Cognitive Systems

Baking Quality Assurance into your Application

Speaker name – edmund.reinhardt@ca.ibm.com
Rational Developer for i – IBM architect

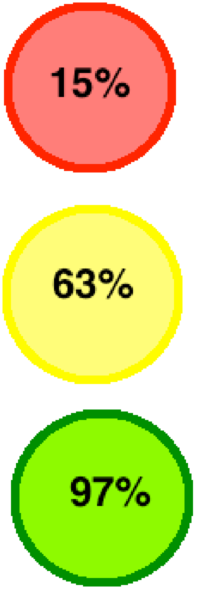
™

Agenda

- What is code coverage
- What was already delivered
- What’s new
- Demo

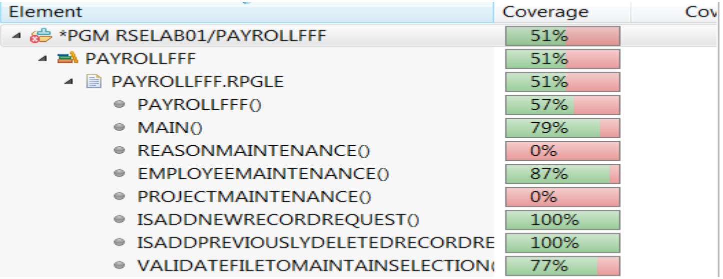
RD i Code Coverage

- Do you know if your testcases actually test what you need them too ?
- Do you have Dead code in your applications ?
- What is your QA story ?

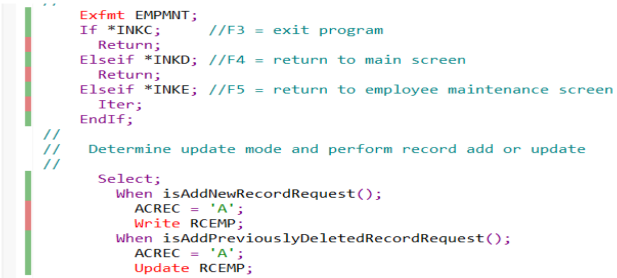


RDi expands into new territory – Quality Assurance

- RDi Line level Code Coverage Analysis Capability



Element	Coverage	Cov
*PGM RSELAB01/PAYROLLFFF	51%	
PAYROLLFFF	51%	
PAYROLLFFF.RPGLE	51%	
PAYROLLFFF()	57%	
MAIN()	79%	
REASONMAINTENANCE()	0%	
EMPLOYEEMAINTENANCE()	87%	
PROJECTMAINTENANCE()	0%	
ISADDNEWRECORDREQUEST()	100%	
ISADDPREVIOUSLYDELETEDRECORDRE	100%	
VALIDATEFILETOMAINTAINSELECTION(	77%	

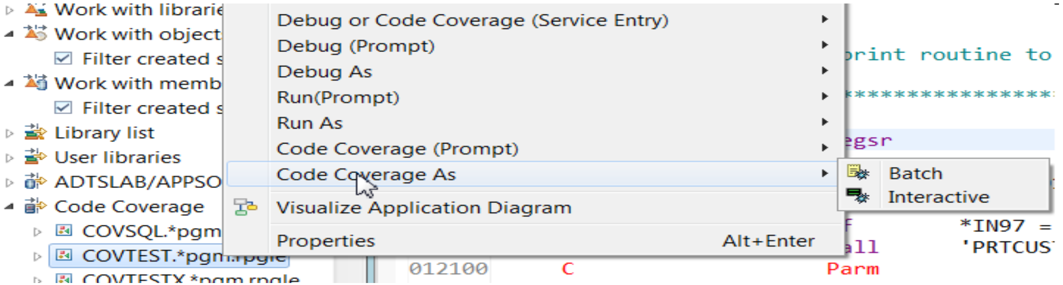


```
      Exfmt EMPMNT;
      If *INKC;      //F3 = exit program
        Return;
      Elseif *INKD; //F4 = return to main screen
        Return;
      Elseif *INKE; //F5 = return to employee maintenance screen
        Iter;
      EndIf;
//
//    Determine update mode and perform record add or update
//
        Select;
          When isAddNewRecordRequest();
            ACREC = 'A';
            Write RCEMP;
          When isAddPreviouslyDeletedRecordRequest();
            ACREC = 'A';
            Update RCEMP;
```

- You can see exactly which lines were covered and not.
- These results can be accumulated see the coverage of the total suite of tests as well as speeding up the analysis of subsequent test runs.
- Can determine effectiveness of tests, and guide improvements
- Can help locate dead code

Launching Code Coverage

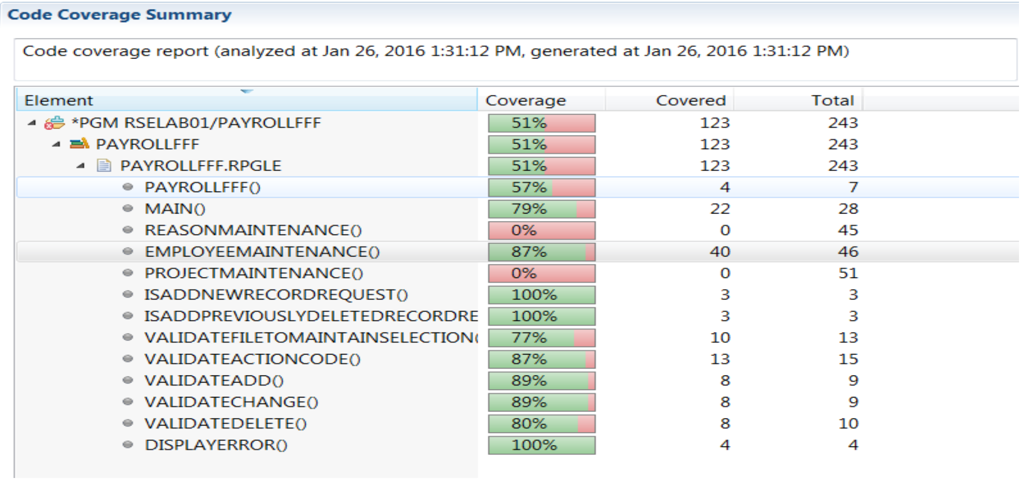
- Code coverage can be launched on any program or service program that can be debugged – independent of language



- Batch program analysis supported as of RDi 9.1
- Interactive and SEP supported added in RDi 9.1.1 (December 2014)

Code Coverage Report

- After running code coverage, a report is shown as an editor.
- You can drill down through programs, modules and procedures and see the coverage statistics for each



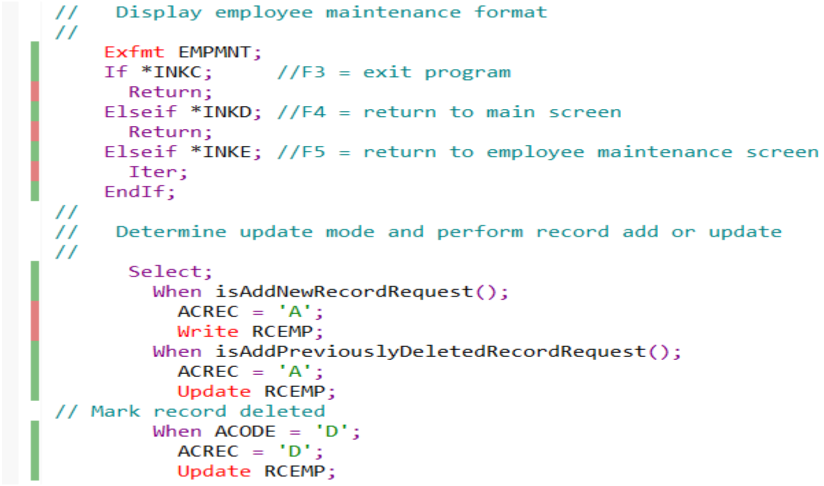
Code Coverage Summary

Code coverage report (analyzed at Jan 26, 2016 1:31:12 PM, generated at Jan 26, 2016 1:31:12 PM)

Element	Coverage	Covered	Total
*PGM RSELAB01/PAYROLLFFF	51%	123	243
PAYROLLFFF	51%	123	243
PAYROLLFFF.RPGLE	51%	123	243
PAYROLLFFF()	57%	4	7
MAIN()	79%	22	28
REASONMAINTENANCE()	0%	0	45
EMPLOYEEMAINTENANCE()	87%	40	46
PROJECTMAINTENANCE()	0%	0	51
ISADDNEWRECORDREQUEST()	100%	3	3
ISADDPREVIOUSLYDELETEDRECORDRE	100%	3	3
VALIDATEFILETOMAINTAINSELECTION(	77%	10	13
VALIDATEACTIONCODE()	87%	13	15
VALIDATEADD()	89%	8	9
VALIDATECHANGE()	89%	8	9
VALIDATEDELETE()	80%	8	10
DISPLAYERROR()	100%	4	4

Coverage annotated in the editor

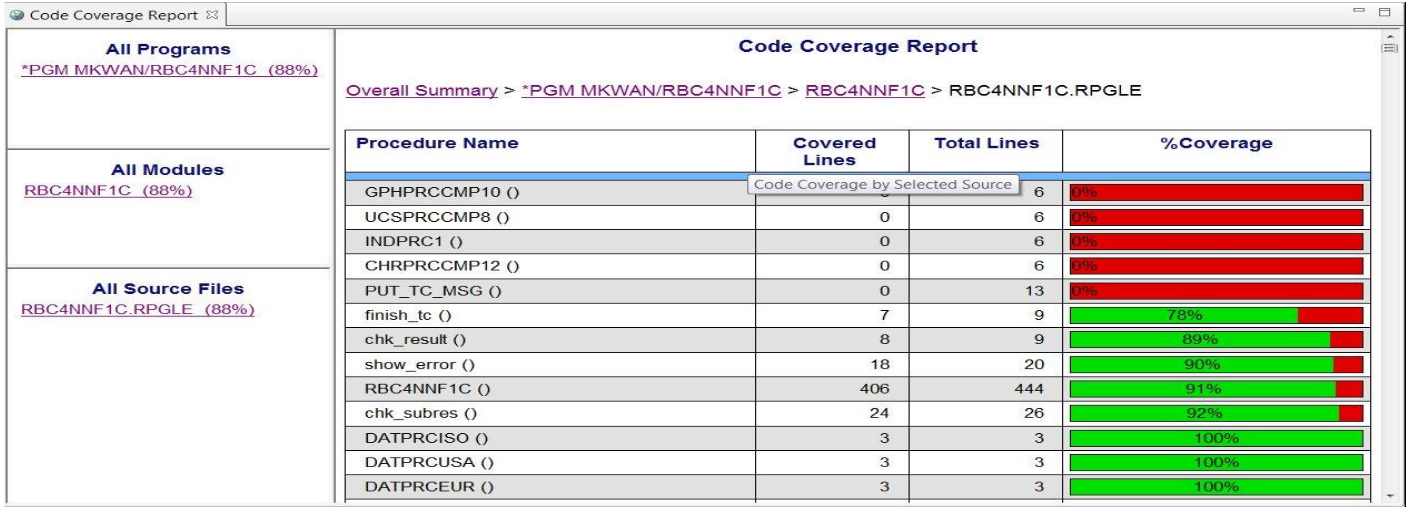
- Drilling down from the report, the editor will be opened on the related member with green and red annotations showing which lines were covered.



```
//   Display employee maintenance format
//
    Exfmt EMPMNT;
    If *INKC;     //F3 = exit program
      Return;
    Elseif *INKD; //F4 = return to main screen
      Return;
    Elseif *INKE; //F5 = return to employee maintenance screen
      Iter;
    EndIf;
//
//   Determine update mode and perform record add or update
//
      Select;
        When isAddNewRecordRequest();
          ACREC = 'A';
          Write RCEMP;
        When isAddPreviouslyDeletedRecordRequest();
          ACREC = 'A';
          Update RCEMP;
// Mark record deleted
        When ACODE = 'D';
          ACREC = 'D';
          Update RCEMP;
```

HTML Report

- The results can shown via HTML or PDF reports so they are available for those who are not in the Eclipse client, i.e. quality assurance, management etc.



