The DevOps approach:

Develop and test –new capabilities!

Speaker Name and Title

Track 3 - IBM zEnterprise Technology Summit

Develop and test –new capabilities!

Session #3

Learn about recent announcements on our Eclipse based integrated development environment, comprising COBOL, PL/I, C++, assembler and Java development tools for use in batch, CICS, IMS™ and DB2 processing environments.

We'll show how this environment supports the design, creation, deployment and maintenance of traditional transactional applications and modern composite applications running on z/OS operating systems.

We'll highlight the new integrated debugger which provides full edit, compile and debug capabilities out-of-the-box, removing the need for additional products for capabilities such as debug and code coverage.

RDz Integrated Debugger…

A GUI-based, powerful, multi-platform, multi-language debugger:

- ✓ **Platform exploitation:**
  - ✓ Language and subsystem support in RDz v9.0.1
  - ✓ COBOL V5.1, V4, V3.4
    - ✓ Batch, Batch IMS, Batch DB2, CICS 5.1, 4.2, 4.1
  - ✓ Full asynchronous mode:
    - ✓ Thread-level control of multithreaded applications
  - ✓ Automonitor support

RDz Integrated Debugger…

A GUI-based, powerful, multi-platform, multi-language debugger:

- ✓ **Interactive Code coverage:**
  - ✓ Measures and reports on the test coverage of an application
- ✓ **Host-offload architecture:**
  - ✓ Remote debugger with only a small footprint on the mainframe:
    - Leverages workstation CPUs enabling faster processing of debug information
    - Enables scalability and reliability
  - ✓ Debugger client is supported on Windows and Linux
- ✓ **Simple and Secure Connections:**
  - ✓ Single client can handle multiple debug sessions on multiple hosts or an application the spans multiple systems

RDz Integrated Debugger…

Support for numerous views enabling problem determination:

- ✓ **Debug View**
  - ✓ For managing program debugging
- ✓ **Breakpoints View**
  - ✓ for setting and working with the following breakpoint types:
    - **Line breakpoints**: triggered when the line they are set on is about to be executed
    - **Entry breakpoints**: triggered when the entry points they apply to are entered
    - **Address breakpoints**: triggered before the disassembly instruction at a particular address is executed
    - **Load breakpoints**: triggered when a DLL or object module is loaded
    - **Conditional breakpoints**: triggered conditionally depending on optional breakpoint parameters used to control the behavior of these breakpoints
    - **Event breakpoints**: triggered when the debugger recognizes an exception thrown by the application

RDz Integrated Debugger…

Support for numerous views enabling problem determination:

- ✓ **Monitors View**
  - ✓ for working with monitored variables, expressions, and/or registers
- ✓ **Memory View**
  - ✓ for viewing and mapping memory used by your application
- ✓ **Modules View**
  - ✓ for viewing the list of modules loaded while running your program
  - ✓ for navigating to the individual compile units and source files in your application to view function entry points and set breakpoints on them
- ✓ **Variables View**
  - ✓ for viewing the list of variables in your application and editing those variables
- ✓ **Registers View**
  - ✓ For viewing the registers in your program

RDz Integrated Debugger…

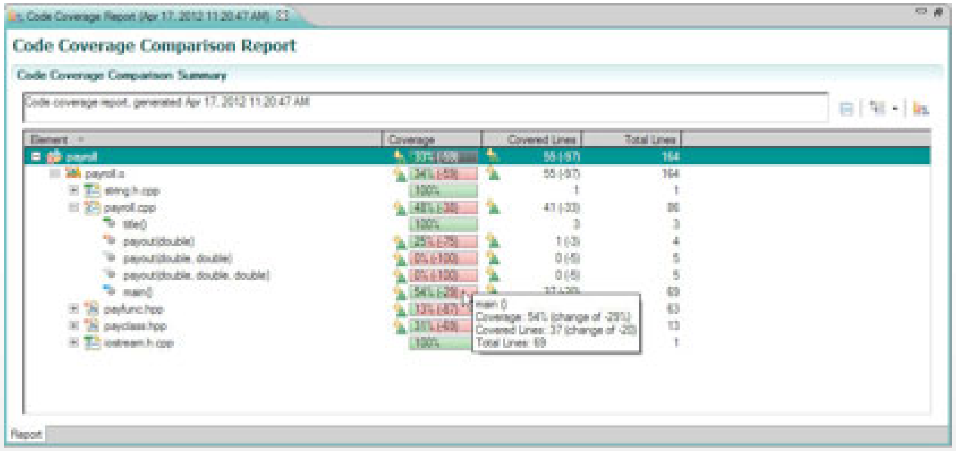
Support for numerous views enabling problem determination:

- ✓ **Debug Console**
  - ✓ for issuing commands to the debug engine, viewing output from the engine, and seeing the results of the commands you have issued
- ✓ **Consoles View**
  - ✓ For displaying the screen output of your program

RDz Interactive Code Coverage

You can generate code coverage statistics from within the workbench:

- ✓ RDz interactive code coverage determines the extent and effectiveness of test coverage
  - ✓ If you can debug it… you can analyze code coverage
    - Does not require instrumentation of the executable
- ✓ Measures and captures code coverage statistics for an application
  - ✓ Acceptance criteria are customizable
- ✓ Reports on line-level or function-level coverage
  - ✓ Workbench and HTML report formats are supported
    - Supports comparisons to previous results (delta markers indicate differences)
  - ✓ Code coverage is also conveyed through customizable decorations in the editor



