# Rational Developer for i Sandbox for IBM i Lab Exercise Workbook

## Rational Developer for i

## Lab 10 – Working with Code Coverage

This lab covers launching code coverage, comparing and sharing code coverage reports.

Version 4, January 2021

The most up to date version of this document can be found on Rational Developer for i - Hands-On Labs at http://ibm.biz/rdi_labs.

# Lab 10 – Working with Code Coverage

## *Overview*

This tutorial teaches you how to configure and generate code coverage reports for a test run. You then see by views, html reports, and editor annotations exactly which lines of the programs have been executed or not executed by a particular scenario. You will become familiar with tools that help you analyze the effectiveness of your automated or manual tests.

## *Learning objectives*

- Launch code coverage on a selected program or program service
- View and analyze coverage statistics
- Create a comparison report on selected code coverage runs
- Merge multiple code coverage results into a single code coverage report
- Import and export code coverage results

## *Skill level and prerequisites*

Introductory.

| | |
|---|---|
| ⚠ | **Important!**<br><br>You should complete **RDi Lab01** 'Getting started' before you work on this lab.  Lab01 contains the following information and instructions:<br>    • Which IBM i server to connect to<br>    • Which User ID to use<br>    • How to start RDi, create a connection and connect<br>    • How to setup the correct library list for this lab |

Knowledge of basic Microsoft Windows operations such as working with the desktop, mouse operations such as opening folders and drag-and-drop is assumed.  It will also be helpful if you understand DDS and ILE RPG.

## Conventions used in this workbook

| **Bold font** | is used to highlight user interface controls |
| `Mono-spaced font` | is used for user input text and code blocks |
| *Italic font* | is used for variable names and glossary terms |

The following icons are also used to identify categories of information:

| Icon | Purpose | Explanation |
|---|---|---|
| ⚠️ | Important! | This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive. |
| *i* | Information | This symbol indicates information that might not be necessary to complete a step, but is helpful or good to know. |
| 🚧 | Trouble-shooting | This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information. |

## Client System requirements

The labs require IBM Rational Developer for IBM i (RDi) to be installed on your workstation. If you do not yet have this, you can download it for free from http://ibm.biz/rdi_trial. As of version 9.5, RDi includes a built-in emulator so you will not need any additional software. If you are using a previous version, then any 5250 emulator will work.  The IBM i Access Client Solutions contains a best of breed emulator that is freely available to those who have an IBM i that is V6R1 or later.

## Host System requirements

The easiest way to ensure you have everything you need, is to use the EM Sandbox demonstration IBM i server that is set up and ready to use with these lab exercises.

> **Tip**
> If this is **not** an instructor led class with PC's provided, you may need to install and setup the IBM software on your PC first.
>
> 120 day Trial of Rational Developer for i can be downloaded here:
> http://ibm.biz/rdi_trial
> Page to request userid for IBM i demonstration system:
> http://ibm.biz/rdi_labs_getuserid

# 1   Launching code coverage

This module teaches you how to launch code coverage using a service entry breakpoint and how to edit configuration settings.

## 1.1   Setting service entry point

Code coverage can be launched on any program or service program that can be debugged. In this lesson, you will be working with the ILE RPG program PAYROLLFFF.

> **Note:**
> PAYROLLFFF is the same RPG program as PAYROLLG but in fully free-form. Fully free-form RPG source has been supported by Rational Developer for i since version 9.5. The purpose of using this program is to introduce you fully free-form RPG while teaching you about code coverage.
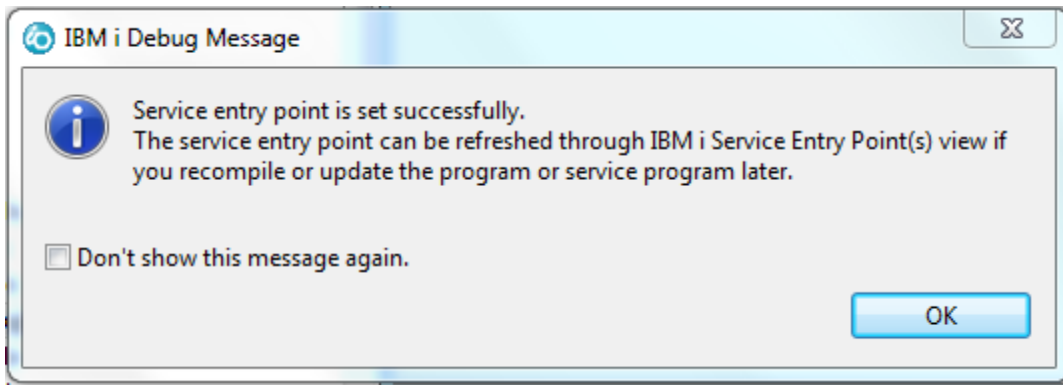
Let's begin with setting a service entry point to code coverage mode.