All Programs
*PGM MKWAN/RBC4NNF1C (88%)

All Modules
RBC4NNF1C (88%)

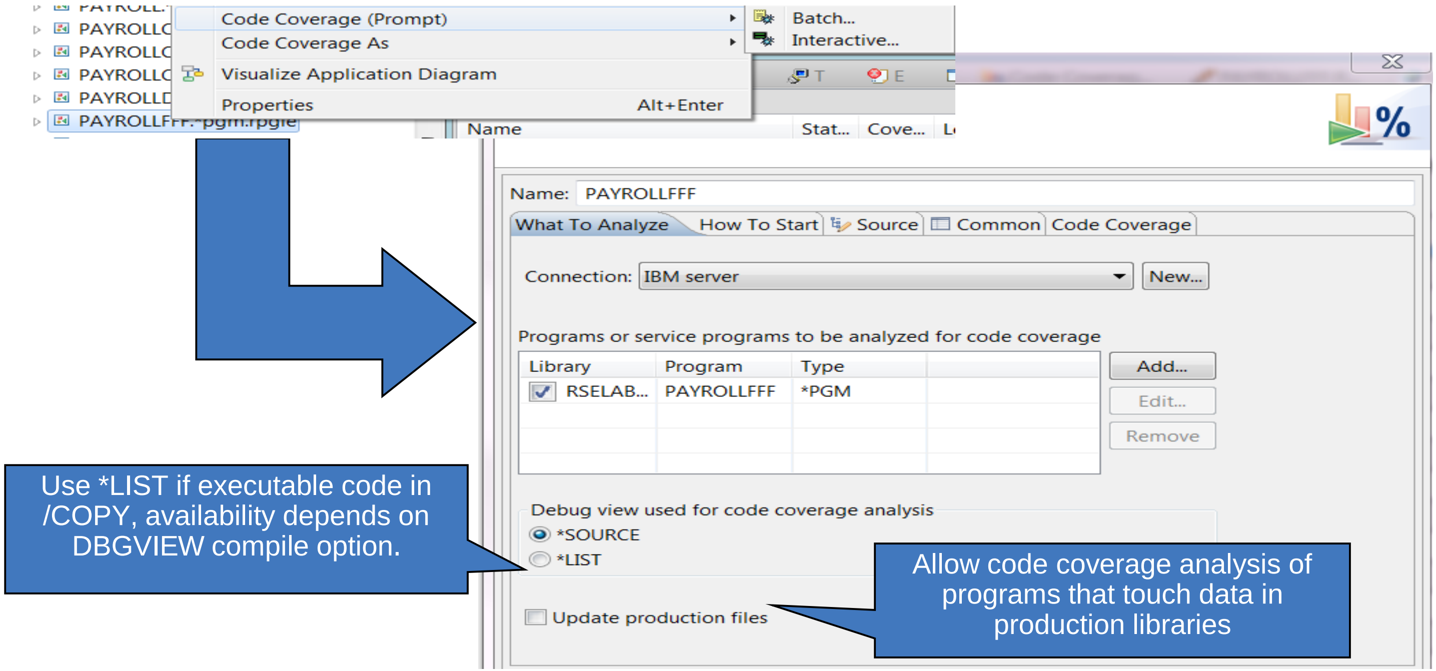
All Source Files
RBC4NNF1C.RPGLE (88%)

Code Coverage Report

Overall Summary > *PGM MKWAN/RBC4NNF1C > RBC4NNF1C > RBC4NNF1C.RPGLE

Procedure Name	Covered Lines	Total Lines	%Coverage
GPHPRCCMP10 ()	0	6	0%
UCSPRCCMP8 ()	0	6	0%
INDPRC1 ()	0	6	0%
CHRPRCCMP12 ()	0	6	0%
PUT_TC_MSG ()	0	13	0%
finish_tc ()	7	9	78%
chk_result ()	8	9	89%
show_error ()	18	20	90%
RBC4NNF1C ()	406	444	91%
chk_subres ()	24	26	92%
DATPRCISO ()	3	3	100%
DATPRCUSA ()	3	3	100%
DATPRCEUR ()	3	3	100%

Launch Options



Name: PAYROLLFFF

Connection: IBM server

Programs or service programs to be analyzed for code coverage

Library	Program	Type
☑ RSELAB...	PAYROLLFFF	*PGM

Debug view used for code coverage analysis

● *SOURCE

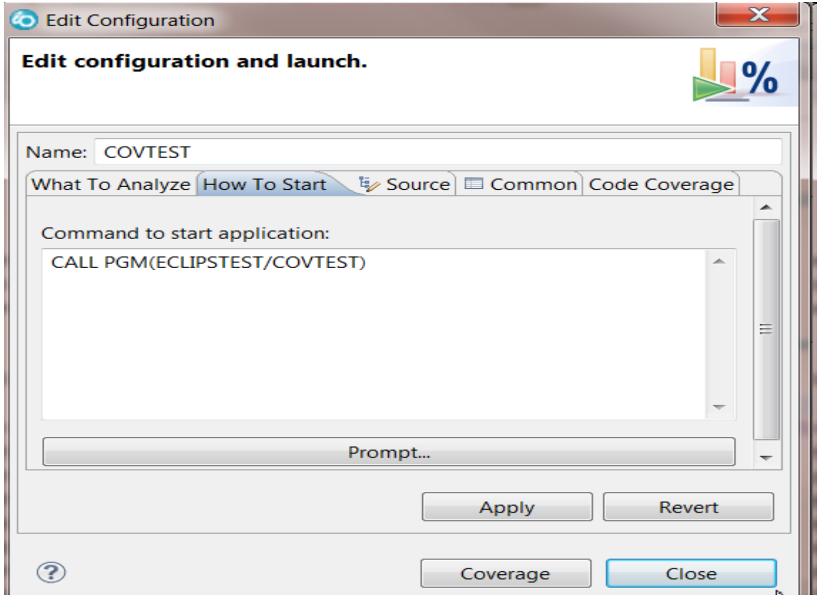
○ *LIST

□ Update production files

Use *LIST if executable code in /COPY, availability depends on DBGVIEW compile option.

Allow code coverage analysis of programs that touch data in production libraries

Customize how the program is invoked

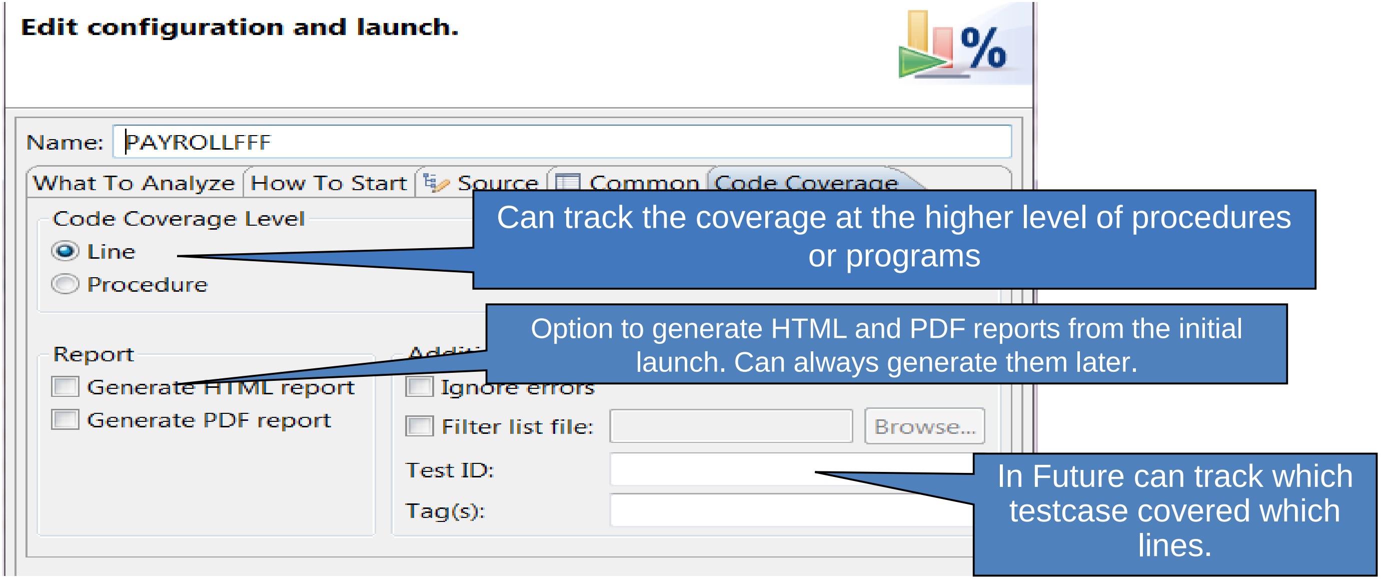


Name: COVTEST

Command to start application:

```
CALL PGM(ECLIPSTEST/COVTEST)
```

Customize the Code Coverage Analysis



Edit configuration and launch.

Name: PAYROLLFFF

Code Coverage Level

● Line

○ Procedure

Report

□ Generate HTML report

□ Generate PDF report

□ Ignore errors

□ Filter list file:

Test ID:

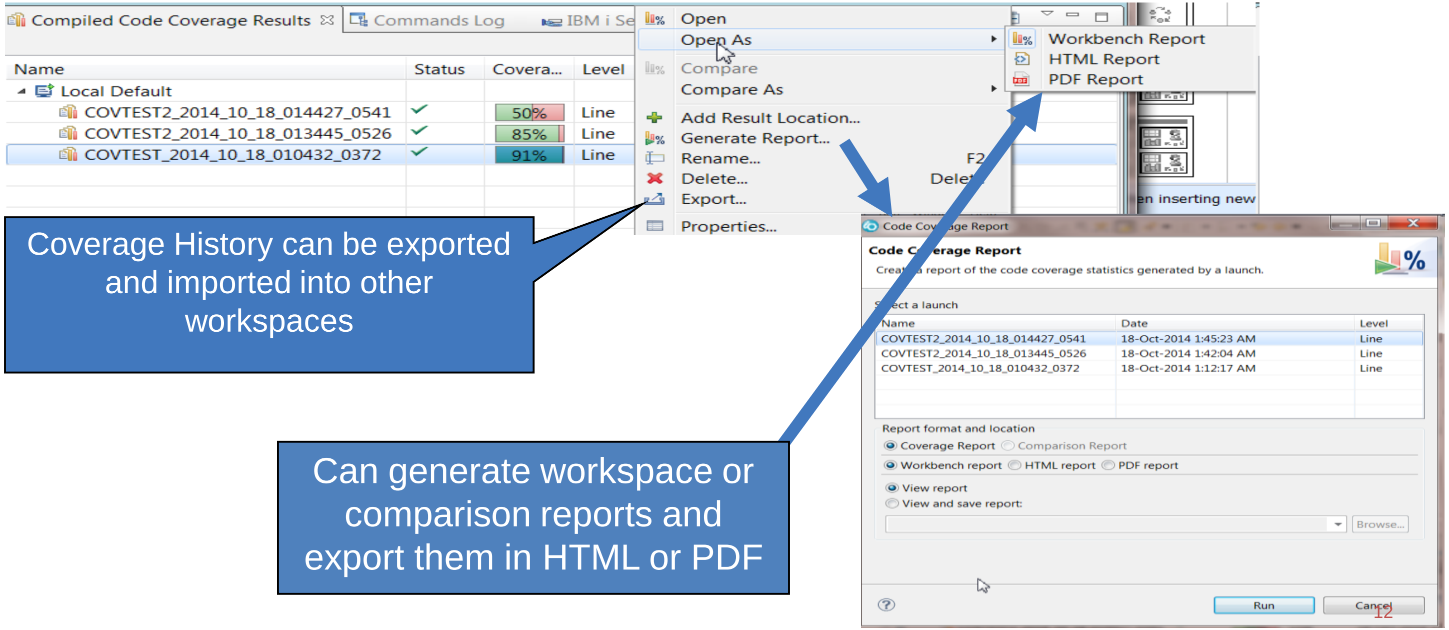
Tag(s):

Can track the coverage at the higher level of procedures or programs

Option to generate HTML and PDF reports from the initial launch. Can always generate them later.

In Future can track which testcase covered which lines.

Code Coverage History view operations on reports



Name	Status	Covera...	Level
Local Default			
COVTEST2_2014_10_18_014427_0541	✓	50%	Line
COVTEST2_2014_10_18_013445_0526	✓	85%	Line
COVTEST_2014_10_18_010432_0372	✓	91%	Line

Code Coverage Report

Code Coverage Report

Create a report of the code coverage statistics generated by a launch.

Select a launch

Name	Date	Level
COVTEST2_2014_10_18_014427_0541	18-Oct-2014 1:45:23 AM	Line
COVTEST2_2014_10_18_013445_0526	18-Oct-2014 1:42:04 AM	Line
COVTEST_2014_10_18_010432_0372	18-Oct-2014 1:12:17 AM	Line

Report format and location

● Coverage Report ○ Comparison Report

● Workbench report ○ HTML report ○ PDF report

● View report

○ View and save report:

Browse...

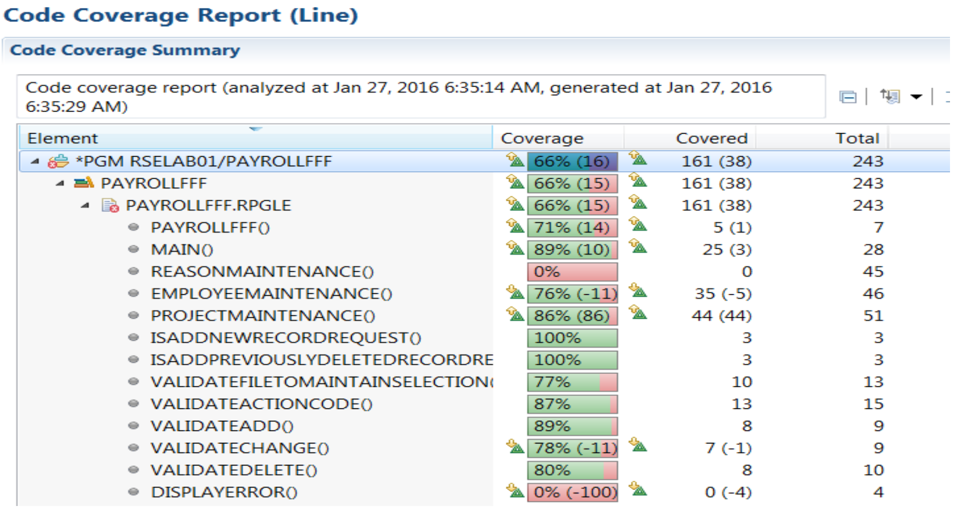
Run Cancel

Coverage History can be exported and imported into other workspaces

Can generate workspace or comparison reports and export them in HTML or PDF

Compare 2 runs to see improvement in coverage

- Select 2 runs in the history view and compare – see % improvement for different procedures



Code Coverage Report (Line)

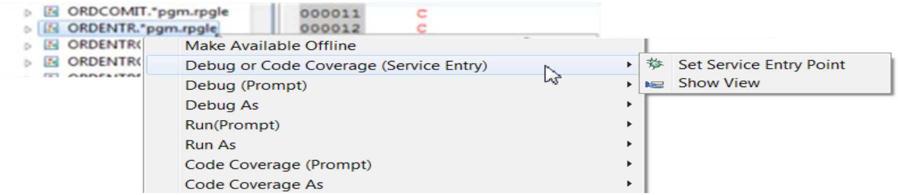
Code Coverage Summary

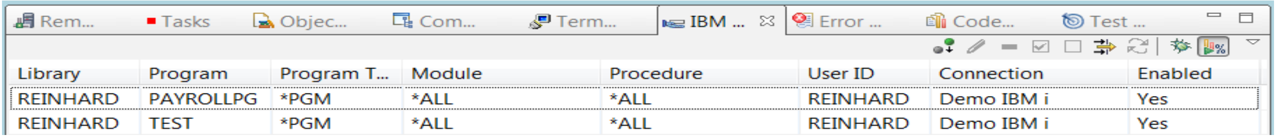
Code coverage report (analyzed at Jan 27, 2016 6:35:14 AM, generated at Jan 27, 2016 6:35:29 AM)

