql
 delete from saxdata
  where (xmldocpath = :saxctlds.prcdocpath) and
        (xmldocname = :saxctlds.prcdocname);
```

39

Some classic code smells

Duplicated routines

Contrived complexity – forced usage of complicated design patterns where simpler would have sufficed

Too many parameters

Excessively long or short variable names

Excessively long line of code with many operators

40

Some classic code smells

Duplicated routines

Contrived complexity – forced usage of complicated design patterns where simpler would have sufficed

Too many parameters

Excessively long or short variable names

Excessively long line of code with many operators

41

Some classic code smells

Duplicated routines

Contrived complexity – forced usage of complicated design patterns where simpler would have sufficed

Too many parameters

Excessively long or short variable names

Excessively long line of code with many operators

42

Some classic code smells

Duplicated routines

Contrived complexity – forced usage of complicated design patterns where simpler would have sufficed

Too many parameters

Excessively long or short variable names

Excessively long line of code with many operators

43

Some classic code smells

Duplicated routines

Contrived complexity – forced usage of complicated design patterns where simpler would have sufficed

Too many parameters

Excessively long or short variable names

Excessively long line of code with many operators

44

Using intermediate variables makes the code easier to read and debug

Instead of this:

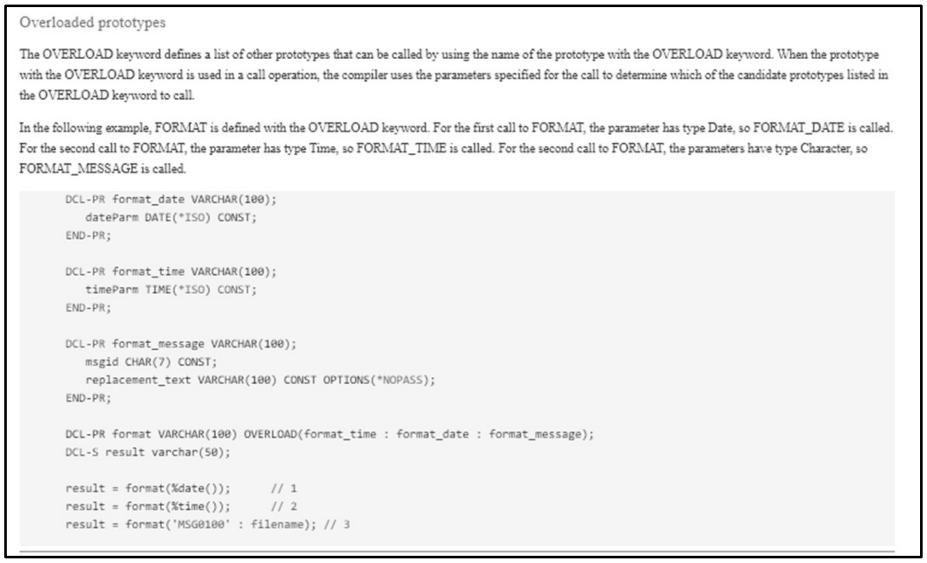
```
If hours <= 40;
wages = hours * hourlyrate;
else;
wages = (hourlyrate * 40) + (hourlyrate * 1.5) * (hours – 40);
endif;
```

Consider this:

```
If hours <= 40;
Wages = hours * hourlyrate;
else;
overtimerate = hourlyrate * 1.5;
overtimehours = hours – 40;
wages = (hourlyrate * 40) + (overtimerate * overtimehours);
endif;
```

45

Some code changes for clarifying procedure calls



Overloaded prototypes

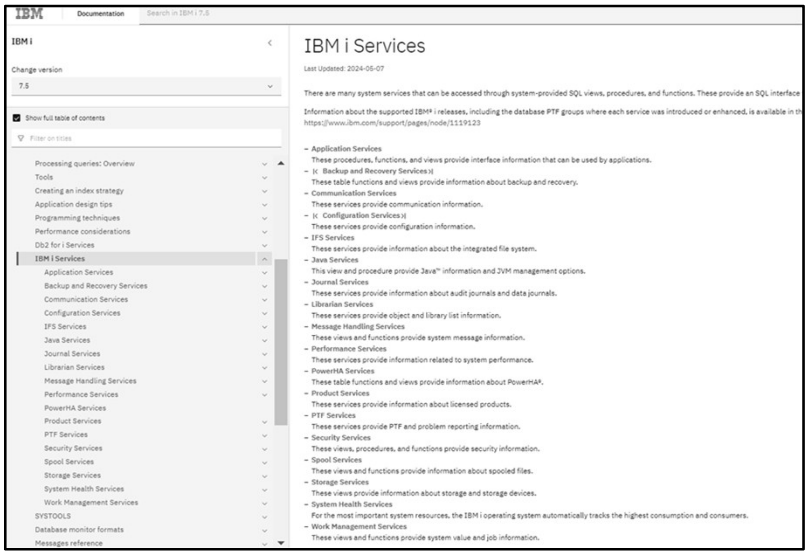
The OVERLOAD keyword defines a list of other prototypes that can be called by using the name of the prototype with the OVERLOAD keyword. When the prototype with the OVERLOAD keyword is used in a call operation, the compiler uses the parameters specified for the call to determine which of the candidate prototypes listed in the OVERLOAD keyword to call.

In the following example, FORMAT is defined with the OVERLOAD keyword. For the first call to FORMAT, the parameter has type Date, so FORMAT_DATE is called. For the second call to FORMAT, the parameter has type Time, so FORMAT_TIME is called. For the second call to FORMAT, the parameters have type Character, so FORMAT_MESSAGE is called.