Make sure your assigned team library RSELABxx is part of the library list (right click on Library list and add if needed) and you are in the Remote Systems Explorer perspective (Window / Open Perspective...):
__1.    In the Remote Systems view expand the **Library list** filter, if it isn't expanded already.
__2.    Expand library RSELABxx, if it isn't expanded already.
__3.    Right-click program **PAYROLLFFF** in library RSELABxx.
__4.    Click **Debug or Code Coverage (Service Entry) > Set Service Entry Point** on the pop-up menu to set a service entry point.
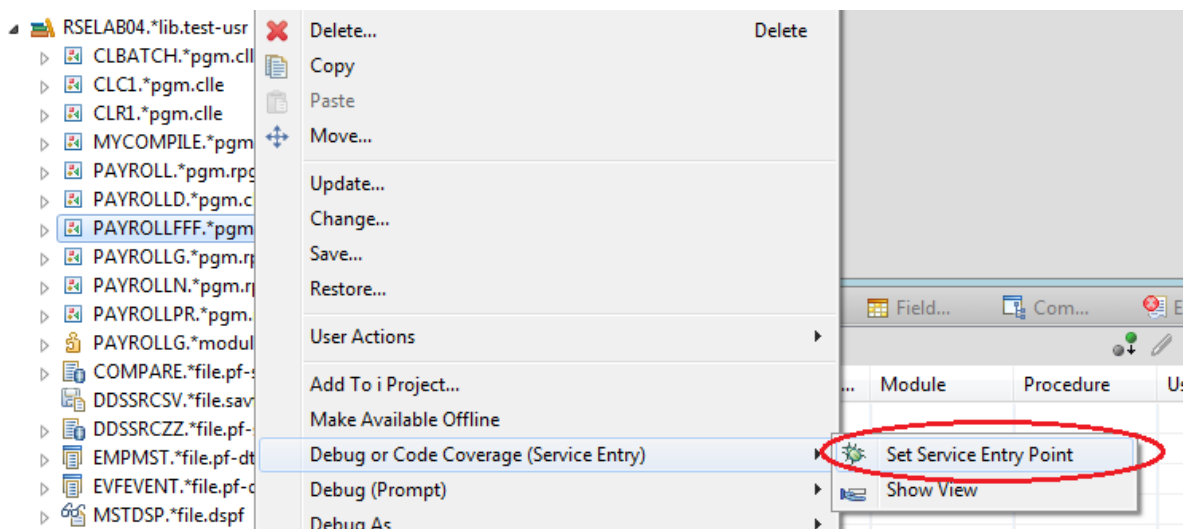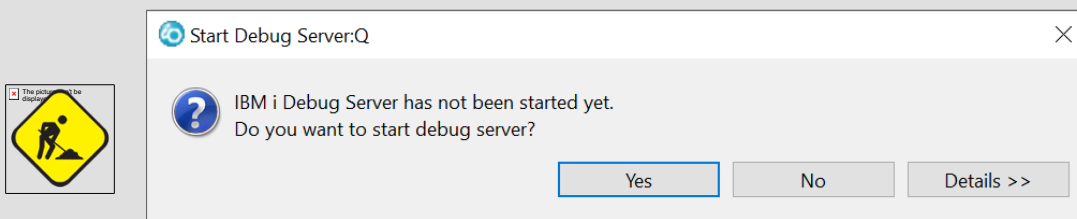
A message displays indicating the service entry point was successfully set.