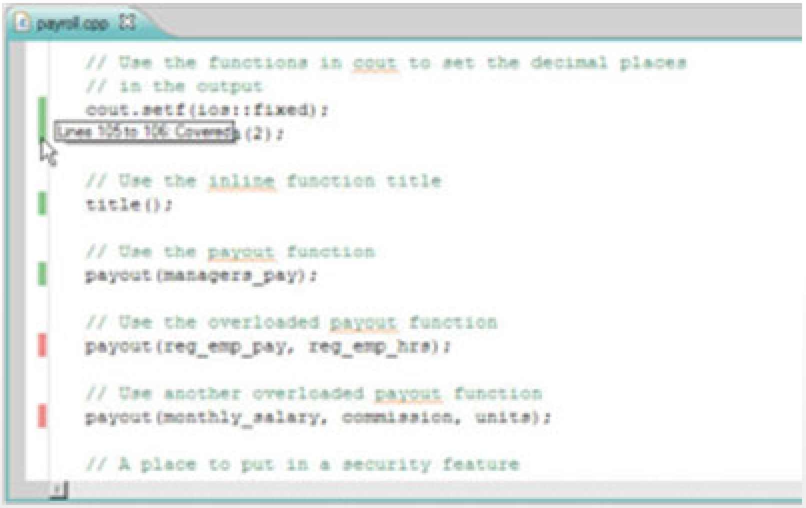
Code Coverage Comparison Report

Code Coverage Comparison Summary

Code coverage report, generated Apr 17, 2012 11:20:47 AM

Element	Coverage	Covered Lines	Total Lines
payroll	33% (-59)	55 (-97)	164
payroll.a	34% (-59)	55 (-97)	164
string.h.cpp	100%	1	1
payroll.cpp	48% (-38)	41 (-33)	86
title()	100%	3	3
payout(double)	25% (-75)	1 (-3)	4
payout(double, double)	0% (-100)	0 (-5)	5
payout(double, double, double)	0% (-100)	0 (-5)	5
main()	54% (-29)	37 (-20)	69
payfunc.hpp	13% (-87)		63
payclass.hpp	31% (-69)		13
iostream.h.cpp	100%		1

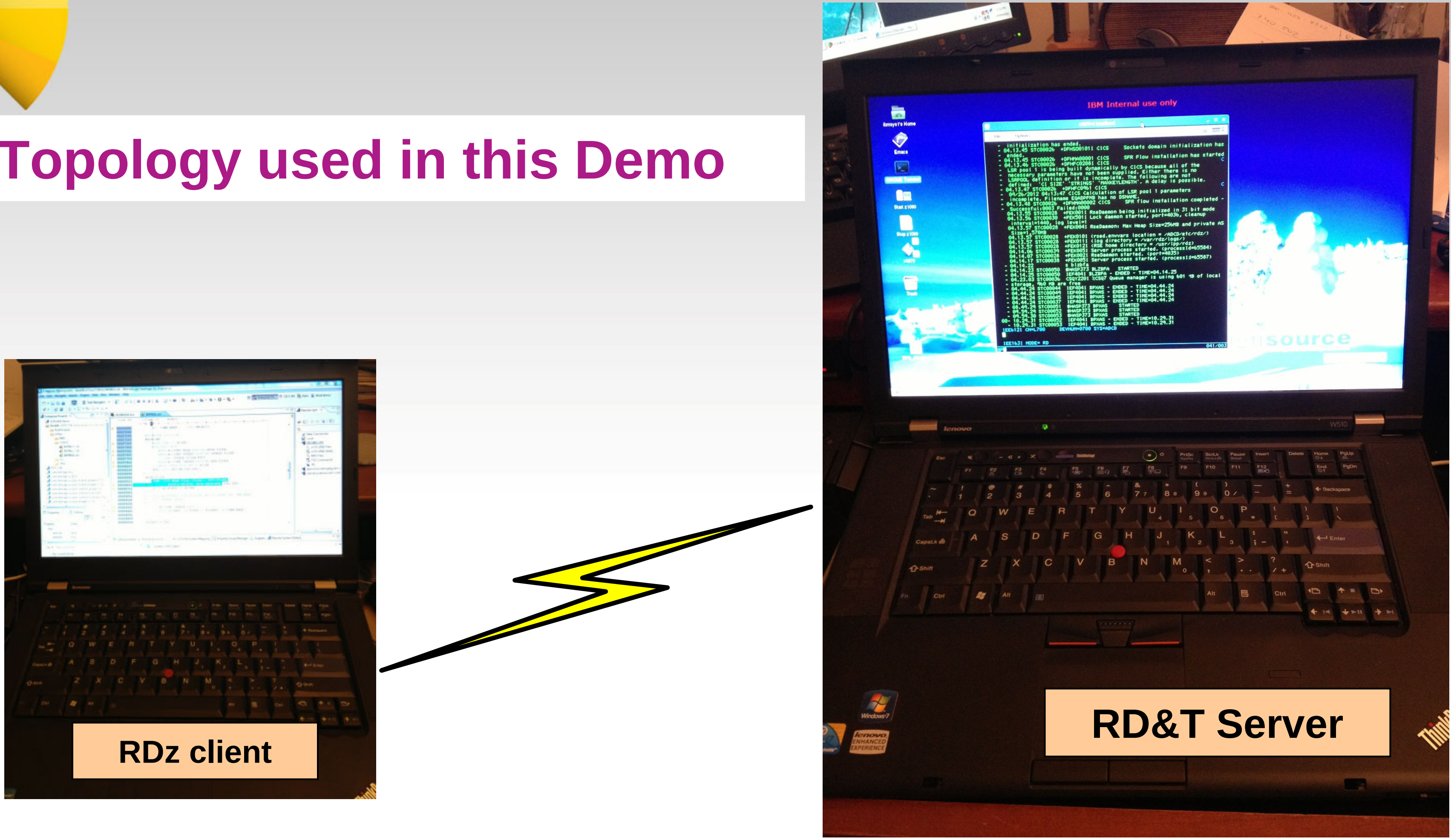
main ()
Coverage: 54% (change of -29%)
Covered Lines: 37 (change of -20)
Total Lines: 69



```
// Use the functions in cout to set the decimal places
// in the output
cout.setf(ios::fixed);
Lines 105 to 106: Covered(2);

// Use the inline function title
title();

// Use the payout function
payout(managers_pay);

// Use the overloaded payout function
payout(reg_emp_pay, reg_emp_hrs);

// Use another overloaded payout function
payout(monthly_salary, commission, units);

// A place to put in a security feature
```