```
DCL-PR format_date VARCHAR(100);
   dateParm DATE(*ISO) CONST;
END-PR;

DCL-PR format_time VARCHAR(100);
   timeParm TIME(*ISO) CONST;
END-PR;

DCL-PR format_message VARCHAR(100);
   msgid CHAR(7) CONST;
   replacement_text VARCHAR(100) CONST OPTIONS(*NOPASS);
END-PR;

DCL-PR format VARCHAR(100) OVERLOAD(format_time : format_date : format_message);
DCL-S result varchar(50);

result = format(%date());      // 1
result = format(%time());      // 2
result = format('MSG0100' : filename); // 3
```

https://www.ibm.com/support/pages/node/1106409

SQL Services!!!



IBM i Services

Last Updated: 2024-05-07

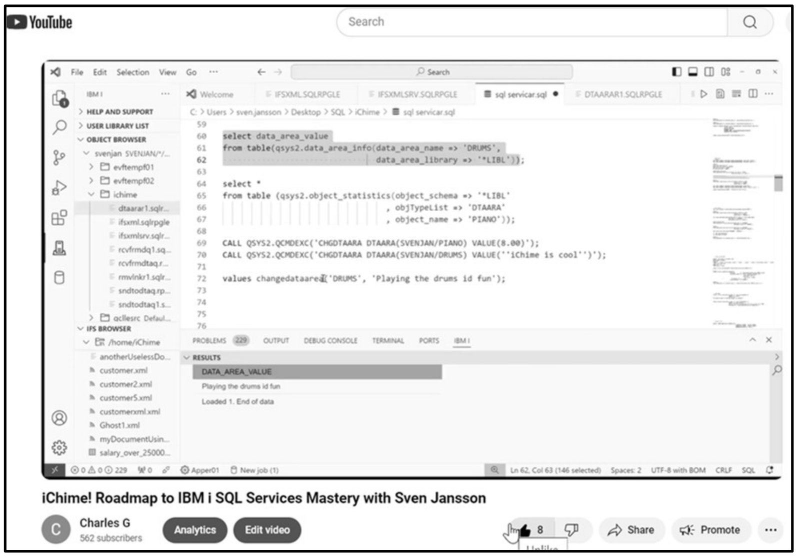
There are many system services that can be accessed through system-provided SQL views, procedures, and functions. These provide an SQL interface

Information about the supported IBM® i releases, including the database PTF groups where each service was introduced or enhanced, is available in th
https://www.ibm.com/support/pages/node/1119123

- Application Services
  These procedures, functions, and views provide interface information that can be used by applications.
- |< Backup and Recovery Services >|
  These table functions and views provide information about backup and recovery.
- Communication Services
  These services provide communication information.
- |< Configuration Services >|
  These services provide configuration information.
- IFS Services
  These services provide information about the integrated file system.
- Java Services
  This view and procedure provide Java™ information and JVM management options.
- Journal Services
  These services provide information about audit journals and data journals.
- Librarian Services
  These services provide object and library list information.
- Message Handling Services
  These views and functions provide system message information.
- Performance Services
  These services provide information related to system performance.
- PowerHA Services
  These table functions and views provide information about PowerHA®.
- Product Services
  These services provide information about licensed products.
- PTF Services
  These services provide PTF and problem reporting information.
- Security Services
  These views, procedures, and functions provide security information.
- Spool Services
  These views and functions provide information about spooled files.
- Storage Services
  These views provide information about storage and storage devices.
- System Health Services
  For the most important system resources, the IBM i operating system automatically tracks the highest consumption and consumers.
- Work Management Services
  These views and functions provide system value and job information.

https://www.ibm.com/docs/en/i/7.5?topic=optimization-i-services

SQL Services in Action – iChime video (register at ichime.io)



```
select data_area_value
from table(qsys2.data_area_info(data_area_name => 'DRUMS',
                                data_area_library => '*LIBL'));

select *
from table (qsys2.object_statistics(object_schema => '*LIBL'
                                    , objTypeList => 'DTAARA'
                                    , object_name => 'PIANO'));

CALL QSYS2.QCMDEXC('CHGDTAARA DTAARA(SVENJAN/PIANO) VALUE(8.00)');
CALL QSYS2.QCMDEXC('CHGDTAARA DTAARA(SVENJAN/DRUMS) VALUE(''iChime is cool'')');