**IBM i Debug Message**

Service entry point is set successfully.
The service entry point can be refreshed through IBM i Service Entry Point(s) view if you recompile or update the program or service program later.

☐ Don't show this message again.

OK

## Troubleshooting
If you get this error message instead,

**Start Debug Server:Q**

IBM i Debug Server has not been started yet.
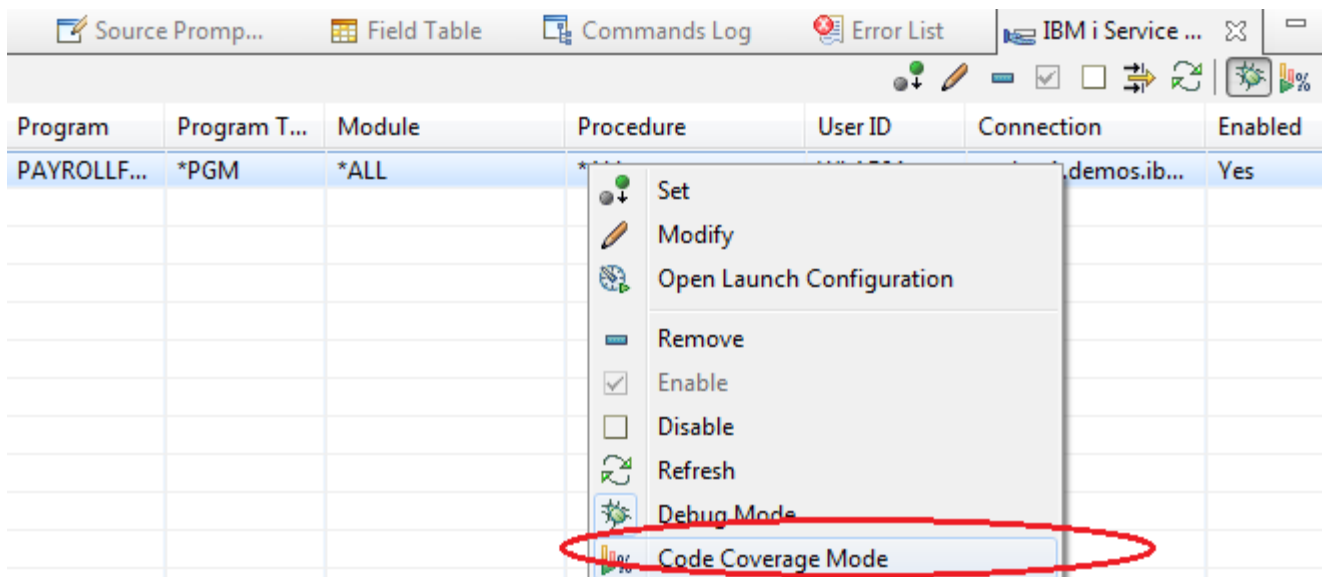Do you want to start debug server?

Yes    No    Details >>

indicating that the Debug Server has not been started yet. Click Yes button to start the Debug server.

The Service Entry Points view is automatically added to the stacked views below the edit view. It lists all service entry points set in this workbench instance. You use this view to delete, activate, de-activate, modify and refresh service entry points.

| Source Prompter | Field Table | Commands Log | Error List | Job Status | IBM i Service Entry Points ⊠ | Code |
|---|---|---|---|---|---|---|
| Program | Program Type | Module | | Procedure | Condition | |
| PAYROLLFFF | *PGM | *ALL | | *ALL | | |

__5.    Right-click the service entry point and select **Code Coverage Mode**. Alternatively, select the Code Coverage toolbar button in the view .

| Source Promp... | Field Table | Commands Log | Error List | IBM i Service ... ⊠ |
|---|---|---|---|---|
| Program | Program T... | Module | Procedure | User ID | Connection | Enabled |
| PAYROLLF... | *PGM | *ALL | | | | .demos.ib... | Yes |

- Set
- Modify
- Open Launch Configuration
- Remove
- Enable
- Disable
- Refresh
- Debug Mode
- Code Coverage Mode

**Note:**
The default mode for service entry points is debug mode. If you restart the IDE, the mode will be reset to debug mode.