Element	Coverage	Covered	Total
*PGM RSELAB01/PAYROLLFFF	66% (16)	161 (38)	243
PAYROLLFFF	66% (15)	161 (38)	243
PAYROLLFFF.RPGLE	66% (15)	161 (38)	243
PAYROLLFFF()	71% (14)	5 (1)	7
MAIN()	89% (10)	25 (3)	28
REASONMAINTENANCE()	0%	0	45
EMPLOYEEMAINTENANCE()	76% (-11)	35 (-5)	46
PROJECTMAINTENANCE()	86% (86)	44 (44)	51
ISADDNEWRECORDREQUEST()	100%	3	3
ISADDPREVIOUSLYDELETEDRECORDRE	100%	3	3
VALIDATEFILETOMAINTAINSELECTION(	77%	10	13
VALIDATEACTIONCODE()	87%	13	15
VALIDATEADD()	89%	8	9
VALIDATECHANGE()	78% (-11)	7 (-1)	9
VALIDATEDELETE()	80%	8	10
DISPLAYERROR()	0% (-100)	0 (-4)	4

Service Entry Points

- A special kind of entry breakpoint that can be set directly from Remote System Explorer
- Debugger is automatically started when first line of program is executed
- Supports batch, interactive, and Web service programs

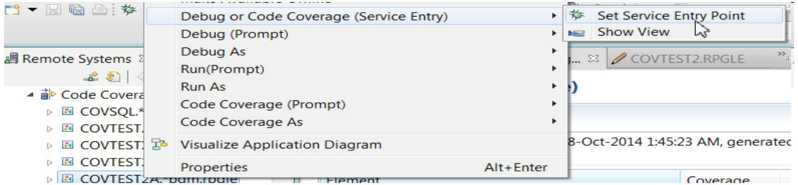




Library	Program	Program T...	Module	Procedure	User ID	Connection	Enabled
REINHARD	PAYROLLPG	*PGM	*ALL	*ALL	REINHARD	Demo IBM i	Yes
REINHARD	TEST	*PGM	*ALL	*ALL	REINHARD	Demo IBM i	Yes

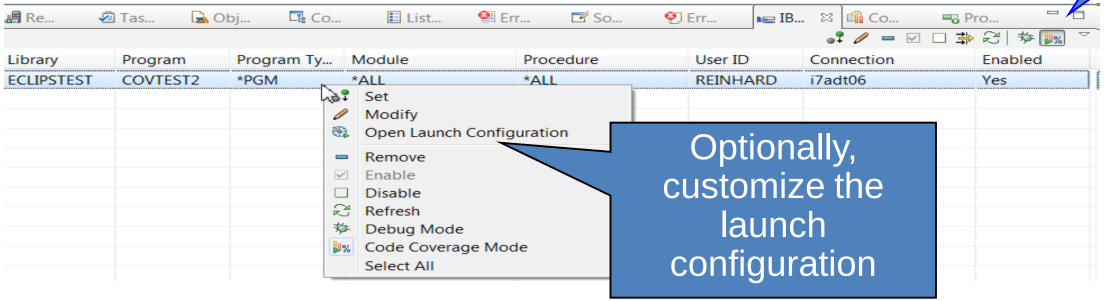
Use Service Entry Breakpoints with Code Coverage

- Set the SEP breakpoint just as you would for debugging



- In SEP view, toggle to Code Coverage mode from Debug mode

Switch to Code Coverage mode



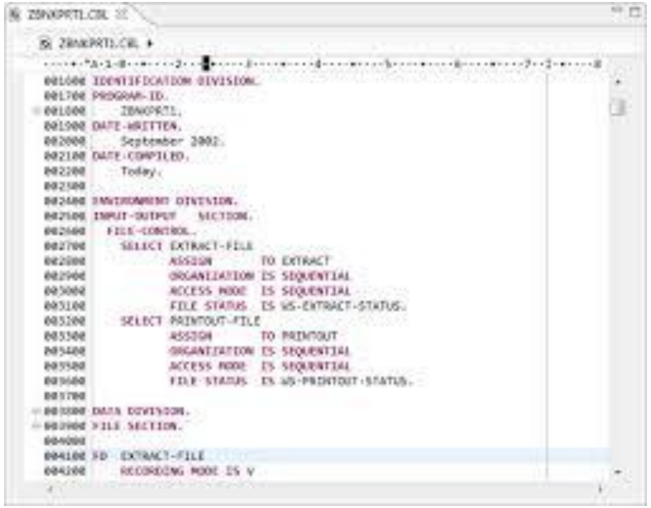
Library	Program	Program Ty...	Module	Procedure	User ID	Connection	Enabled
ECLIPSTEST	COVTEST2	*PGM	*ALL	*ALL	REINHARD	i7adt06	Yes

Optionally, customize the launch configuration

- Invoke the application in whatever manner you normally would

RDi 9.1.1 COBOL Code Coverage Enhancement

- The coverage analysis is now reported by paragraph as well as by line



ZBNKPRT1.CBL

```
----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7-|--+----8
001600 IDENTIFICATION DIVISION.
001700 PROGRAM-ID.
001800     ZBNKPRT1.
001900 DATE-WRITTEN.
002000     September 2002.
002100 DATE-COMPILED.
002200     Today.
002300
002400 ENVIRONMENT DIVISION.
002500 INPUT-OUTPUT   SECTION.
002600   FILE-CONTROL.
002700     SELECT EXTRACT-FILE
002800            ASSIGN       TO EXTRACT
002900            ORGANIZATION IS SEQUENTIAL
003000            ACCESS MODE  IS SEQUENTIAL
003100            FILE STATUS  IS WS-EXTRACT-STATUS.
003200     SELECT PRINTOUT-FILE
003300            ASSIGN       TO PRINTOUT
003400            ORGANIZATION IS SEQUENTIAL
003500            ACCESS MODE  IS SEQUENTIAL
003600            FILE STATUS  IS WS-PRINTOUT-STATUS.
003700
003800 DATA DIVISION.
003900 FILE SECTION.
004000
004100 FD  EXTRACT-FILE
004200     RECORDING MODE IS V
```

New Code Coverage Features in RDi 9.5

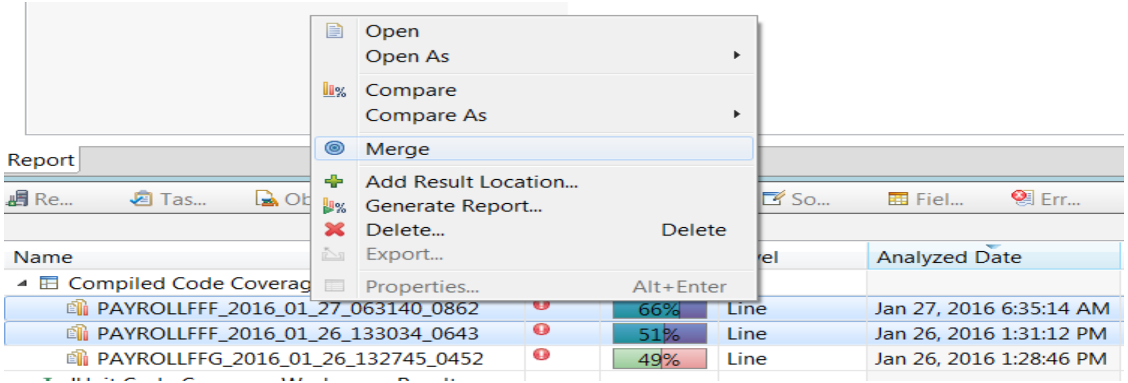
- Much, much, much faster! Typically 20x or better improvement
  - The architecture was completely redone to streamline code coverage to a minimum of interaction between the RDi client and the IBM i server
  - Solution is ready for real world scale of programs and tests

- Ability to use shared results, so coverage results of test cases can be computed once and reused for the team

- Ability to merge code coverage results from multiple runs
  - Compute the combined coverage and show exactly which lines are missing
  - Exactly which modules/files/procedures do not meet specified thresholds

The results of multiple test runs can be merged

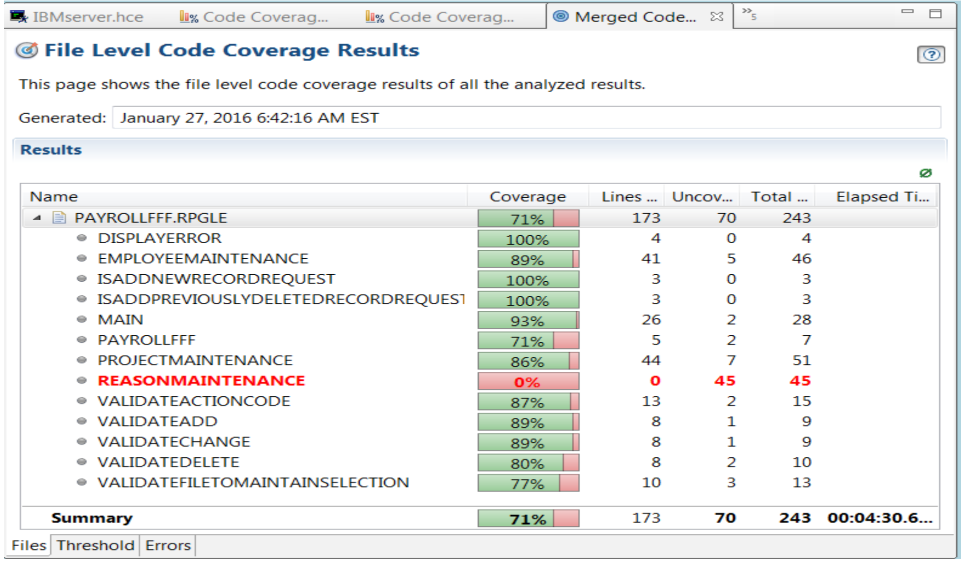
- The bucket of manual or automatic tests are all run
- The coverage results are all analyzed and merged into a single baseline



Name			el	Analyzed Date
Compiled Code Coverag				
PAYROLLFFF_2016_01_27_063140_0862		66%	Line	Jan 27, 2016 6:35:14 AM
PAYROLLFFF_2016_01_26_133034_0643		51%	Line	Jan 26, 2016 1:31:12 PM
PAYROLLFFG_2016_01_26_132745_0452		49%	Line	Jan 26, 2016 1:28:46 PM

Holes in the Testbucket can now be analyzed

- Procedures and files below a set threshold are highlighted in red and need more testing



File Level Code Coverage Results

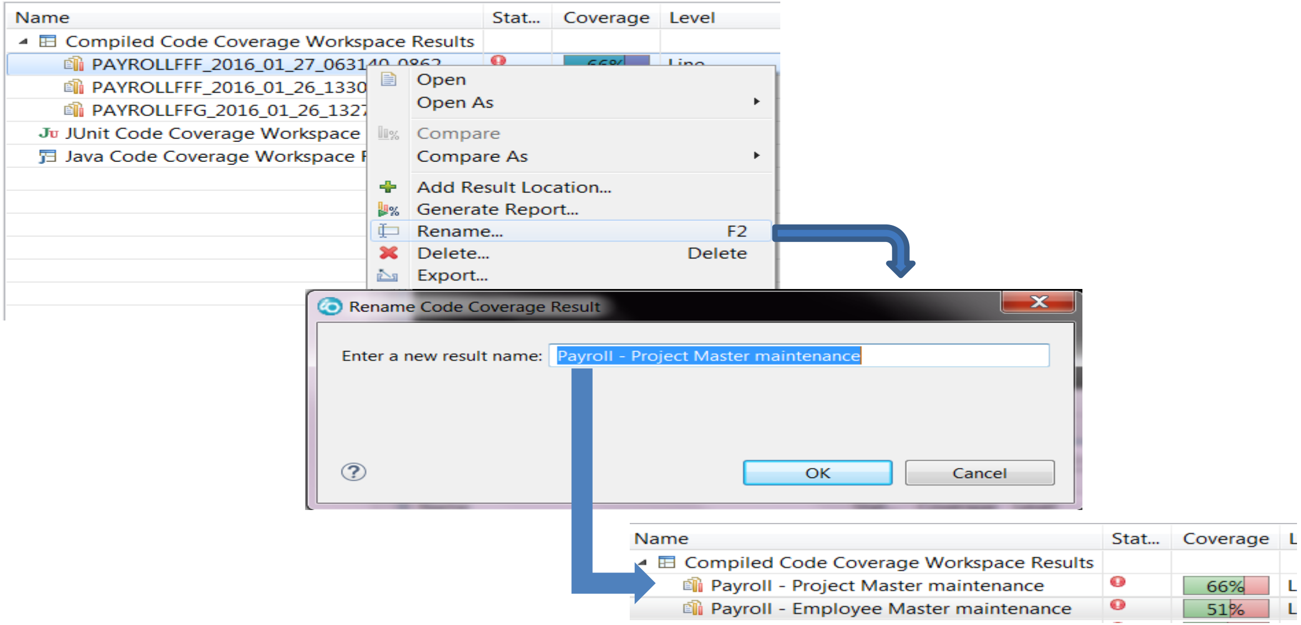
This page shows the file level code coverage results of all the analyzed results.