values changedataarea('DRUMS', 'Playing the drums id fun');
```

```
DATA_AREA_VALUE
Playing the drums id fun
Loaded 1. End of data
```

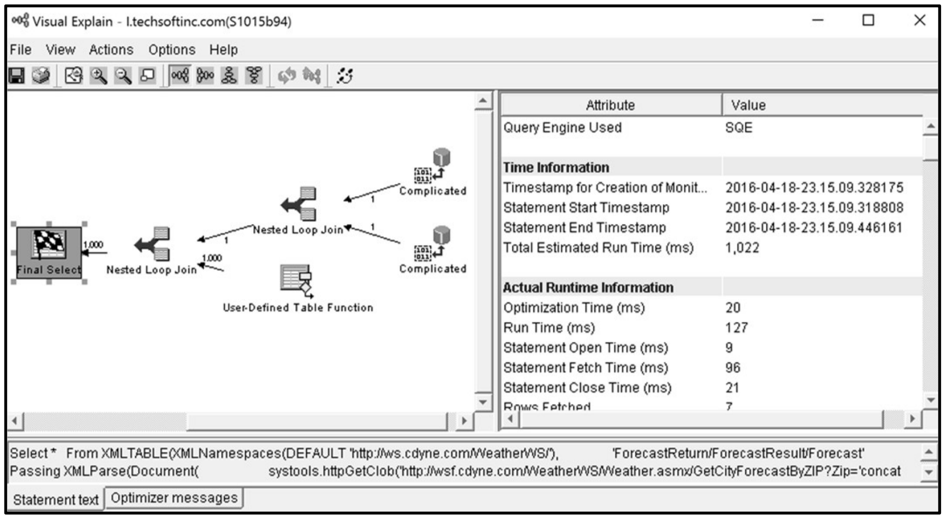
iChime! Roadmap to IBM i SQL Services Mastery with Sven Jansson

Charles G
562 subscribers

https://www.youtube.com/watch?v=DL4zvt45Ksg

48

ACS Visual Explain

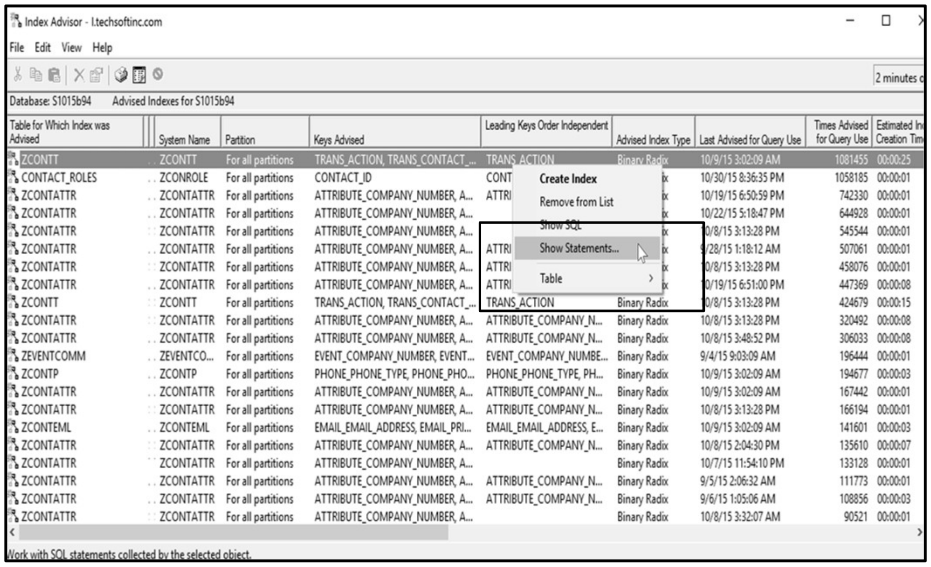


Attribute	Value
Query Engine Used	SQE
Time Information	
Timestamp for Creation of Monit...	2016-04-18-23.15.09.328175
Statement Start Timestamp	2016-04-18-23.15.09.318808
Statement End Timestamp	2016-04-18-23.15.09.446161
Total Estimated Run Time (ms)	1,022
Actual Runtime Information	
Optimization Time (ms)	20
Run Time (ms)	127
Statement Open Time (ms)	9
Statement Fetch Time (ms)	96
Statement Close Time (ms)	21
Rows Fetched	7

```
Select *   From XMLTABLE(XMLNamespaces(DEFAULT 'http://ws.cdyne.com/WeatherWS/'),          'ForecastReturn/ForecastResult/Forecast'
Passing XMLParse(Document(             systools.httpGetClob('http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZIP?Zip='concat
```

49

ACS Index Advisor



Index Advisor - l.techsoftinc.com

Database: S1015b94 Advised Indexes for S1015b94

Table for Which Index was Advised	System Name	Partition	Keys Advised	Leading Keys Order Independent	Advised Index Type	Last Advised for Query Use	Times Advised for Query Use	Estimated Index Creation Time
ZCONTT	ZCONTT	For all partitions	TRANS_ACTION, TRANS_CONTACT_...	TRANS_ACTION	Binary Radix	10/9/15 3:02:09 AM	1081455	00:00:25
CONTACT_ROLES	ZCONROLE	For all partitions	CONTACT_ID	CONT	ix	10/30/15 8:36:35 PM	1058185	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRI	ix	10/19/15 6:50:59 PM	742330	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...		ix	10/22/15 5:18:47 PM	644928	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...		ix	0/8/15 3:13:28 PM	545544	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRI	ix	/28/15 1:18:12 AM	507061	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRI	ix	0/8/15 3:13:28 PM	458076	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRI	ix	0/19/15 6:51:00 PM	447369	00:00:08
ZCONTT	ZCONTT	For all partitions	TRANS_ACTION, TRANS_CONTACT_...	TRANS_ACTION	Binary Radix	0/8/15 3:13:28 PM	424679	00:00:15
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	10/8/15 3:13:28 PM	320492	00:00:08
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	10/8/15 3:48:52 PM	306033	00:00:08
ZEVENTCOMM	ZEVENTCO...	For all partitions	EVENT_COMPANY_NUMBER, EVENT...	EVENT_COMPANY_NUMBE...	Binary Radix	9/4/15 9:03:09 AM	196444	00:00:01
ZCONTP	ZCONTP	For all partitions	PHONE_PHONE_TYPE, PHONE_PHO...	PHONE_PHONE_TYPE, PH...	Binary Radix	10/9/15 3:02:09 AM	194677	00:00:03
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	10/9/15 3:02:09 AM	167442	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	10/8/15 3:13:28 PM	166194	00:00:01
ZCONTEML	ZCONTEML	For all partitions	EMAIL_EMAIL_ADDRESS, EMAIL_PRI...	EMAIL_EMAIL_ADDRESS, E...	Binary Radix	10/9/15 3:02:09 AM	141601	00:00:03
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	10/8/15 2:04:30 PM	135610	00:00:07
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...		Binary Radix	10/7/15 11:54:10 PM	133128	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	9/5/15 2:06:32 AM	111773	00:00:01
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...	ATTRIBUTE_COMPANY_N...	Binary Radix	9/6/15 1:05:06 AM	108856	00:00:03
ZCONTATTR	ZCONTATTR	For all partitions	ATTRIBUTE_COMPANY_NUMBER, A...		Binary Radix	10/8/15 3:32:07 AM	90521	00:00:01

Work with SQL statements collected by the selected object.

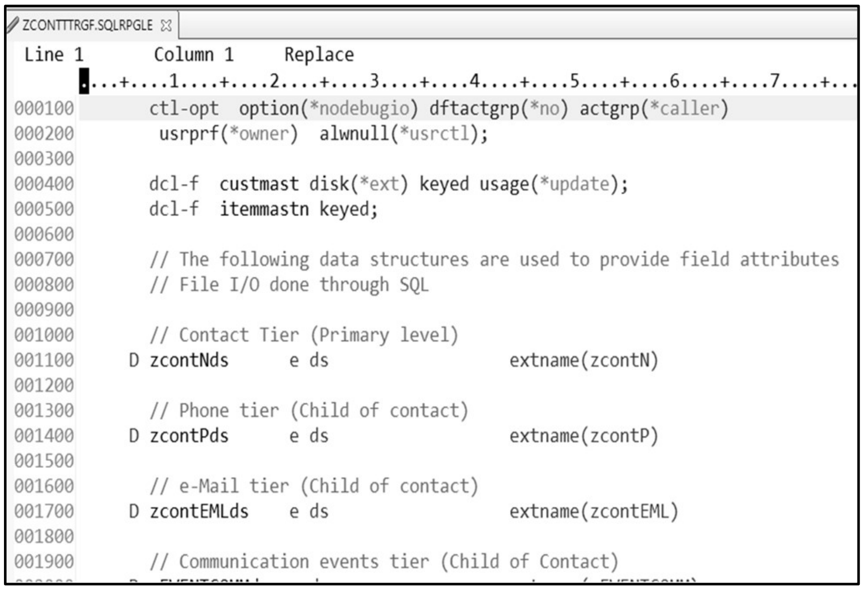
50

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

51

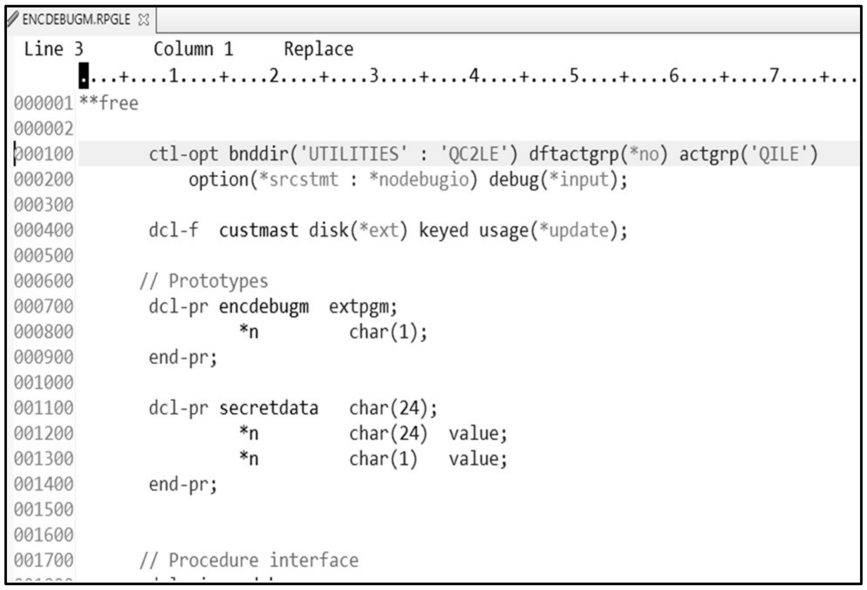
Examples of Field Renaming