IBM Rational Developer for i Sandbox – RDi Lab 10 – Code Coverage
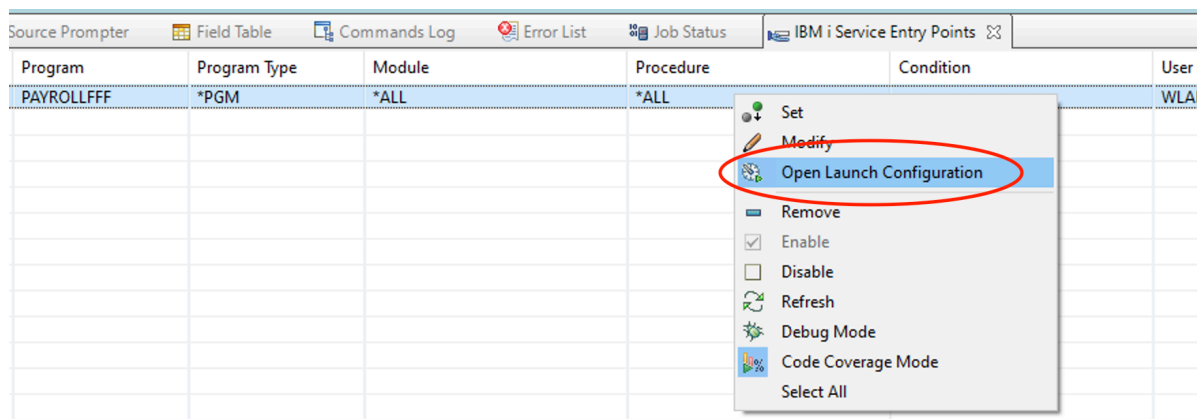
## 1.2  Editing code coverage configurations

> **Note:**
> This section is for your information only, continue to the next section to resume the exercises.

For this exercise you don't have to change the default configurations but here are the steps to follow when you work on modifying the configurations in your own environment.

__1.  In the IBM i Service Entry Points view, right click the service entry point and select **Open Launch Configuration**.