Generated: January 27, 2016 6:42:16 AM EST

Results

Name	Coverage	Lines ...	Uncov...	Total ...	Elapsed Ti...
PAYROLLFFF.RPGLE	71%	173	70	243	
DISPLAYERROR	100%	4	0	4	
EMPLOYEEMAINTENANCE	89%	41	5	46	
ISADDNEWRECORDREQUEST	100%	3	0	3	
ISADDPREVIOUSLYDELETEDRECORDREQUEST	100%	3	0	3	
MAIN	93%	26	2	28	
PAYROLLFFF	71%	5	2	7	
PROJECTMAINTENANCE	86%	44	7	51	
REASONMAINTENANCE	**0%**	**0**	**45**	**45**	
VALIDATEACTIONCODE	87%	13	2	15	
VALIDATEADD	89%	8	1	9	
VALIDATECHANGE	89%	8	1	9	
VALIDATEDELETE	80%	8	2	10	
VALIDATEFILETOMAINTAINSELECTION	77%	10	3	13	
Summary	**71%**	173	**70**	**243**	**00:04:30.6...**

Results can be given meaningful names



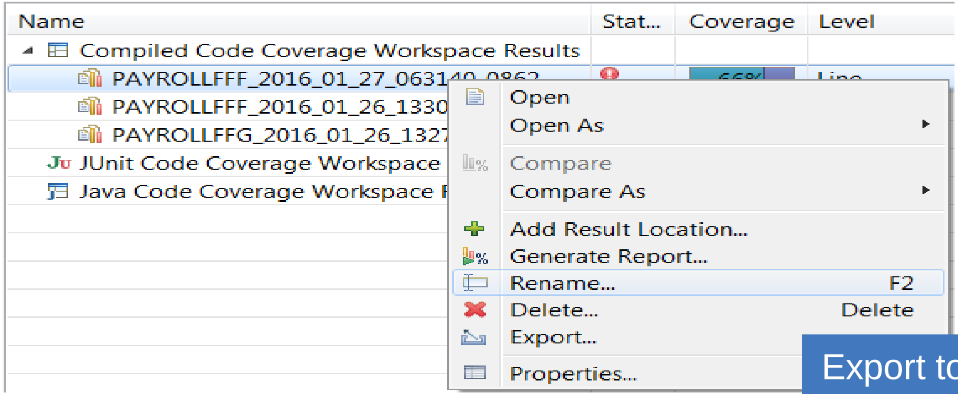
Name	Stat...	Coverage	Level
Compiled Code Coverage Workspace Results			
PAYROLLFFF_2016_01_27_063140_0862		66%	Line
PAYROLLFFF_2016_01_26_1330			
PAYROLLFFG_2016_01_26_1327			
JUnit Code Coverage Workspace			
Java Code Coverage Workspace R			

Rename Code Coverage Result

Enter a new result name: Payroll - Project Master maintenance

Name	Stat...	Coverage	L
Compiled Code Coverage Workspace Results			
Payroll - Project Master maintenance		66%	L
Payroll - Employee Master maintenance		51%	L

Results can be exported to a shared location

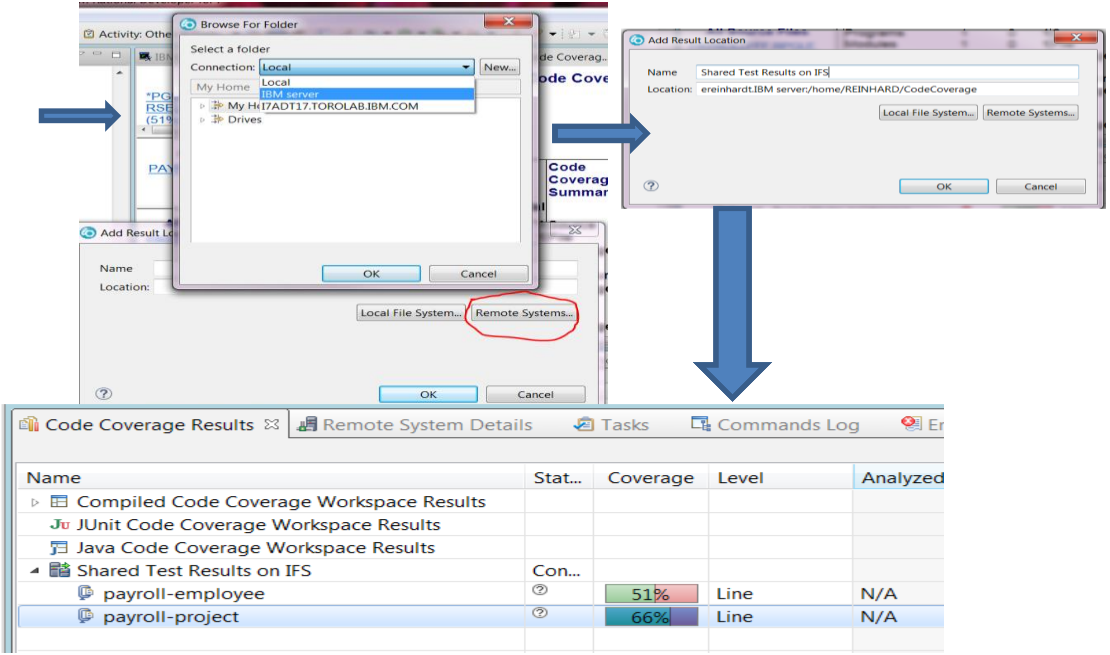


Name	Stat...	Coverage	Level
Compiled Code Coverage Workspace Results			
PAYROLLFFF_2016_01_27_063140_0862		66%	Line
PAYROLLFFF_2016_01_26_1330			
PAYROLLFFG_2016_01_26_1327			
JUnit Code Coverage Workspace			
Java Code Coverage Workspace F			

Export to:

- Your PC
- IFS
- Shared drive

Shared result can be viewed & analyzed from any RDi client



Connection: Local

- Local
- IBM server
- I7ADT17.TOROLAB.IBM.COM

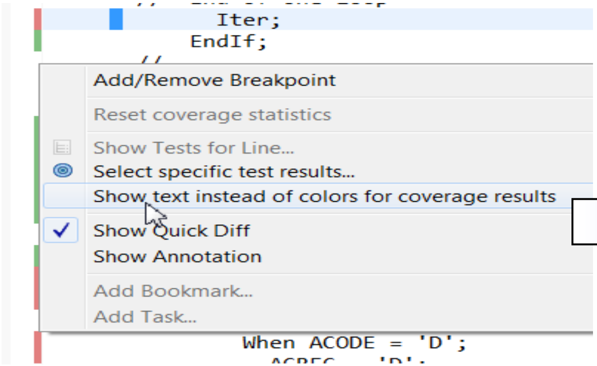
Name: Shared Test Results on IFS

Location: ereinhardt.IBM server:/home/REINHARD/CodeCoverage

Name	Stat...	Coverage	Level	Analyzed
Compiled Code Coverage Workspace Results				
JUnit Code Coverage Workspace Results				
Java Code Coverage Workspace Results				
Shared Test Results on IFS	Con...			
payroll-employee	?	51%	Line	N/A
payroll-project	?	66%	Line	N/A

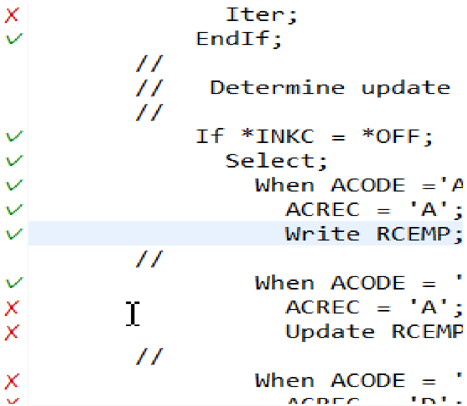
Customize coverage annotations

Some may be color-blind and so the annotations can be customized to show different symbols as well!



```
            Iter;
          EndIf;
                When ACODE = 'D';
```

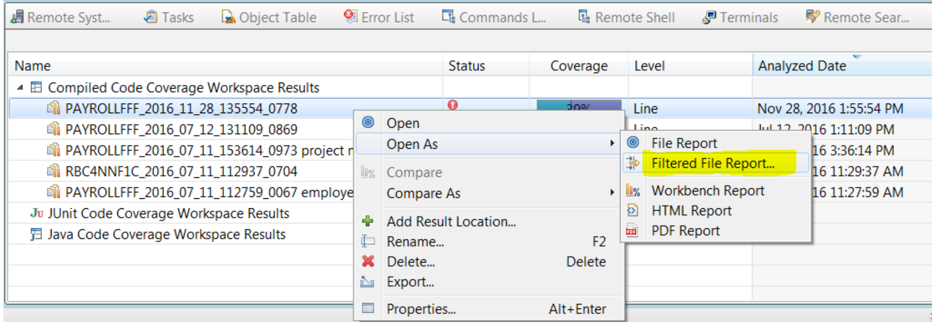
Add/Remove Breakpoint
Reset coverage statistics
Show Tests for Line...
Select specific test results...
Show text instead of colors for coverage results
✓ Show Quick Diff
Show Annotation
Add Bookmark...
Add Task...



```
✗             Iter;
✓           EndIf;
       //
       //   Determine update
       //
✓           If *INKC = *OFF;
✓             Select;
✓               When ACODE ='A
✓                 ACREC = 'A';
✓                 Write RCEMP;
       //
✓               When ACODE = '
✗                 ACREC = 'A';
✗                 Update RCEMP
       //
✗               When ACODE = '
```

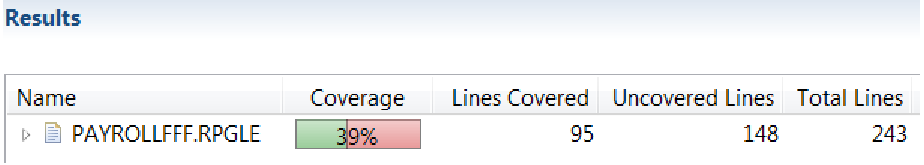
Code Coverage – Filtered File Report

- After running code coverage, open as a Filtered File report option is available



Name	Status	Coverage	Level	Analyzed Date
Compiled Code Coverage Workspace Results				
PAYROLLFFF_2016_11_28_135554_0778			Line	Nov 28, 2016 1:55:54 PM
PAYROLLFFF_2016_07_12_131109_0869			Line	Jul 12, 2016 1:11:09 PM
PAYROLLFFF_2016_07_11_153614_0973 project r				16 3:36:14 PM
RBC4NNF1C_2016_07_11_112937_0704				16 11:29:37 AM
PAYROLLFFF_2016_07_11_112759_0067 employe				16 11:27:59 AM
JUnit Code Coverage Workspace Results				
Java Code Coverage Workspace Results				

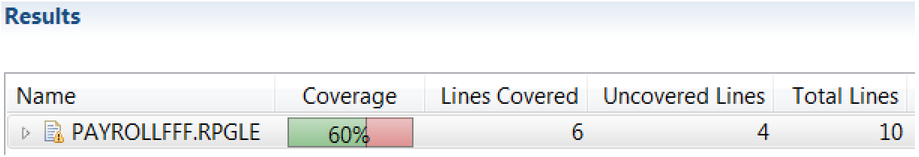
- Specify a filter string and the report will show code coverage results for lines that match the filter string
  - File report



Results

Name	Coverage	Lines Covered	Uncovered Lines	Total Lines
PAYROLLFFF.RPGLE	39%	95	148	243

  - Filtered report



Results

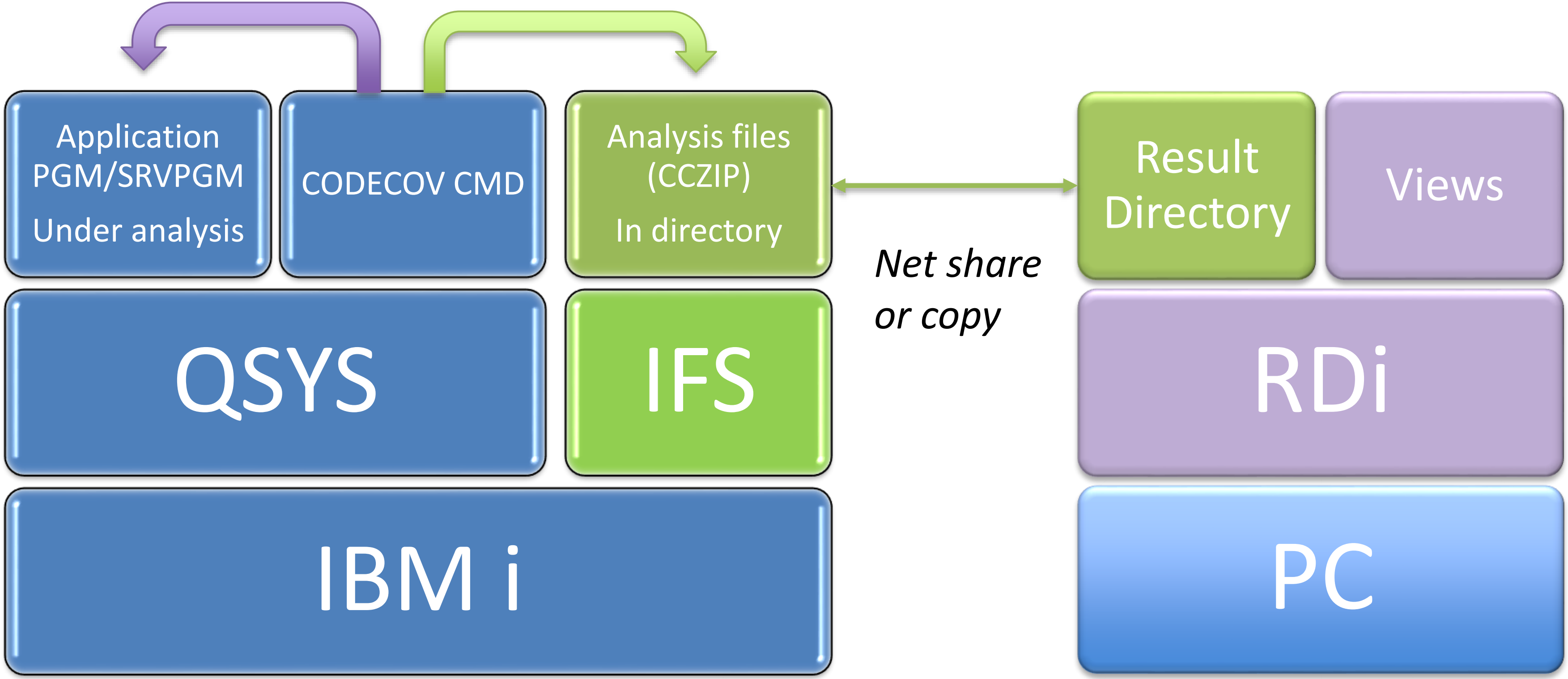
Name	Coverage	Lines Covered	Uncovered Lines	Total Lines
PAYROLLFFF.RPGLE	60%	6	4	10

Introduction to IBM i Headless Code Coverage

Imagine…

- Having code coverage on an individual invocation is interesting
- But for transforming your quality, things have to be automated
- What if you could check the coverage of your regular test runs?
- And you only used RDi to visualize the results.