```
       ctl-opt  option(*nodebugio) dftactgrp(*no) actgrp(*caller)
        usrprf(*owner)  alwnull(*usrctl);

       dcl-f  custmast disk(*ext) keyed usage(*update);
       dcl-f  itemmastn keyed;

       // The following data structures are used to provide field attributes
       // File I/O done through SQL

       // Contact Tier (Primary level)
     D zcontNds      e ds                 extname(zcontN)

       // Phone tier (Child of contact)
     D zcontPds      e ds                 extname(zcontP)

       // e-Mail tier (Child of contact)
     D zcontEMLds    e ds                 extname(zcontEML)

       // Communication events tier (Child of Contact)
```

52

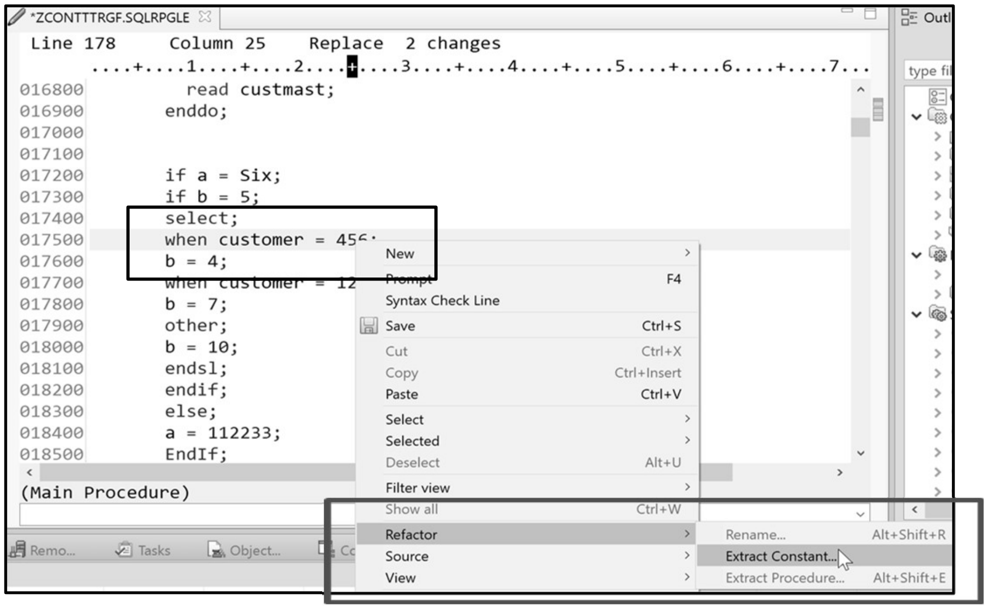
Examples of Extract Constant – Character value