DEMO

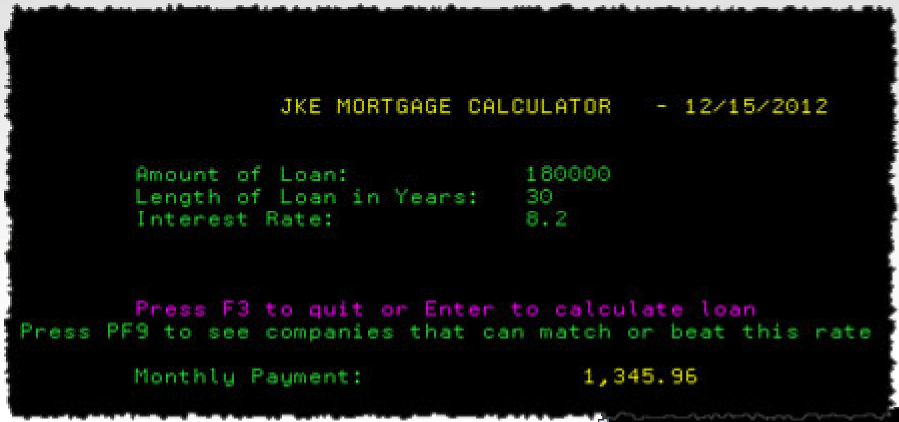
Topology used in this Demo



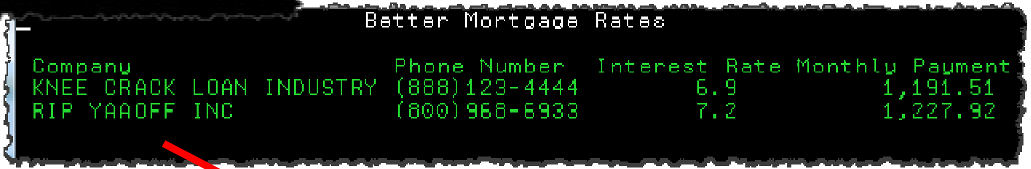
RDz client

RD&T Server

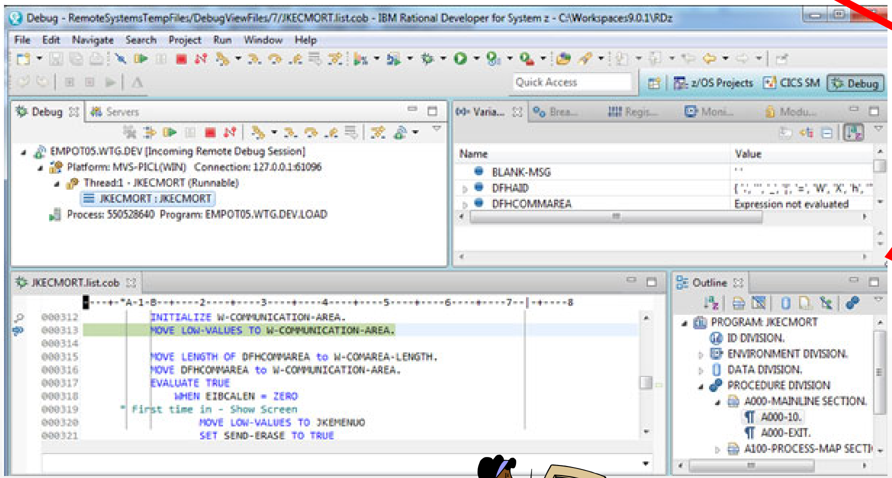
Scenario : Debugging JKE Mortgage Calculator



```
           JKE MORTGAGE CALCULATOR   - 12/15/2012

      Amount of Loan:             180000
      Length of Loan in Years:    30
      Interest Rate:              8.2


      Press F3 to quit or Enter to calculate loan
  Press PF9 to see companies that can match or beat this rate

      Monthly Payment:                1,345.96
```



```
                         Better Mortgage Rates

 Company                         Phone Number   Interest Rate Monthly Payment
 KNEE CRACK LOAN INDUSTRY (888)123-4444          6.9           1,191.51
 RIP YAAOFF INC           (800)968-6933          7.2           1,227.92
```