How headless code coverage is structured



Requirements on OS and Programs

- Freely available with RDS option 60 on the IBM i

IBM i	MRM	MRI
V7R4	5770WDS option 60	
V7R3	PTF 5770WDS SI65229	2924 PTF 5770WDS SI64655
V7R2	PTF 5770WDS SI65228	2924 PTF 5770WDS SI64544
V7R1	Time to upgrade	

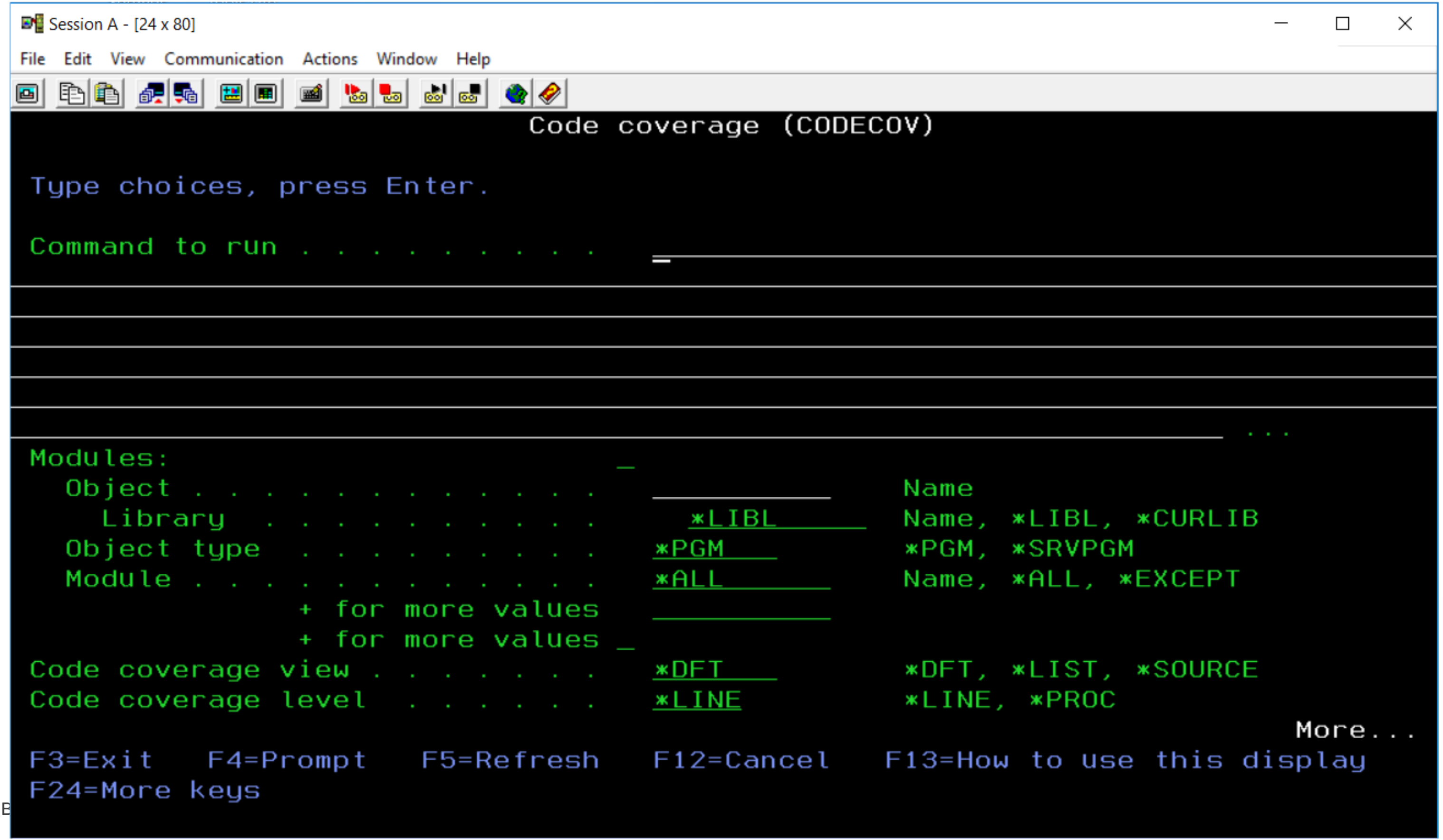
Requirements on job environment

- Programs and service programs for instrumentation must be compiled with one of the following debug options:
  - DBGVIEW(*SOURCE)
  - DBGVIEW(*LIST)
  - DBGVIEW(*ALL)
- Headless code coverage tools are located under the QDEVTOOLS library
- Add it to the library list: ADDLIBLE QDEVTOOLS
- Main command: **QDEVTOOLS/CODECOV**
- **As of GA CODECOV will be in QSYS and there will be no need to modify the library list.**

Features of Headless Code Coverage Tools

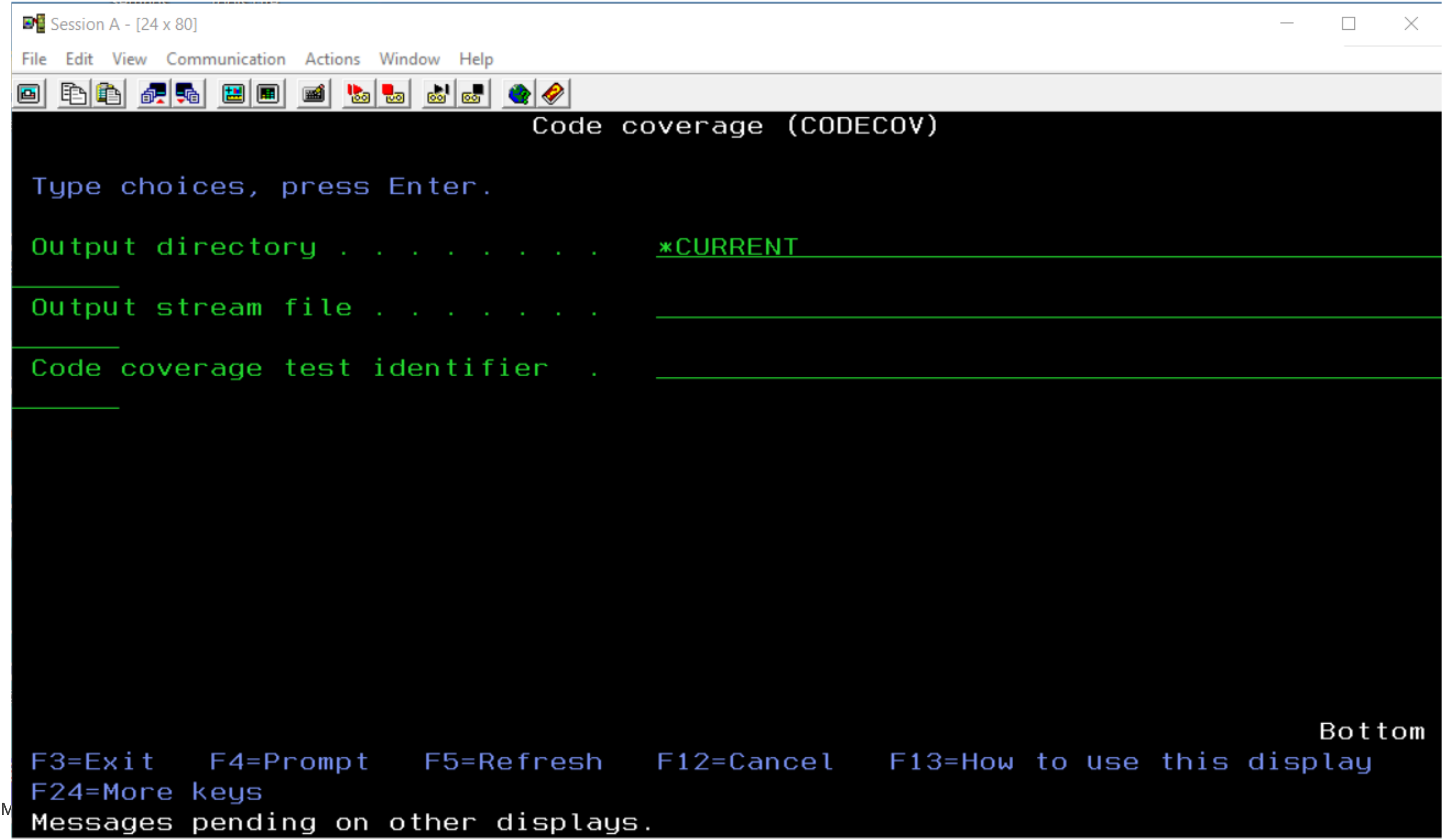
- Start a code coverage session against a run command
  - Support both “CALL” and customized user commands
- Add programs and service programs into the instrumentation list
- Module filter support
  - A list of modules
    - (M1 M2 M3)
  - All modules
    - (*ALL)
  - EXCEPT keyword:
    - (*EXCEPT M1 M2)   - All modules except M1 and M2
- Support both *SOURCE and *LIST views
- Support LINE and PROCEDURE level coverage
- Save code coverage result in designated location
  - When output directory is specified, save result as a generated file under the target directory
  - When output stream file is specified, save result in the designated file

Prompt screens for CODECOV 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, *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
```

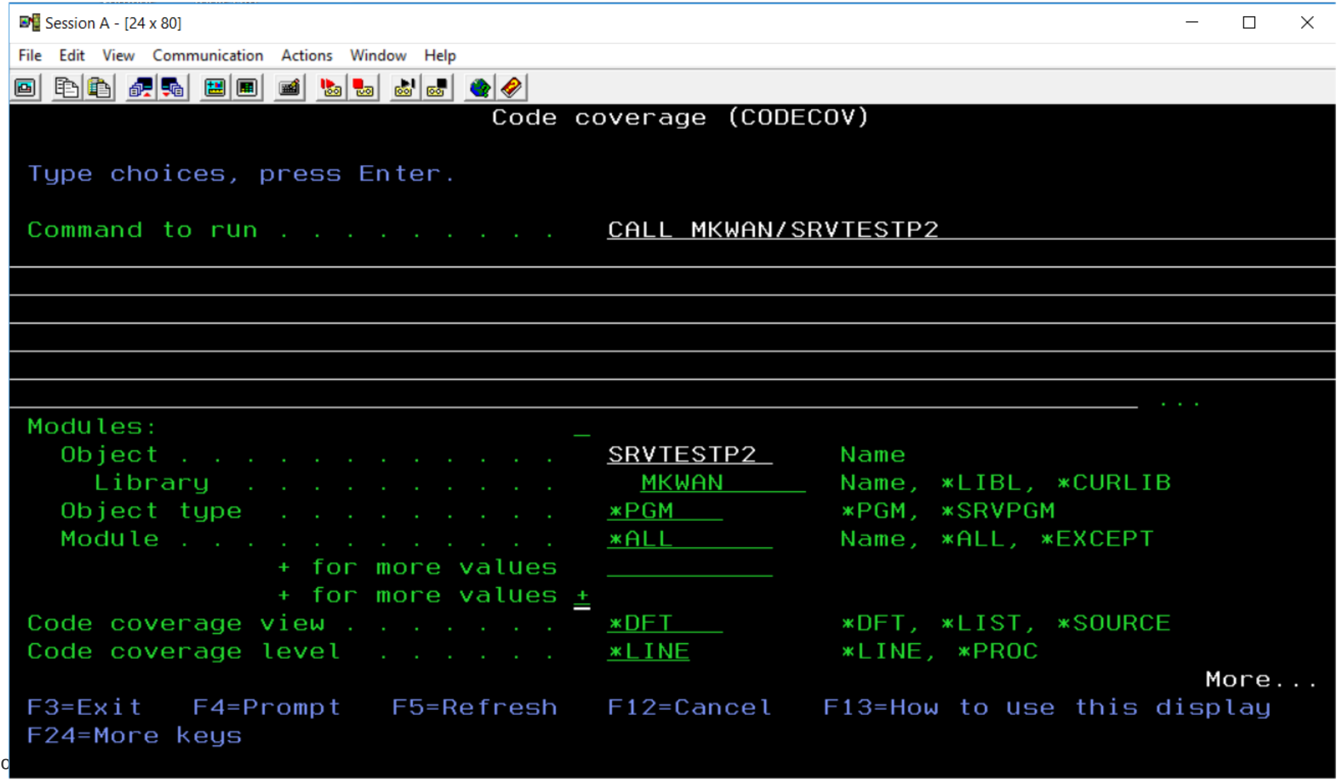
Prompt screens for CODECOV command (cont.)



```
                           Code coverage (CODECOV)

 Type choices, press Enter.

 Output directory . . . . . . . .   *CURRENT

 Output stream file . . . . . . .

 Code coverage test identifier  .