The Edit Configuration window opens.
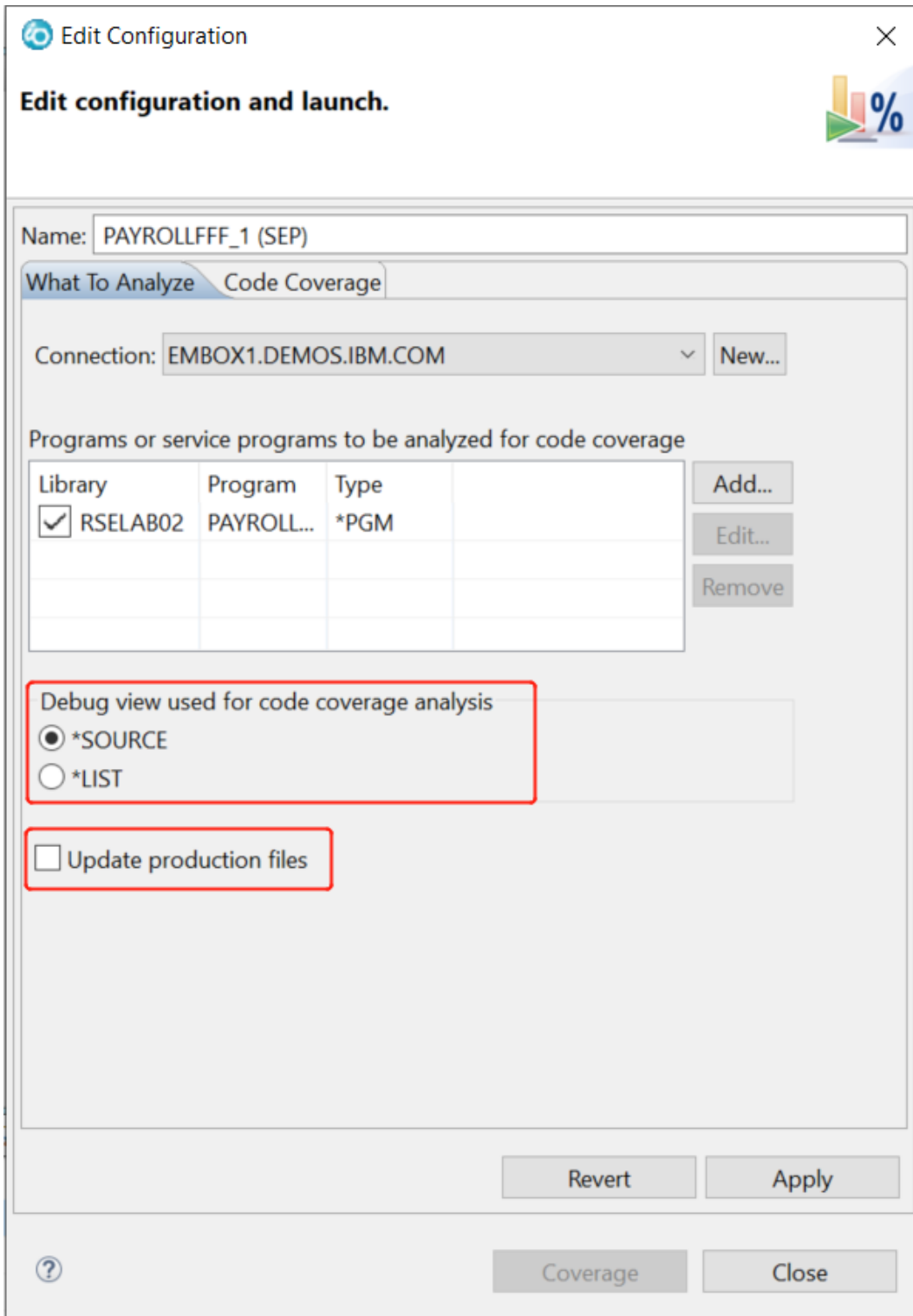
In the **What To Analyze** tab, you can select the debug view to use for code coverage analysis. The *SOURCE view will only provide coverage for the statements that appear in the main source member. If some executable statements are included by /COPY or /INCLUDE directives, coverage for those statements can only be detected with the *LIST view. In order to use a debug view for analysis, the program needs to be compiled with that DBGVIEW parameter.

To update production files, click the checkbox at the bottom. This will allow the application under analysis to update database records in production libraries.

Now let's take a look at the Code Coverage tab.



Here you can specify the Code Coverage Level. By default, code coverage is run at the line level. You may select to track the coverage at the higher level of procedures.

In the Report section below, you are given the option to generate PDF reports from the initial launch. However, you can always generate them later.

__2.    Click **Apply** to save the configuration.
__3.    Click **Close**.

## 1.3  Running code coverage

Now we will execute a manual test of a sample application and then analyze how much of the source code was covered by our test.

__1.    Start a 5250-emulation session and sign-on to the server.

__2.    Enter the command **ADDLIBLE RSELABxx** to add the library RSELABxx to the library list.

```
Selection or command
===> ADDLIBLE RSELAB08_

F3=Exit    F4=Prompt    F9=Retrieve
```

__3.    Enter **CALL RSELABxx/PAYROLLFFF** to invoke the program.

```
Selection or command
===> CALL RSELAB08/PAYROLLFFF_

F3=Exit    F4=Prompt    F9=Retrieve    F12=Cancel
F23=Set initial menu
Library RSELAB08 added to library list.
MA   a                      MW
```

You will see the first screen of the PAYROLLFFF program. Notice that the status bar at the bottom of the IDE displays the message Running code coverage for PAYROLLFFF.