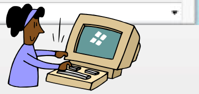
Debug - RemoteSystemsTempFiles/DebugViewFiles/7/JKECMORT.list.cob - IBM Rational Developer for System z - C:\Workspaces9.0.1\RDz

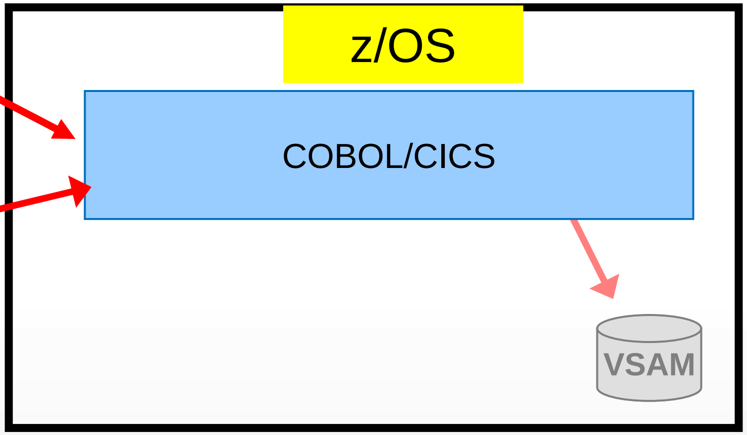
```
EMPOT05.WTG.DEV [Incoming Remote Debug Session]
  Platform: MVS-PICL(WIN)  Connection: 127.0.0.1:61096
    Thread:1 - JKECMORT (Runnable)
      JKECMORT : JKECMORT
  Process: 550528640  Program: EMPOT05.WTG.DEV.LOAD
```

Name	Value
BLANK-MSG	' '
DFHAID	[',', '"', '_', '/', '=', 'W', 'X', 'h', '"
DFHCOMMAREA	Expression not evaluated

```
       ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--|-+----8
000312           INITIALIZE W-COMMUNICATION-AREA.
000313           MOVE LOW-VALUES TO W-COMMUNICATION-AREA.
000314
000315           MOVE LENGTH OF DFHCOMMAREA to W-COMAREA-LENGTH.
000316           MOVE DFHCOMMAREA to W-COMMUNICATION-AREA.
000317           EVALUATE TRUE
000318               WHEN EIBCALEN = ZERO
000319      * First time in - Show Screen
000320                   MOVE LOW-VALUES TO JKEMENUO
000321                   SET SEND-ERASE TO TRUE
```

```
PROGRAM: JKECMORT
  ID DIVISION.
  ENVIRONMENT DIVISION.
  DATA DIVISION.
  PROCEDURE DIVISION
    A000-MAINLINE SECTION.
      A000-10.
      A000-EXIT.
    A100-PROCESS-MAP SECTI
```





•Existing COBOL/CICS/BMS application

Objective: Debug Mortgage CICS transaction using Integrated debug

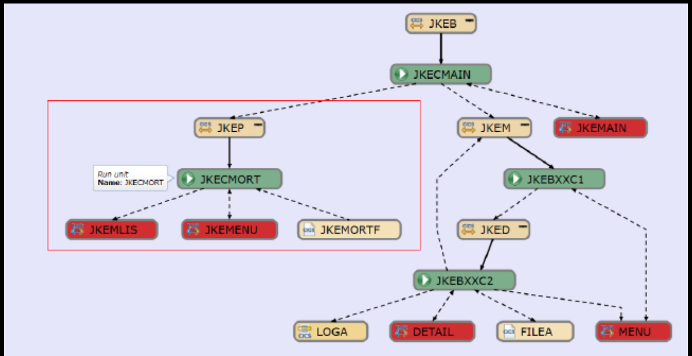
1. Create and activate a CICS Debug profile