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Messages pending on other displays.
```

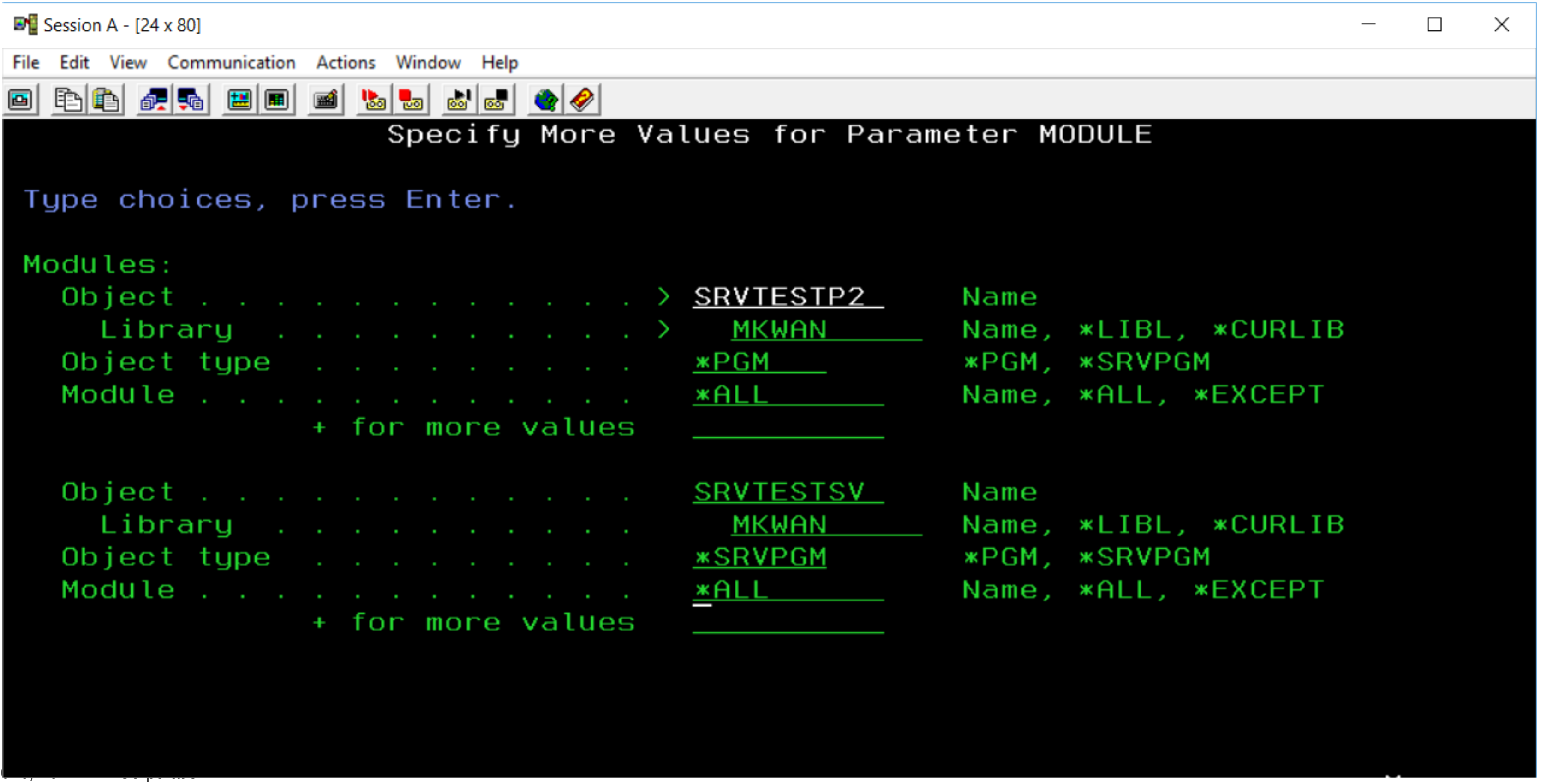
Include one program into the MODULES list

- Enter one program into the Modules list
- Enter “+” into the second “+ for more values” field



```
                          Code coverage (CODECOV)

 Type choices, press Enter.

 Command to run . . . . . . . . .   CALL MKWAN/SRVTESTP2

                                                                          ...
 Modules:                          _
   Object . . . . . . . . . . . .   SRVTESTP2       Name
     Library  . . . . . . . . . .     MKWAN         Name, *LIBL, *CURLIB
   Object type  . . . . . . . . .   *PGM            *PGM, *SRVPGM
   Module . . . . . . . . . . . .   *ALL            Name, *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
```

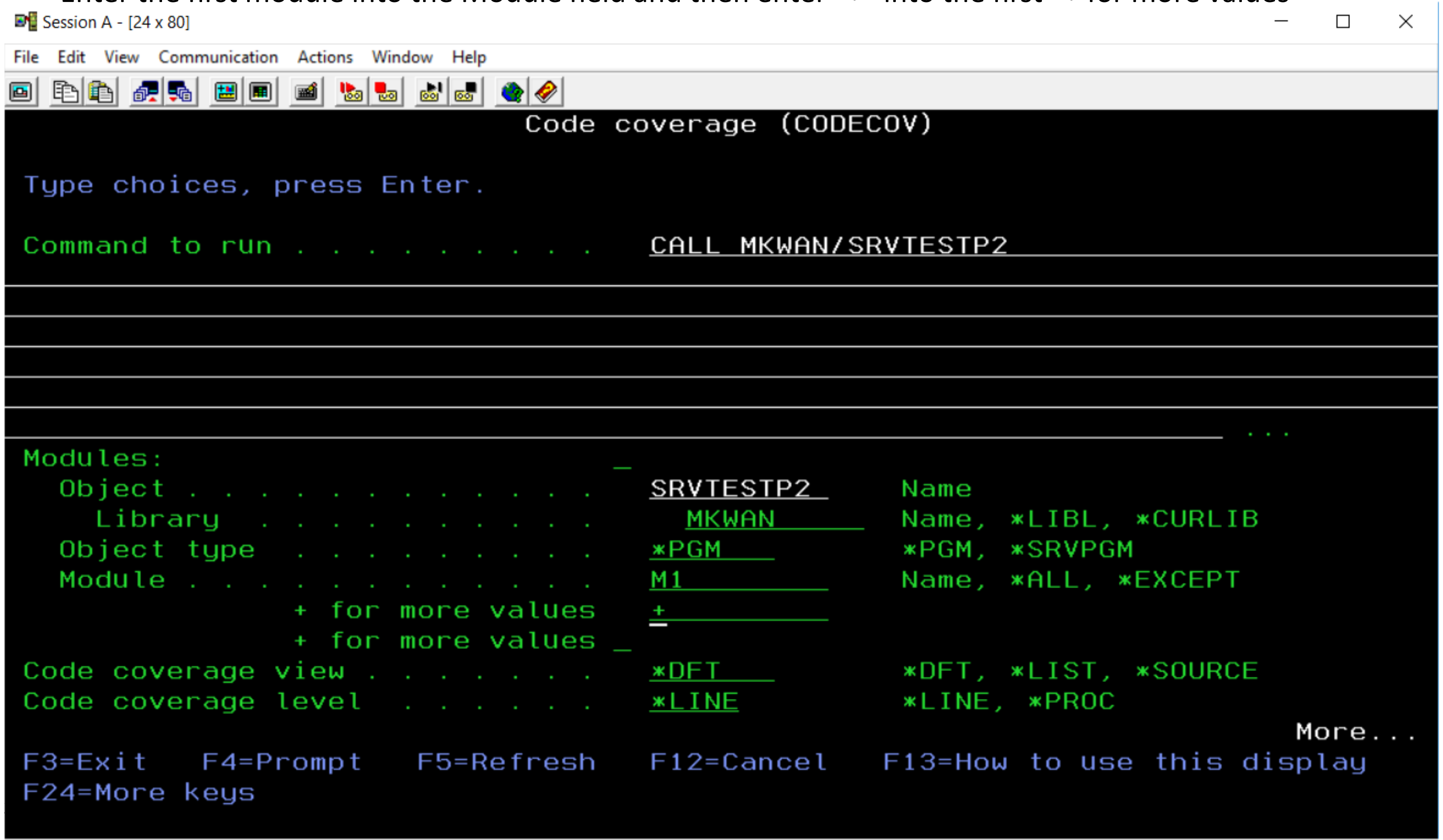
Include another service program into the MODULES list



```
                 Specify More Values for Parameter MODULE

 Type choices, press Enter.

 Modules:
   Object . . . . . . . . . . . . > SRVTESTP2       Name
     Library  . . . . . . . . . . >   MKWAN         Name, *LIBL, *CURLIB
   Object type  . . . . . . . . .   *PGM            *PGM, *SRVPGM
   Module . . . . . . . . . . . .   *ALL            Name, *ALL, *EXCEPT
                + for more values   __________

   Object . . . . . . . . . . . .   SRVTESTSV       Name
     Library  . . . . . . . . . .     MKWAN         Name, *LIBL, *CURLIB
   Object type  . . . . . . . . .   *SRVPGM         *PGM, *SRVPGM
   Module . . . . . . . . . . . .   *ALL            Name, *ALL, *EXCEPT
                + for more values   __________
```

Specify a list of modules

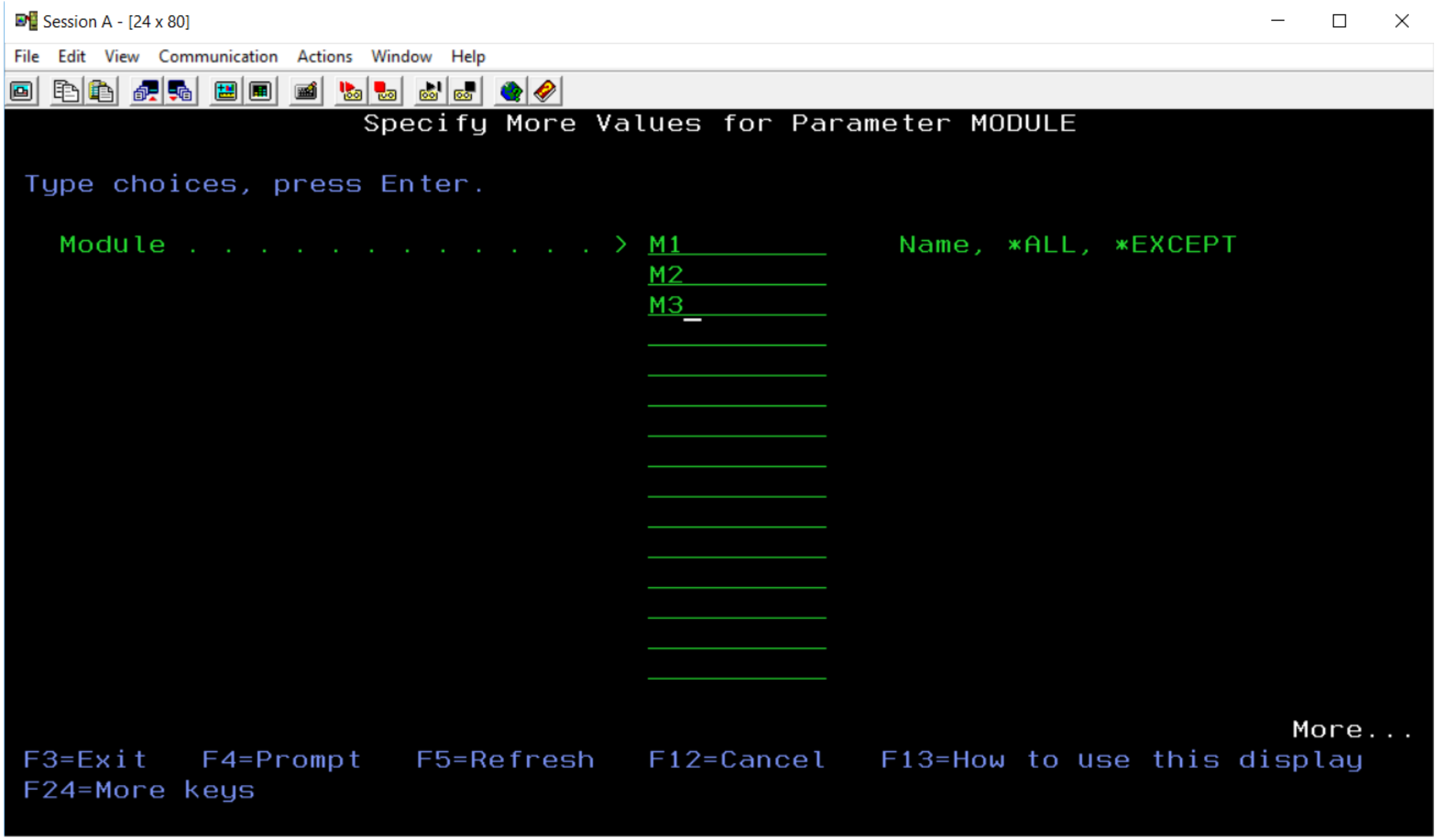
- Enter the first module into the Module field and then enter “+” into the first “+ for more values”



```
                             Code coverage (CODECOV)

 Type choices, press Enter.

 Command to run . . . . . . . . .   CALL MKWAN/SRVTESTP2






                                                                     ...
 Modules:                         _
   Object . . . . . . . . . . . .   SRVTESTP2     Name
     Library  . . . . . . . . . .     MKWAN       Name, *LIBL, *CURLIB
   Object type  . . . . . . . . .   *PGM          *PGM, *SRVPGM
   Module . . . . . . . . . . . .   M1            Name, *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
```

Specify a list of modules (cont.)

Enter other modules



```
           Specify More Values for Parameter MODULE

Type choices, press Enter.