Running code coverag...AYROLLFFF ...
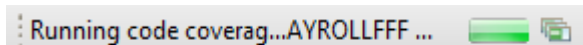
__4.    Type **x** beside **Employee Master Maintenance**.
__5.    Press **Enter**.

```
   PRG01                          Time Reporting System                    2/22/16
                                  Maintenance Selection                    10:21:54


              Enter an X beside the application you want to maintain



                          X  Employee Master Maintenance
                          _  Project Master Maintenance
                          _  Reason Code Master Maintenance








          F3-End of Job

MA     a                                                                   09/025
```

__6.     Type **1001** for the Employee Number.
__7.     Type **A** for the Action Code to add employee 1001.
__8.     Press **Enter**.

```
Employee Number    1001____

Action Code        A     A-Add
                         C-Change
                         D-Delete
```

__9.     Type any information you like about the employee.
__10.    Press **Enter**.

```
Number   001001    Name    NAME1_____

        Category           A
        Department         ABC___
        Location           LOC1_____
        USRID              UID_____
        Normal week hours  400_    (eg. 40.0 enter 400)

                   Time Reporting History

                             Current   Year To    Prior
                             Month     Date       Year
        Project Related      00000     0000000    0000000
        Non Project Related  00000     0000000    0000000
```

__11.    Type **1001** for the Employee Number.
__12.    Type **C** for the Action Code to change the information of employee 1001.
__13.    Press **Enter**.
__14.    Change some information about the employee.
__15.    Press **Enter**.

```
Number   001001    Name    NAME1_____

        Category           B
        Department         CBA___
        Location           LOC1_____
        USRID              1001_____
        Normal week hours  400_    (eg. 40.0 enter 400)

                   Time Reporting History

                             Current   Year To    Prior
                             Month     Date       Year
        Project Related      00000     0000000    0000000
        Non Project Related  00000     0000000    0000000
```

You have added a new employee and updated the information. Now delete the employee and then add the deleted employee back.

__16.    Type **1001** for the Employee Number.

__17.    Type **D** for the Action Code to delete employee 1001**.**

__18.    Press **Enter**.

__19.    The information of employee 1001 is shown. Press **Enter** again to confirm deletion.

__20.    Type **1001** for the Employee Number.

__21.    Type **A** for the Action Code again to add employee 1001 back**.**

__22.    The information of the deleted employee is shown. Press **Enter** again.


Now let's exit the program.

__23.    Press **F3** to end the job.


After the program is exited, a workbench report is opened and an item summarizing the coverage is added to the Code Coverage Results view under the Compiled Code Coverage Workspace Results location.

## Code Coverage Report

Code coverage report for 'PAYROLLFFF_2019_05_11_122226_0131', analyzed 11-May-2019 12:24:28 PM

Export

Off ⚪— On   Show below :   80   %   Refresh          ?   | Files | Modules |

| Name | Coverage | Lines Covered | Uncovered Lines | Total Lines |
|---|---|---|---|---|
| > PAYROLLFFF.RPGLE | 49% | 120 | 123 | 243 |
| Summary (Elapsed time: 123.237 sec) | 49% | 120 | 123 | 243 |

Remote System Details    Tasks    Object Table    Commands Log    Terminals    IBM i Service Entry Points    Code Coverage Results

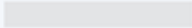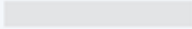| Name | | Status | Coverage | Level | Analyzed Date | Additic |
|---|---|---|---|---|---|---|
| ∨ ▦ Compiled Code Coverage Workspace Results | | | | | | |
| | PAYROLLFFF_2019_05_11_122226_0131 | ⊘ | 49% | Line | 11-May-2019 12:22:26 PM | |
| ▥ Java Code Coverage Workspace Results | | | | | | |
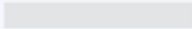
# 2 Analyzing code coverage reports

This module teaches you how to analyze code coverage reports with the code coverage results view. You will learn how to view coverage results in an editor, import and export results, generate HTML or PDF reports, compare results and merge results.

## 2.1 Viewing code coverage results

You can view your code coverage results in either a Workbench, File, PDF report or within the editor. By default, Workbench reports are generated.