2. Show Program Control Flow

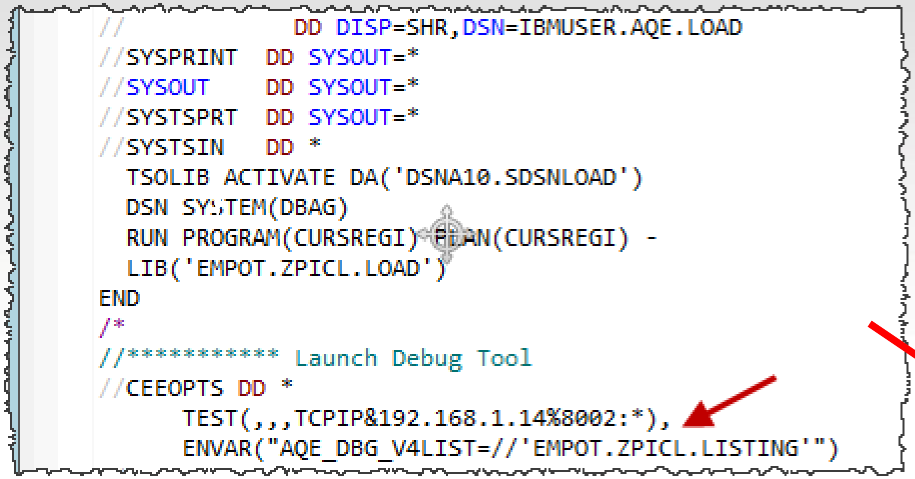
3. Monitor Expression

4. Create Breakpoints, skip all breakpoints

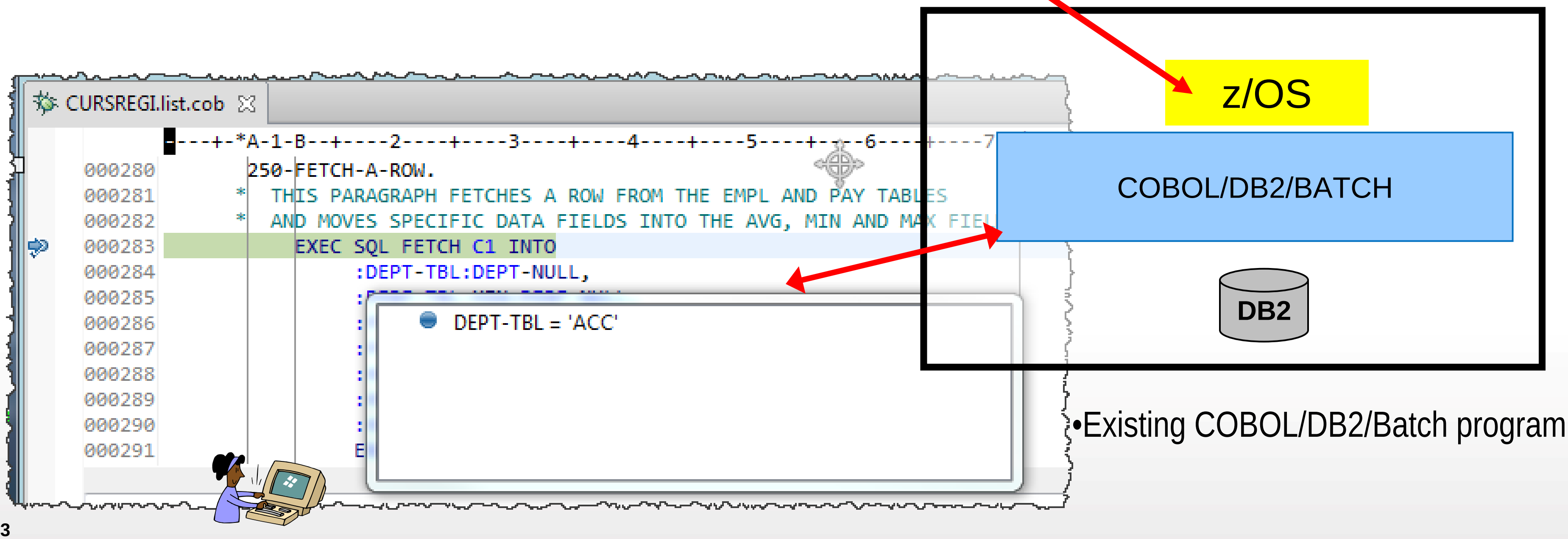
5. Debug and show the COBOL variables, open source code declaration and occurrences



Scenario : Debugging COBOL/DB2 Batch program



```
//                DD DISP=SHR,DSN=IBMUSER.AQE.LOAD
//SYSPRINT  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
  TSOLIB ACTIVATE DA('DSNA10.SDSNLOAD')
  DSN SYSTEM(DBAG)
  RUN PROGRAM(CURSREGI) PLAN(CURSREGI) -
  LIB('EMPOT.ZPICL.LOAD')
END
/*
//*********** Launch Debug Tool
//CEEOPTS DD *
      TEST(,,,TCPIP&192.168.1.14%8002:*),
      ENVAR("AQE_DBG_V4LIST=//'EMPOT.ZPICL.LISTING'")
```



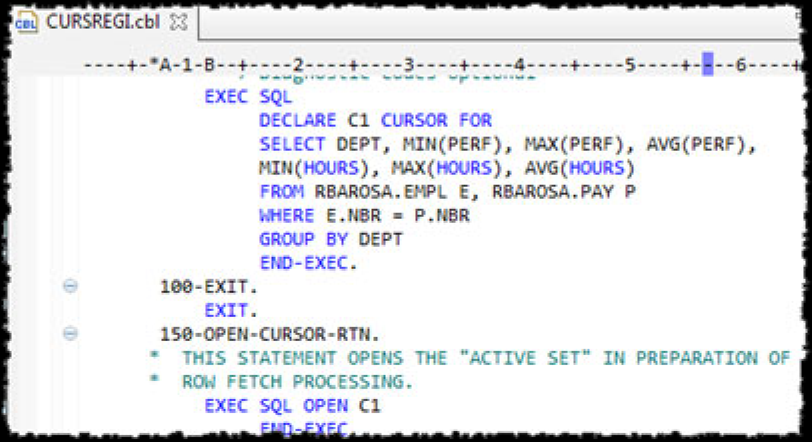
```
----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7
000280        250-FETCH-A-ROW.
000281       *   THIS PARAGRAPH FETCHES A ROW FROM THE EMPL AND PAY TABLES
000282       *   AND MOVES SPECIFIC DATA FIELDS INTO THE AVG, MIN AND MAX FIEL
000283               EXEC SQL FETCH C1 INTO
000284                   :DEPT-TBL:DEPT-NULL,
000285                   :
000286                   :
000287                   :
000288                   :
000289                   :
000290                   :
000291                   E
```