```
**free

       ctl-opt bnddir('UTILITIES' : 'QC2LE') dftactgrp(*no) actgrp('QILE')
           option(*srcstmt : *nodebugio) debug(*input);

       dcl-f  custmast disk(*ext) keyed usage(*update);

      // Prototypes
       dcl-pr encdebugm  extpgm;
                *n         char(1);
       end-pr;

       dcl-pr secretdata   char(24);
                *n         char(24)  value;
                *n         char(1)   value;
       end-pr;


      // Procedure interface
```

53

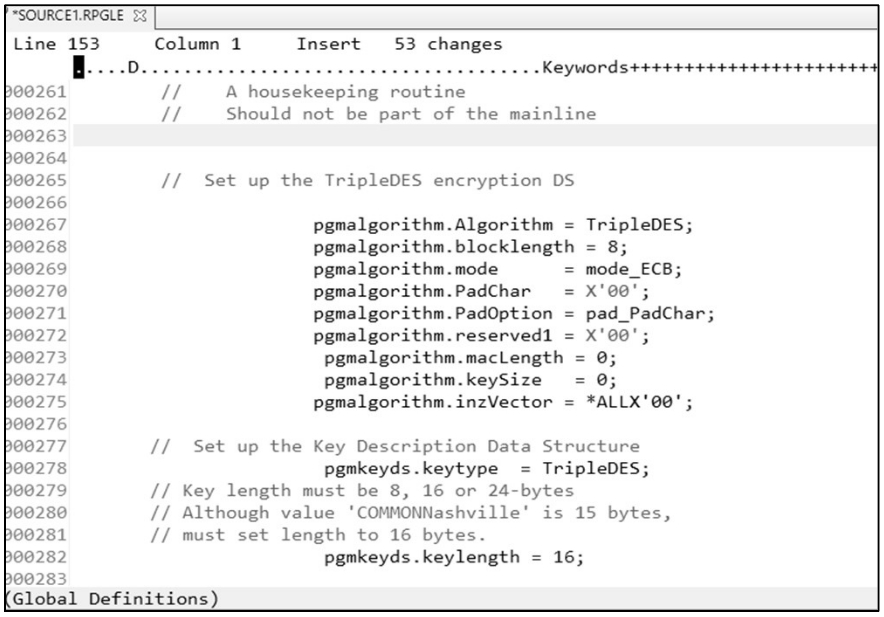
Examples of Extract Constant – Numeric value



```
          read custmast;
        enddo;


        if a = Six;
        if b = 5;
        select;
        when customer = 456;
        b = 4;
        when customer = 12
        b = 7;
        other;
        b = 10;
        endsl;
        endif;
        else;
        a = 112233;
        EndIf;
```

54

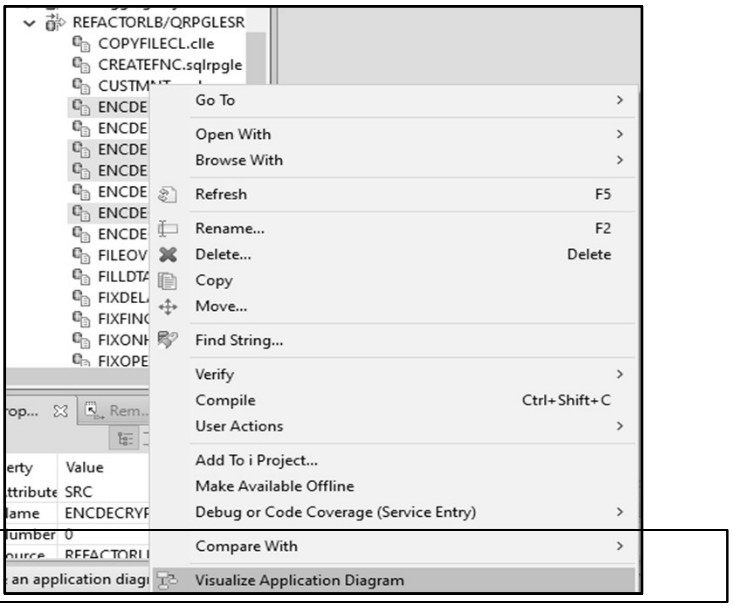
Extract Procedure



```
        //    A housekeeping routine
        //    Should not be part of the mainline


        //  Set up the TripleDES encryption DS

                      pgmalgorithm.Algorithm = TripleDES;
                      pgmalgorithm.blocklength = 8;
                      pgmalgorithm.mode      = mode_ECB;
                      pgmalgorithm.PadChar   = X'00';
                      pgmalgorithm.PadOption = pad_PadChar;
                      pgmalgorithm.reserved1 = X'00';
                       pgmalgorithm.macLength = 0;
                       pgmalgorithm.keySize   = 0;
                      pgmalgorithm.inzVector = *ALLX'00';