  Module . . . . . . . . . . . . > M1__________     Name, *ALL, *EXCEPT
                                   M2__________
                                   M3__________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________
                                   ____________

                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Mandatory Parameters

- Two parameters are mandatory:
  - Command to run: The full command line that will run in code coverage instrumentation mode
    - Use CALL command for running a program, specify program parameters using the PARM parameter:
      - `CALL LIBNAME/PGMNAME PARM(…)`
    - Also support customized user command:
      - `MYCOMMAND MYPARM(…)`
    - Command length restriction: 6000
  - At least one entry in the MODULES list
    - Include at least one program or service program for code coverage instrumentation
    - Does not have to be the main program

Code Coverage View

- Three code coverage view configurations are supported:
  - *SOURCE
  - *LIST
  - *DFT
- Programs and service programs included in instrumentation must be compiled with one of the following debug options:
  - DBGVIEW(*SOURCE)
  - DBGVIEW(*LIST)
  - DBGVIEW(*ALL)
- *SOURCE:
  - Use the *SOURCE view for source file display. The programs must be compiled with DBGVIEW(*SOURCE) or DBGVIEW(*ALL)
- *LIST:
  - Use the *LIST view for source file display. The programs must be compiled with DBGVIEW(*LIST) or DBGVIEW(*ALL)
- *DFT: The default behavior
  - Use the *SOURCE view when the program is compiled with DBGVIEW(*SOURCE) or DBGVIEW(*ALL)
  - Use the *LIST view when the program is compiled with DBGVIEW(*LIST)

Code Coverage Level

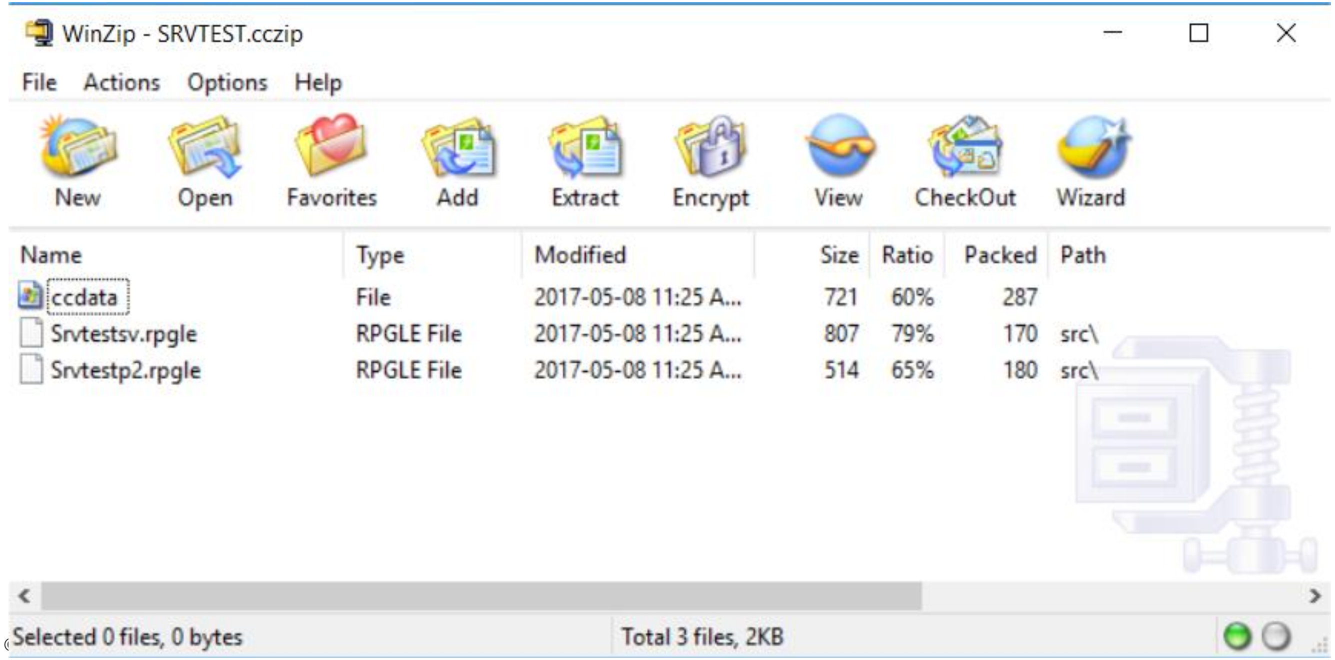
- *LINE: Line level
  - Include line-by-line statistics
  - The default level
- *PROC: Procedure level
  - Only capture the information whether a procedure entry point is hit or not
  - No line-by-line statistics
  - Run faster, but capture less information

Output options

- Two parameters to control output options:
  - Output directory and Output stream file
  - They cannot be specified at the same time
  - Error message is issued if they are specified at the same time:
    - Parameters OUTDIR and OUTSTMF cannot be used at the same time
- Output directory:
  - Code coverage result file is saved to this directory.
  - Use generated name for code coverage result file
    - Generated result name format:
      - *ProgramName_TimeStamp.cczip*
  - The default output directory is *CURRENT.
- Output stream file:
  - Fully qualified IFS file name for code coverage result file
  - If the base name does not have extension
    - Extension .cczip is appended to the base name
  - If the base name has an extension, it must be .cczip, otherwise an error is issued:
    - The extension for the result name … is not .cczip.

Code Coverage Result File

- Code coverage result file has .cczip extension
- Internally it is a ZIP file
- An XML entry (ccdata) contains code coverage statistics
- Source files are stored under the “src” folder inside the .cczip file



Name	Type	Modified	Size	Ratio	Packed	Path
ccdata	File	2017-05-08 11:25 A...	721	60%	287	
Srvtestsv.rpgle	RPGLE File	2017-05-08 11:25 A...	807	79%	170	src\
Srvtestp2.rpgle	RPGLE File	2017-05-08 11:25 A...	514	65%	180	src\

Selected 0 files, 0 bytes

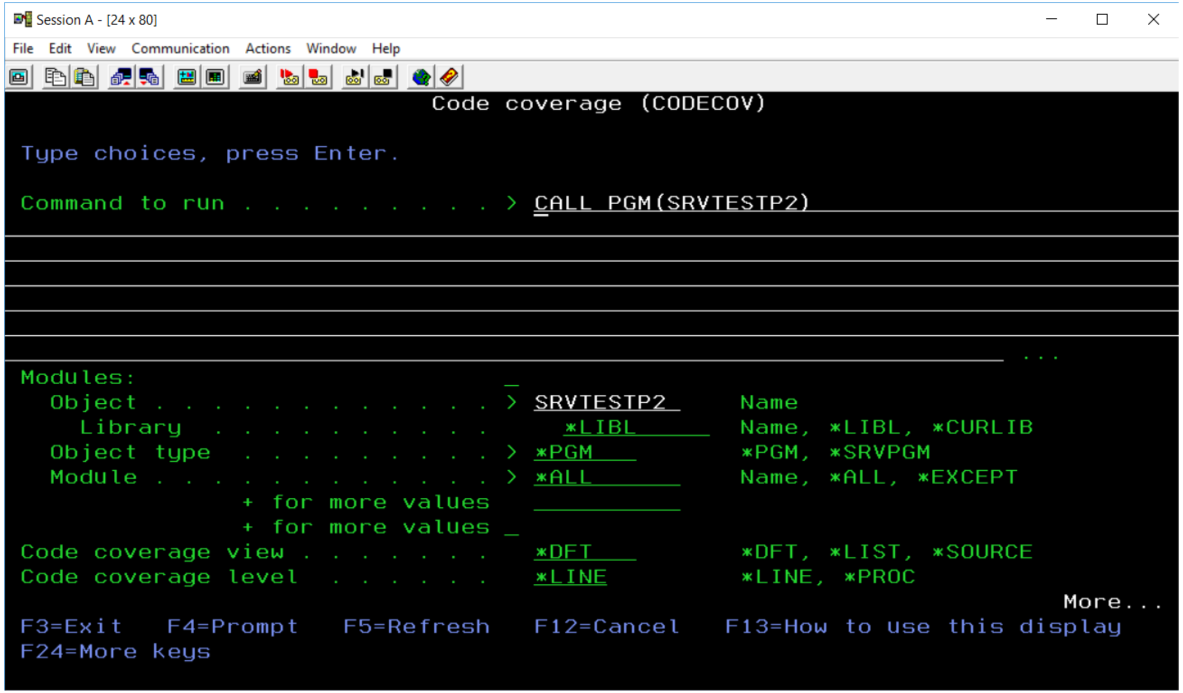
Total 3 files, 2KB

Code Coverage Test Identifier

- An identifier to set on the current code coverage session
    - Not used for any feature right now
    - Can be used for the merge function in the future
        - Code coverage results with the same test identifier can be merged together

A simple code coverage session

- Enter full command line in “Command to run”
- Enter one program into the “Modules” list
- Alternatively, type full command line from green screen:
  - CODECOV CMD(CALL SRVTESTP2) MODULE((SRVTESTP2))



```
                          Code coverage (CODECOV)

Type choices, press Enter.

Command to run . . . . . . . . . > CALL PGM(SRVTESTP2)







                                                                      ...
Modules:                                _
  Object . . . . . . . . . . . . . > SRVTESTP2     Name
    Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
  Object type  . . . . . . . . . . > *PGM          *PGM, *SRVPGM
  Module . . . . . . . . . . . . . > *ALL          Name, *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
```

Requirements: Viewing Code Coverage Data

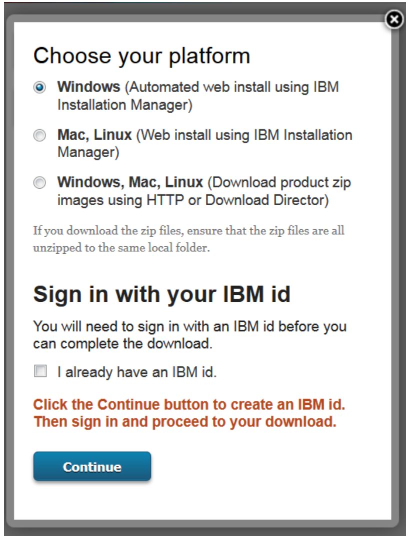
- CCZIP results generated by CODECOV can be viewed on RDi v9.5.1.2 or later

- Install RDi 9.5.1.2

  NOTE: If you don’t have access to the above location, install as follows:

  - Go to: https://www.ibm.com/developerworks/downloads/r/rdi/
  - Click on:

  - Select Window (Automated web install using IBM Installation Manager)
    - Follow the steps in Automated procedure for full instructions.



Choose your platform

● **Windows** (Automated web install using IBM Installation Manager)

○ **Mac, Linux** (Web install using IBM Installation Manager)

○ **Windows, Mac, Linux** (Download product zip images using HTTP or Download Director)

If you download the zip files, ensure that the zip files are all unzipped to the same local folder.

Sign in with your IBM id

You will need to sign in with an IBM id before you can complete the download.

□ I already have an IBM id.

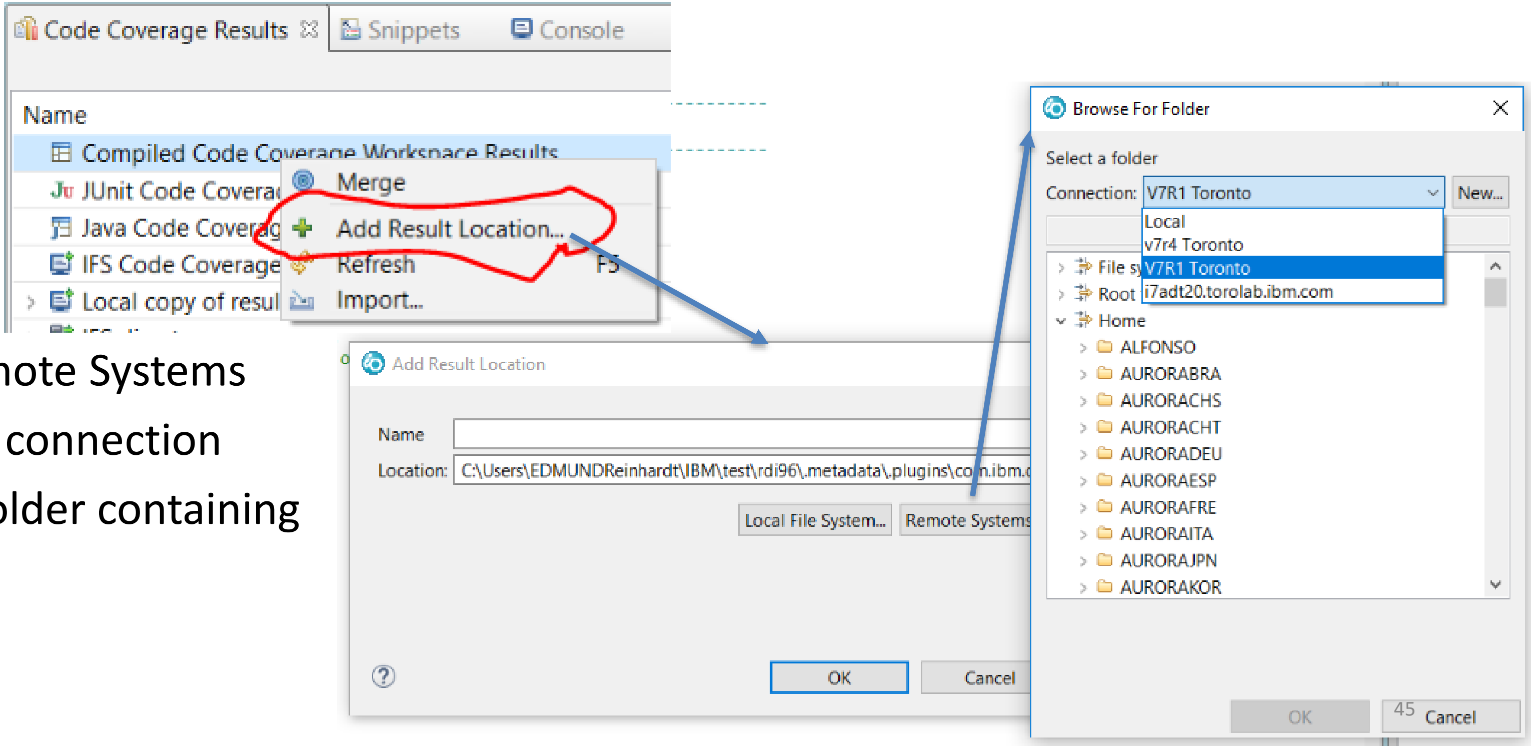
Click the Continue button to create an IBM id. Then sign in and proceed to your download.