DEPT-TBL = 'ACC'

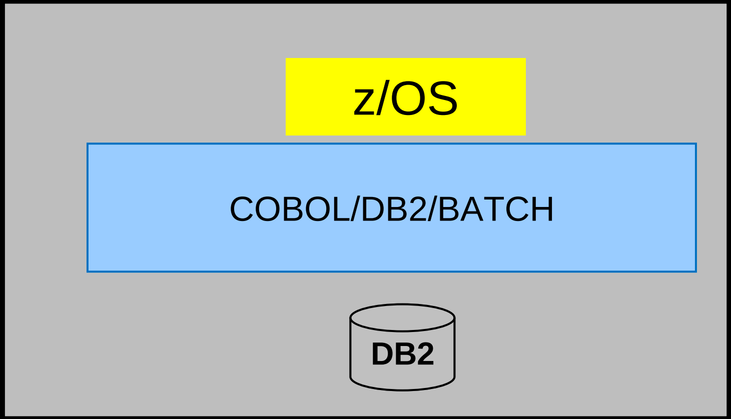
- Existing COBOL/DB2/Batch program

Objective: Debugging COBOL/DB2 Batch program

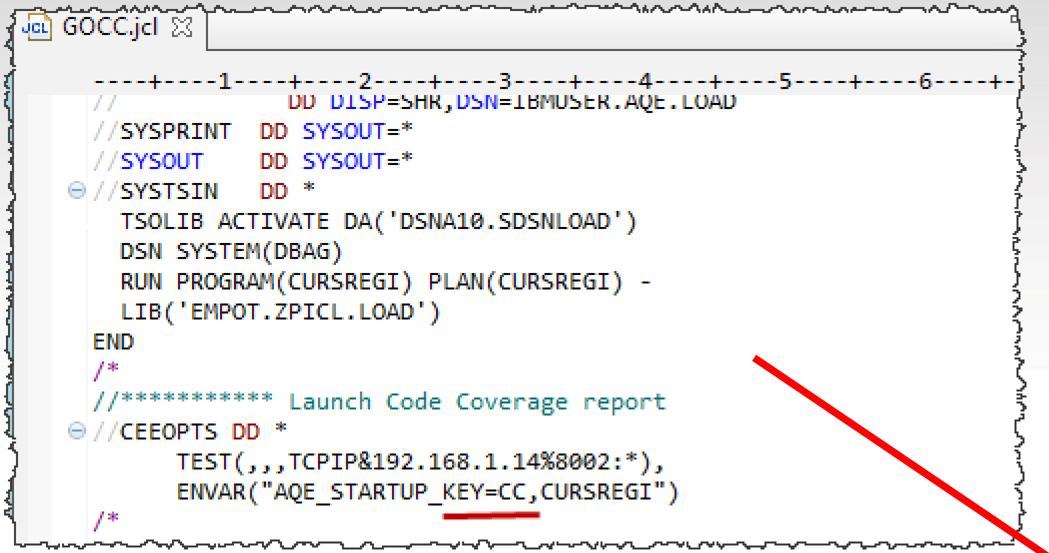
1. Submit a JCL to execute a COBOL/ BATCH/DB2
2. Show Program Control Flow
3. Open source code declaration and occurrences
4. Conditional Breakpoint
5. Debug and show the COBOL variables,



```
----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+
           EXEC SQL
               DECLARE C1 CURSOR FOR
               SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PERF),
               MIN(HOURS), MAX(HOURS), AVG(HOURS)
               FROM RBAROSA.EMPL E, RBAROSA.PAY P
               WHERE E.NBR = P.NBR
               GROUP BY DEPT
               END-EXEC.
       100-EXIT.
           EXIT.
       150-OPEN-CURSOR-RTN.
      *  THIS STATEMENT OPENS THE "ACTIVE SET" IN PREPARATION OF
      *  ROW FETCH PROCESSING.
           EXEC SQL OPEN C1
               END-EXEC
```