       //  Set up the Key Description Data Structure
                       pgmkeyds.keytype  = TripleDES;
       // Key length must be 8, 16 or 24-bytes
       // Although value 'COMMONNashville' is 15 bytes,
       // must set length to 16 bytes.
                       pgmkeyds.keylength = 16;
```

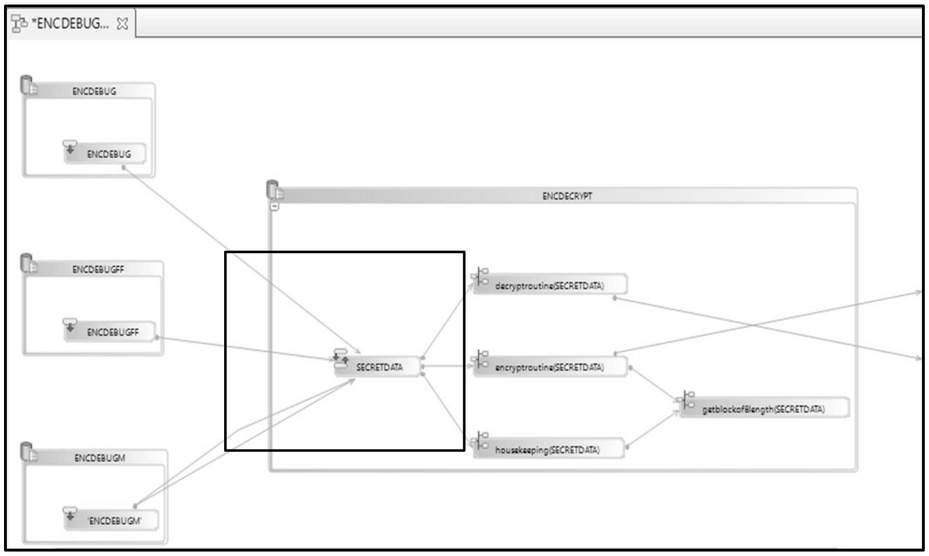
55

Visualize Application Diagram can help identify redundant code



56

Visualize Application Diagram can help identify redundant code



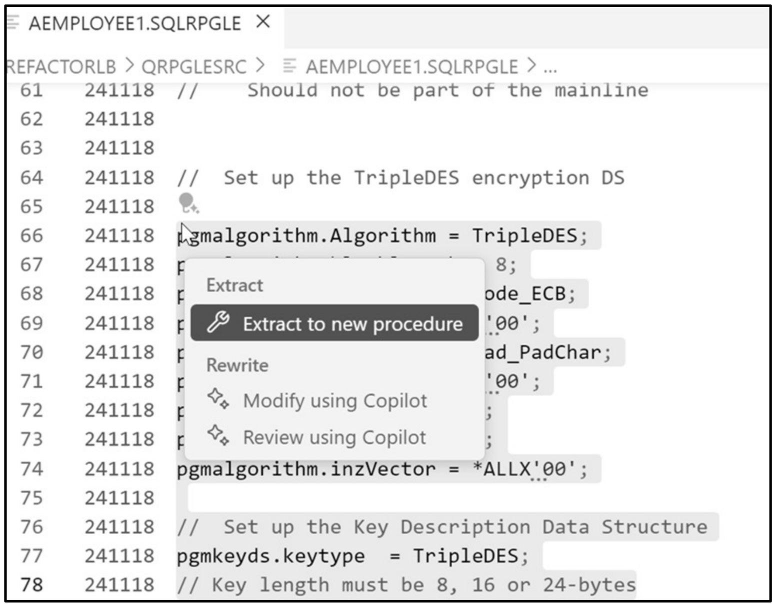
57

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

58

Extract Procedure