Viewing headless code coverage results on RDi

- From Code Coverage Results view, add Result Location

- Specify Remote Systems
- Select IBMi connection
- Select IFS folder containing

  cczip files



Name:

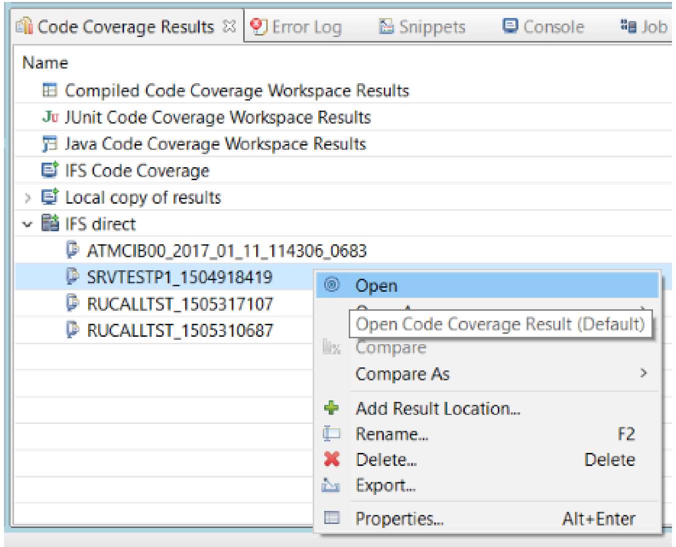
Location: C:\Users\EDMUNDReinhardt\IBM\test\rdi96\.metadata\.plugins\com.ibm.c

Select a folder

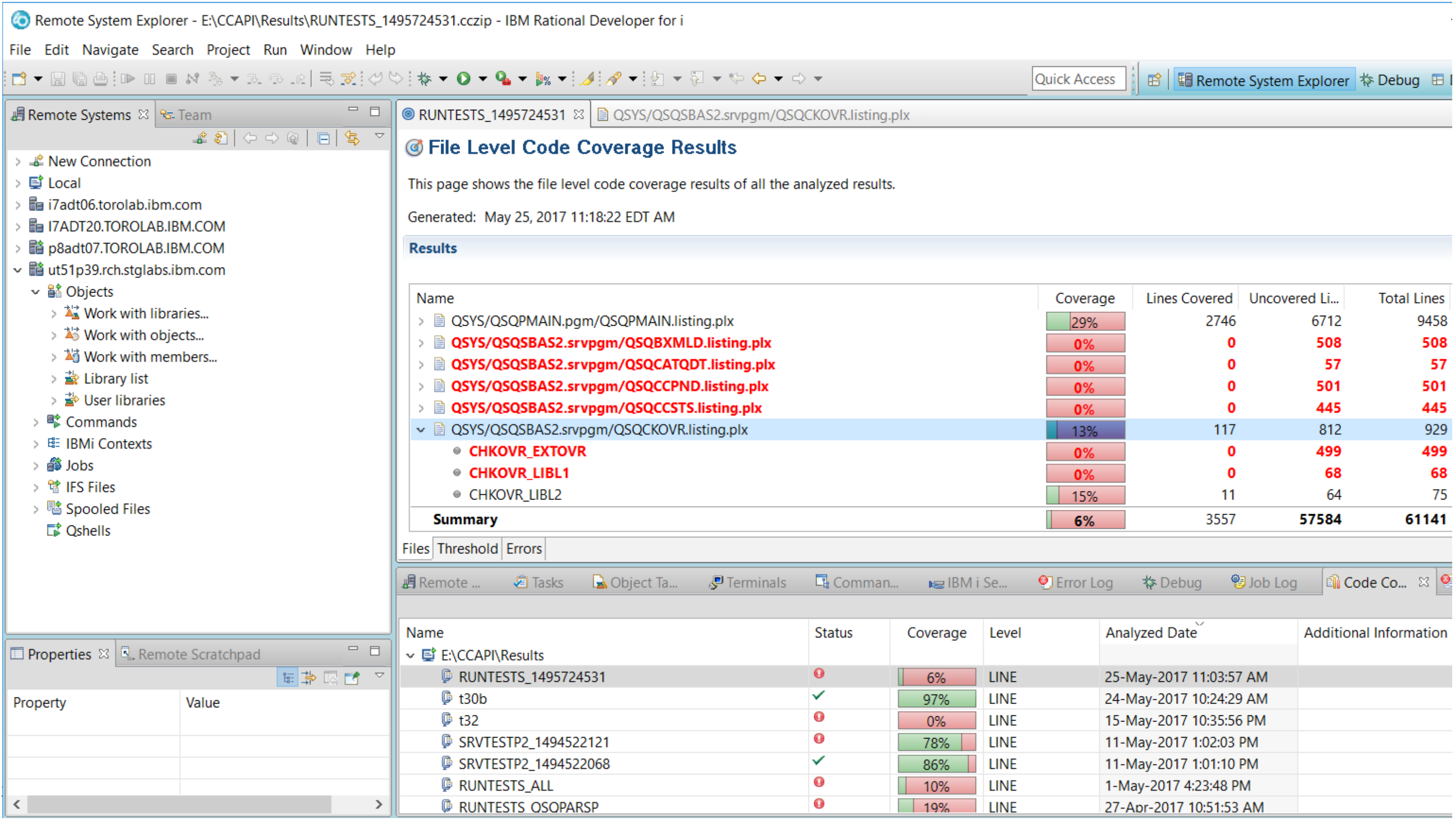
Connection: V7R1 Toronto

Analyzing headless code coverage results on RDi

- The results will be listed and can be opened as Workbench Reports, Merged and compared.
- Note that the HTML and PDF reports are not yet available.
- Alternatively, the IFS directory could be mounted or the *.cczip files could be copied to a local directory and have Local Result Location
  - (this is required for 9.5.1.2)



RDi –- Code Coverage report and Code Coverage Results view



- New Connection
- Local
- i7adt06.torolab.ibm.com
- I7ADT20.TOROLAB.IBM.COM
- p8adt07.TOROLAB.IBM.COM
- ut51p39.rch.stglabs.ibm.com
  - Objects
    - Work with libraries...
    - Work with objects...
    - Work with members...
    - Library list
    - User libraries
  - Commands
  - IBMi Contexts
  - Jobs
  - IFS Files
  - Spooled Files
  - Qshells

File Level Code Coverage Results

This page shows the file level code coverage results of all the analyzed results.

Generated: May 25, 2017 11:18:22 EDT AM

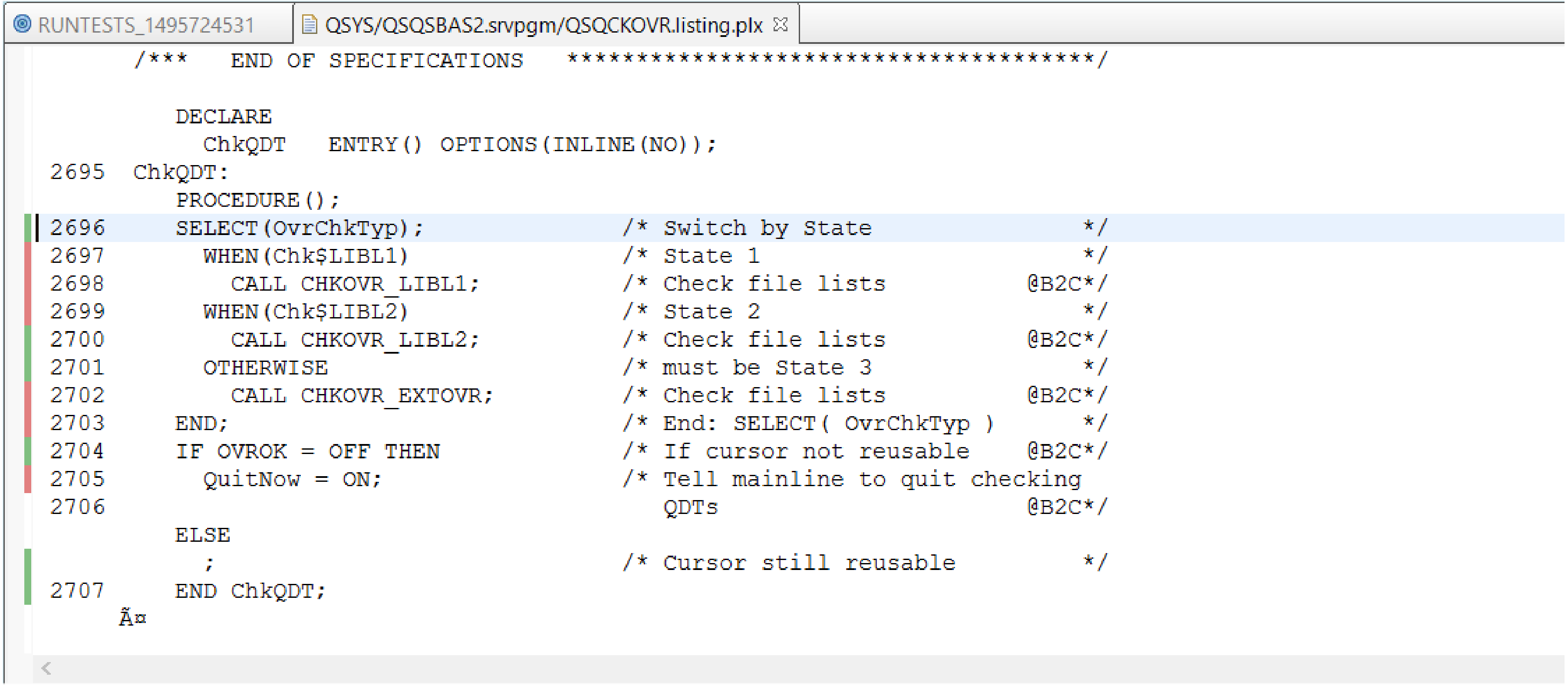
Results

Name	Coverage	Lines Covered	Uncovered Li...	Total Lines
QSYS/QSQPMAIN.pgm/QSQPMAIN.listing.plx	29%	2746	6712	9458
QSYS/QSQSBAS2.srvpgm/QSQBXMLD.listing.plx	**0%**	**0**	**508**	**508**
QSYS/QSQSBAS2.srvpgm/QSQCATQDT.listing.plx	**0%**	**0**	**57**	**57**
QSYS/QSQSBAS2.srvpgm/QSQCCPND.listing.plx	**0%**	**0**	**501**	**501**
QSYS/QSQSBAS2.srvpgm/QSQCCSTS.listing.plx	**0%**	**0**	**445**	**445**
QSYS/QSQSBAS2.srvpgm/QSQCKOVR.listing.plx	13%	117	812	929
CHKOVR_EXTOVR	**0%**	**0**	**499**	**499**
CHKOVR_LIBL1	**0%**	**0**	**68**	**68**
CHKOVR_LIBL2	15%	11	64	75
Summary	**6%**	3557	**57584**	**61141**

Name	Status	Coverage	Level	Analyzed Date	Additional Information
E:\CCAPI\Results					
RUNTESTS_1495724531	❗	6%	LINE	25-May-2017 11:03:57 AM	
t30b	✓	97%	LINE	24-May-2017 10:24:29 AM	
t32	❗	0%	LINE	15-May-2017 10:35:56 PM	
SRVTESTP2_1494522121	❗	78%	LINE	11-May-2017 1:02:03 PM	
SRVTESTP2_1494522068	✓	86%	LINE	11-May-2017 1:01:10 PM	
RUNTESTS_ALL	❗	10%	LINE	1-May-2017 4:23:48 PM	
RUNTESTS_QSQPARSP	❗	19%	LINE	27-Apr-2017 10:51:53 AM	

Property	Value

RDi –- Code Coverage Annotation in source editor



```
       /***   END OF SPECIFICATIONS   **************************************/

          DECLARE
            ChkQDT    ENTRY() OPTIONS(INLINE(NO));
 2695  ChkQDT:
          PROCEDURE();
 2696     SELECT(OvrChkTyp);                 /* Switch by State             */
 2697       WHEN(Chk$LIBL1)                  /* State 1                     */
 2698         CALL CHKOVR_LIBL1;             /* Check file lists        @B2C*/
 2699       WHEN(Chk$LIBL2)                  /* State 2                     */
 2700         CALL CHKOVR_LIBL2;             /* Check file lists        @B2C*/
 2701       OTHERWISE                        /* must be State 3             */
 2702         CALL CHKOVR_EXTOVR;            /* Check file lists        @B2C*/
 2703     END;                               /* End: SELECT( OvrChkTyp )    */
 2704     IF OVROK = OFF THEN                /* If cursor not reusable  @B2C*/
 2705       QuitNow = ON;                    /* Tell mainline to quit checking
 2706                                           QDTs                    @B2C*/
          ELSE
            ;                                /* Cursor still reusable       */
 2707     END ChkQDT;
      Ã¤
```

Vision

- Can raise the maturity of your software development process
  - Get the code coverage on your regular build/test on the host
  - Detect drops in coverage on each build
  - Mandate a minimum threshold for your tests
  - Integrate with automated test solution
  - Capture analysis with only dependence on a command shipped with IBM i
  - Analyze the results with the full power of RDi

Where to learn about Code Coverage

- Learn More about Code Coverage
  - Code Coverage Product Documentation
  - Code Coverage article on DeveloperWorks
  - Demo of interactive Code Coverage automated with Rational Function Tester
  - IBM Application Delivery Intelligence based on Code Coverage available for z
- Test Automation
  - RPGUnit integration for RDi – free, applicable for batch programs only
  - ARCAD Verifier product description
  - ARCAD Verifier Demo
  - Original Software -TestDrive