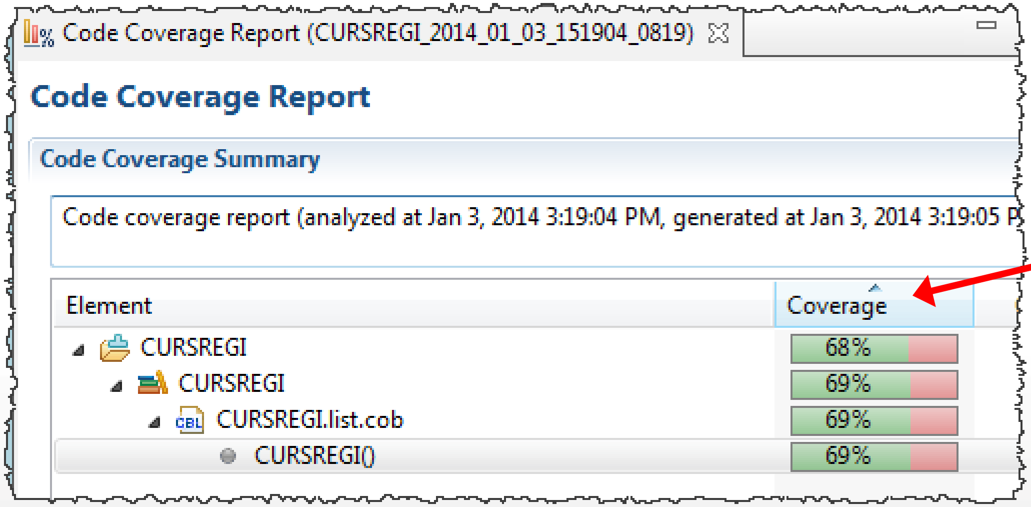


Scenario : Code Coverage using a COBOL Batch program



GOCC.jcl

```
----+----1----+----2----+----3----+----4----+----5----+----6----+-
//            DD DISP=SHR,DSN=IBMUSER.AQE.LOAD
//SYSPRINT  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//SYSTSIN   DD *
  TSOLIB ACTIVATE DA('DSNA10.SDSNLOAD')
  DSN SYSTEM(DBAG)
  RUN PROGRAM(CURSREGI) PLAN(CURSREGI) -
  LIB('EMPOT.ZPICL.LOAD')
END
/*
//*********** Launch Code Coverage report
//CEEOPTS DD *
      TEST(,,,TCPIP&192.168.1.14%8002:*),
      ENVAR("AQE_STARTUP_KEY=CC,CURSREGI")
/*
```



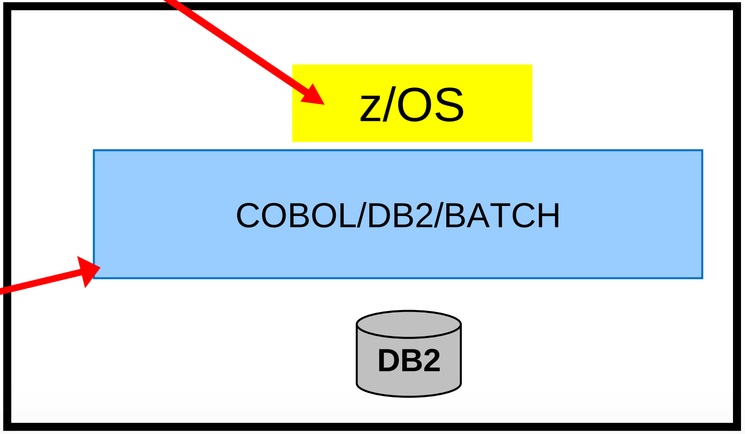
Code Coverage Report (CURSREGI_2014_01_03_151904_0819)

Code Coverage Report

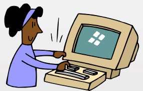
Code Coverage Summary

Code coverage report (analyzed at Jan 3, 2014 3:19:04 PM, generated at Jan 3, 2014 3:19:05 P

Element	Coverage
CURSREGI	68%
CURSREGI	69%
CURSREGI.list.cob	69%
CURSREGI()	69%



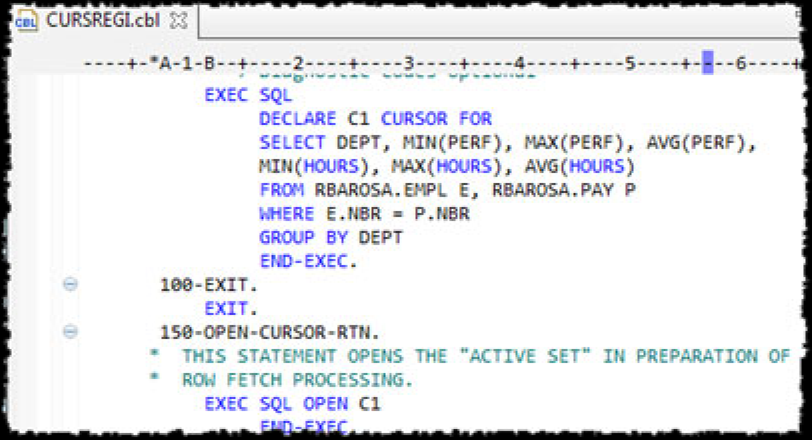
- Existing COBOL/DB2/Batch program



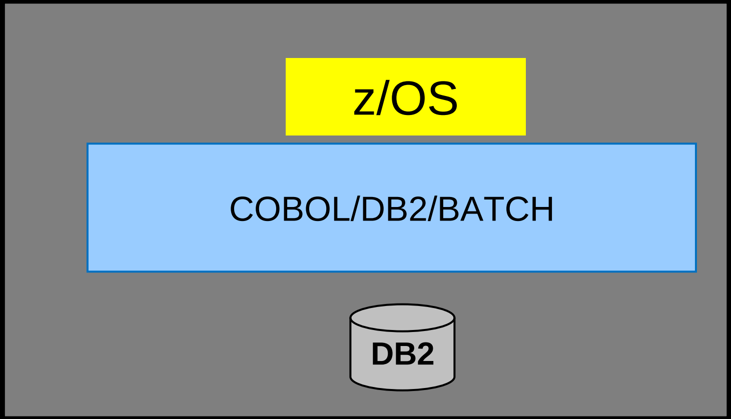
Objective: Code Coverage using a COBOL batch application