```
241118 //    Should not be part of the mainline
241118
241118
241118 //  Set up the TripleDES encryption DS
241118
241118 pgmalgorithm.Algorithm = TripleDES;
241118 p                          8;
241118 p                          ode_ECB;
241118 p                          '00';
241118 p                          ad_PadChar;
241118 p                          '00';
241118 p                          ;
241118 p                          ;
241118 pgmalgorithm.inzVector = *ALLX'00';
241118
241118 //  Set up the Key Description Data Structure
241118 pgmkeyds.keytype  = TripleDES;
241118 // Key length must be 8, 16 or 24-bytes
```

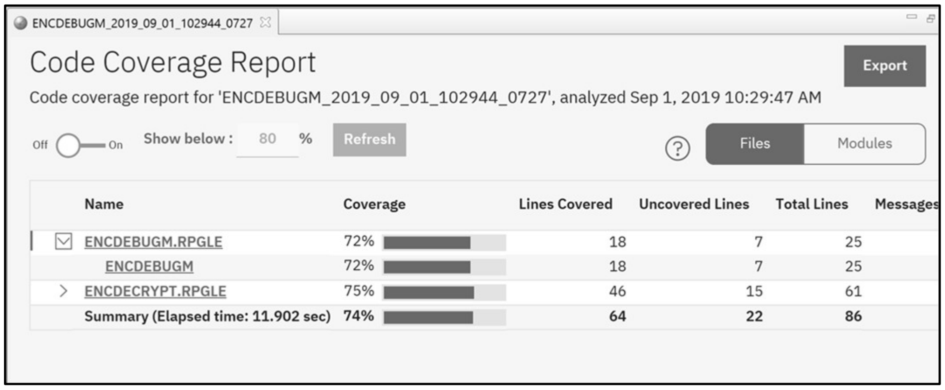
59

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

60

RDi Code Coverage can help identify dead code


ENCDEBUGM_2019_09_01_102944_0727

Code Coverage Report

Code coverage report for 'ENCDEBUGM_2019_09_01_102944_0727', analyzed Sep 1, 2019 10:29:47 AM

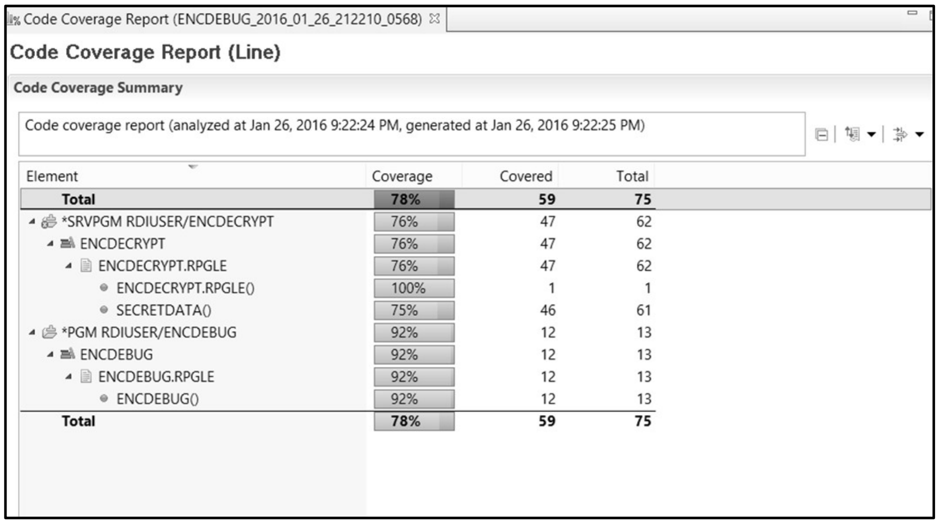
Off On Show below : 80 %

Name	Coverage	Lines Covered	Uncovered Lines	Total Lines	Messages
ENCDEBUGM.RPGLE	72%	18	7	25	
ENCDEBUGM	72%	18	7	25	
ENCDECRYPT.RPGLE	75%	46	15	61	
Summary (Elapsed time: 11.902 sec)	74%	64	22	86	

61

Code coverage report

Can also compare and merge reports!



Code Coverage Report (ENCDEBUG_2016_01_26_212210_0568)

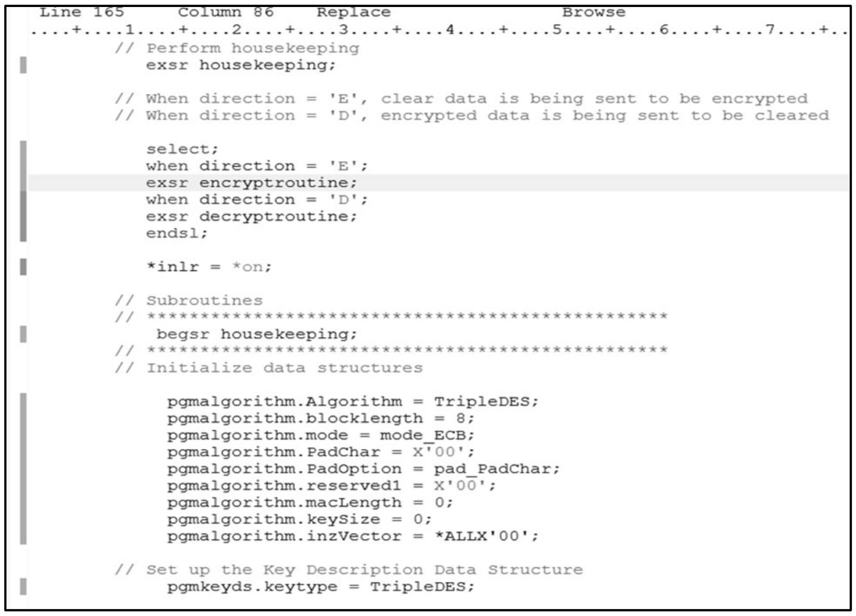
Code Coverage Report (Line)

Code Coverage Summary

Code coverage report (analyzed at Jan 26, 2016 9:22:24 PM, generated at Jan 26, 2016 9:22:25 PM)

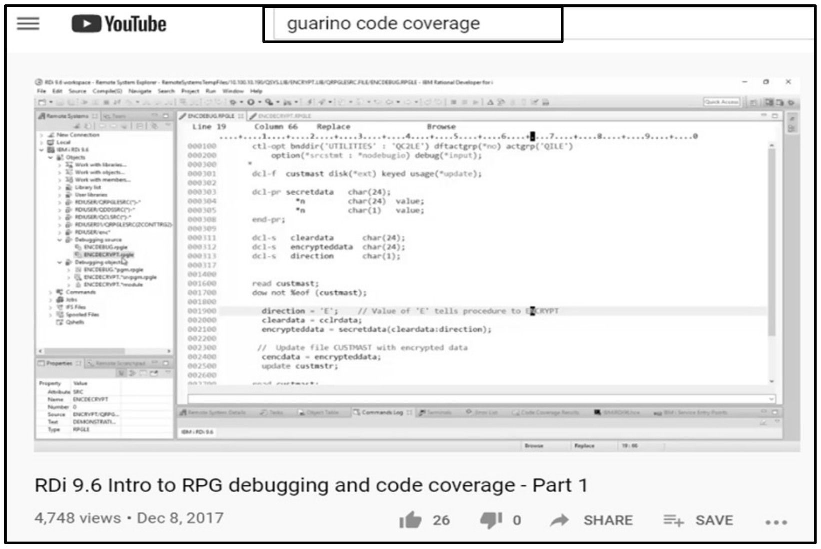
Element	Coverage	Covered	Total
Total	**78%**	**59**	**75**
*SRVPGM RDIUSER/ENCDECRYPT	76%	47	62
ENCDECRYPT	76%	47	62
ENCDECRYPT.RPGLE	76%	47	62
ENCDECRYPT.RPGLE()	100%	1	1
SECRETDATA()	75%	46	61
*PGM RDIUSER/ENCDEBUG	92%	12	13
ENCDEBUG	92%	12	13
ENCDEBUG.RPGLE	92%	12	13
ENCDEBUG()	92%	12	13
Total	**78%**	**59**	**75**

Code coverage report



```
 Line 165     Column 86    Replace              Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+..
        // Perform housekeeping
           exsr housekeeping;

        // When direction = 'E', clear data is being sent to be encrypted
        // When direction = 'D', encrypted data is being sent to be cleared

           select;
           when direction = 'E';
           exsr encryptroutine;
           when direction = 'D';
           exsr decryptroutine;
           endsl;

           *inlr = *on;

        // Subroutines
        // *************************************************
            begsr housekeeping;
        // *************************************************
        // Initialize data structures

             pgmalgorithm.Algorithm = TripleDES;
             pgmalgorithm.blocklength = 8;
             pgmalgorithm.mode = mode_ECB;
             pgmalgorithm.PadChar = X'00';
             pgmalgorithm.PadOption = pad_PadChar;
             pgmalgorithm.reserved1 = X'00';
             pgmalgorithm.macLength = 0;
             pgmalgorithm.keySize = 0;
             pgmalgorithm.inzVector = *ALLX'00';