1. Submit a JCL to compile, link and execute a COBOL program using the Code Coverage capability

2. Verify the results and reports



```
----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+
           EXEC SQL
                DECLARE C1 CURSOR FOR
                SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PERF),
                MIN(HOURS), MAX(HOURS), AVG(HOURS)
                FROM RBAROSA.EMPL E, RBAROSA.PAY P
                WHERE E.NBR = P.NBR
                GROUP BY DEPT
                END-EXEC.
       100-EXIT.
           EXIT.
       150-OPEN-CURSOR-RTN.
      *  THIS STATEMENT OPENS THE "ACTIVE SET" IN PREPARATION OF
      *  ROW FETCH PROCESSING.
           EXEC SQL OPEN C1
                END-EXEC
```



- BACKUP

Positioning RDz 9.0.1 and IBM PD Tools

RDz 9.0.1, with its new Integrated Debugger, was released on December 11, 2013

In its first release, the RDz Integrated Debugger will support:

- COBOL (V3.4, V4, V5) batch, batch IMS, batch DB2, and CICS (V4.1, V4.2 and V5.1)
- Interactive Code Coverage
- RDz roadmap: PL/I, HL ASM, C/C++, IMS TM, DB2 SPs
- 3270 debug support will remain the purview of IBM Debug Tool

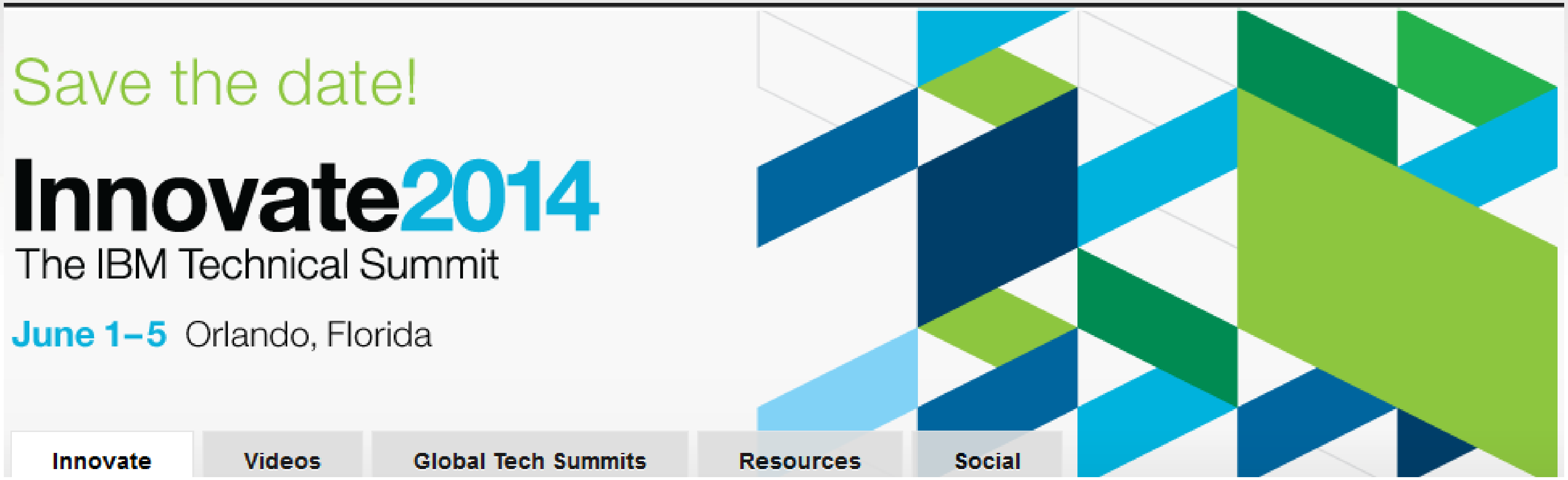
RDz, with its best in class application development features, together with the IBM PD Tools, provides a very compelling ROI for our IBM System z customers

Summary and Next Steps

Next Steps – to help determine your best entry point

- Podcasts
  - ISDz - Construct high quality applications faster with the IBM Integrated Solution for System z Development http://www-01.ibm.com/software/rational/podcasts/2013/#281
  - CIz – CICS Virtualization Testing - coming soon
  - RDz - What's hot for mainframe, IBM i, and distributed applications development? http://www-01.ibm.com/software/rational/podcasts/2013/#286
  - IDEs for DevOps - Application Development in a DevOps Enterprise http://www-01.ibm.com/software/rational/podcasts/2013/#288
- Papers
  - DevOps on z whitepaper - https://www.ibm.com/services/forms/signup.do?source=swg-NA_LMI&S_PKG=ov20523
  - ISDz Solution Brief http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=PM&subtype=SP&appname=SWGE_RA_RA_USEN&htmlfid=RAS14078USEN&attachment=RAS14078USEN.PDF
  - CIz Solution Brief https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-103GS07W
- Screen Cast – Enterprise COBOL V5R1 - http://redmonk.com/jgovernor/2013/09/12/a-moderated-screencast-enterprise-cobol-for-zos-5-1
- Trial – COBOL V5R1 - http://www-03.ibm.com/software/products/en/enterprise-cobol-developer-trial-for-zos

Innovate 2014



Save the date!

Innovate2014

The IBM Technical Summit

June 1–5 Orlando, Florida

http://www-01.ibm.com/software/rational/innovate/

Questions

Thank You