        // Set up the Key Description Data Structure
             pgmkeyds.keytype = TripleDES;
```

Free videos on how to use Code Coverage



```
Line 19      Column 66    Replace               Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
       ctl-opt bnddir('UTILITIES' : 'QC2LE') dftactgrp(*no) actgrp('QILE')
           option(*srcstmt : *nodebugio) debug(*input);
      *
       dcl-f  custmast disk(*ext) keyed usage(*update);

       dcl-pr secretdata   char(24);
                *n         char(24)  value;
                *n         char(1)   value;
       end-pr;

       dcl-s   cleardata      char(24);
       dcl-s   encrypteddata  char(24);
       dcl-s   direction      char(1);


       read custmast;
       dow not %eof (custmast);

         direction = 'E';    // Value of 'E' tells procedure to ENCRYPT
         cleardata = cclrdata;
         encrypteddata = secretdata(cleardata:direction);

        //  Update file CUSTMAST with encrypted data
         cencdata = encrypteddata;
         update custmstr;
```

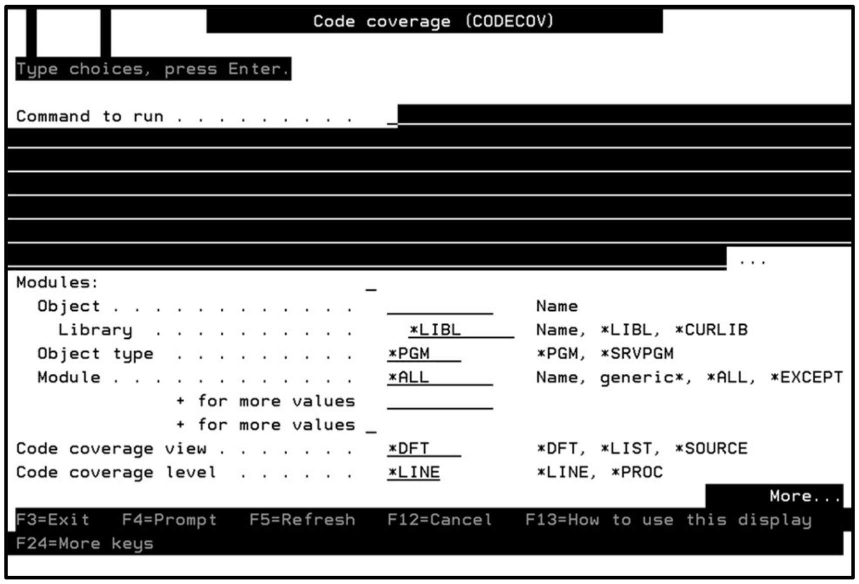
RDi 9.6 Intro to RPG debugging and code coverage - Part 1

4,748 views • Dec 8, 2017 · 26 · 0 · SHARE · SAVE

https://www.youtube.com/watch?v=roesIrpiIAs

64

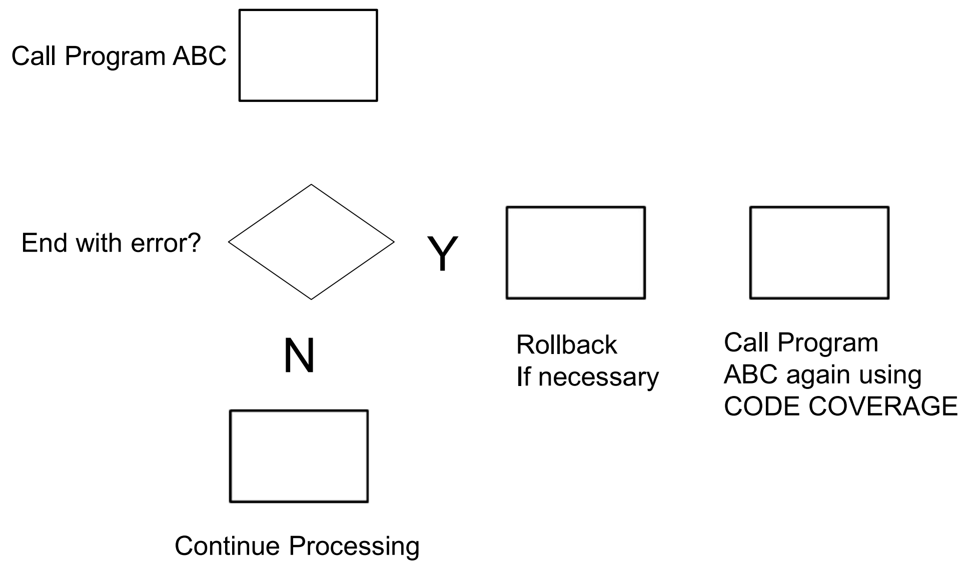
Code Coverage command



```
                         Code coverage (CODECOV)

Type choices, press Enter.

Command to run . . . . . . . . .  _
                                                                 ...
Modules:                                _
  Object . . . . . . . . . . . .                   Name
    Library  . . . . . . . . . .      *LIBL        Name, *LIBL, *CURLIB
  Object type  . . . . . . . . .  *PGM             *PGM, *SRVPGM
  Module . . . . . . . . . . . .  *ALL             Name, generic*, *ALL, *EXCEPT
               + for more values
               + for more values _
Code coverage view . . . . . . .  *DFT             *DFT, *LIST, *SOURCE
Code coverage level  . . . . . .  *LINE            *LINE, *PROC
                                                                  More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

65

Simple idea for using Code Coverage



66

What We'll Cover …

- The Business Case
- What is the End Goal?
- Concerns and Risks
- Some Ways to Get Started
- Refactoring using RDi
- Refactoring using VS Code for IBM i
- Code Coverage
- Wrap up

67

Refactoring With a Splash Of Modern Tooling

Charles Guarino

THANK YOU!!!

68