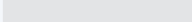__1.    In the Workbench code coverage report, expand the report tree.
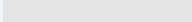
| Name | Coverage | Lines Covered | Uncovered Lines | Total Lines |
|---|---|---|---|---|
| PAYROLLFFF.RPGLE | 49% | 120 | 123 | 243 |
| PAYROLLFFF | 57% | 4 | 3 | 7 |
| MAIN | 79% | 22 | 6 | 28 |
| REASONMAINTENANCE | 0% | 0 | 45 | 45 |
| EMPLOYEEMAINTENANCE | 91% | 42 | 4 | 46 |
| PROJECTMAINTENANCE | 0% | 0 | 51 | 51 |
| ISADDNEWRECORDREQUEST | 100% | 3 | 0 | 3 |
| ISADDPREVIOUSLYDELETEDRECORDREQUEST | 100% | 3 | 0 | 3 |
| VALIDATEFILETOMAINTAINSELECTION | 77% | 10 | 3 | 13 |
| VALIDATEACTIONCODE | 87% | 13 | 2 | 15 |
| VALIDATEADD | 89% | 8 | 1 | 9 |
| VALIDATECHANGE | 78% | 7 | 2 | 9 |
| VALIDATEDELETE | 80% | 8 | 2 | 10 |
| DISPLAYERROR | 0% | 0 | 4 | 4 |

The statistics for programs, modules and procedures are shown.

In Section 1.3, the steps to generate this coverage report do not cover any error scenarios. As a result, the DISPLAYERROR procedure here has 0% coverage rate (0 out of 4 total lines gets covered).

To view code coverage results in the editor:

__2.    Click on the **MAIN()** procedure to see why it only has 79% coverage (or why 6 of the 28 lines were not executed).

| Name | Coverage | | Lines Covered | Uncovered Lines | Total Lines |
|------|----------|---|---------------|-----------------|-------------|
| PAYROLLFFF.RPGLE | 49% | | 120 | 123 | 243 |
| PAYROLLFFF | 57% | | 4 | 3 | 7 |
| MAIN | 79% | | 22 | 6 | 28 |
| REASONMAINTENANCE | 0% | | 0 | 45 | 45 |
| EMPLOYEEMAINTENANCE | 91% | | 42 | 4 | 46 |
| PROJECTMAINTENANCE | 0% | | 0 | 51 | 51 |

__3.    If you are prompted with a Remote System Connection dialog, simply press **OK** and continue.

The source opens in the editor with decorations showing the code coverage for each line. The cursor is positioned to the MAIN procedure. By default, a green bar next to the line indicates that the line was covered; a red bar that it was not covered.

```
Line 64      Column 16    Replace                    Browse
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+.
Dcl-Proc MAIN;
  DoU *INKC;
     *IN60   = *OFF;
     EMESS   = *BLANK;
     EMPAPL = *BLANK;
     PRJAPL = *BLANK;
     RSNAPL = *BLANK;
     //
     //   Write the SELECT format to display until end of job requested,
     //
     DoU not *IN60;
        //   IF the general error indicator *IN60 is on (equal to 1), the
        //   program continues to loop
        Exfmt SELECT;
        If *INKC = '1'; // F3 = end of job
          Return;
        Else;
          ValidateFileToMaintainSelection();
        EndIf;
     EndDo;
  EndDo;
```

Now you can see by the editor annotations exactly which lines of the programs have been executed or not executed by a particular scenario.

In the MAIN procedure, lines 79 (shown in the screenshot above) and 109 were not covered, because we did not exit the program from the main screen.

__4.    Hold your mouse over the indicator to see the coverage specifics, as shown below.

```
Line 94      Column 1     Replace
...+....1....+....2....+....3....+....
//   next executable operation.
//
    Select;
      When EMPAPL = 'X';
        CallP EmployeeMaintenance();
//
        When PRJAPL = 'X';
          CallP ProjectMaintenance();
//
        When RSNAPL = 'X';
          ReasonMaintenance();
Lines 103 to 105: Not Covered
```

Since you have only chosen the Employee Master Maintenance routine in Section 1.3, the last four lines here (lines 101 – 105 in the source) were not covered.

__5.    Scroll down to lines 116 and 259. Verify that the **ReasonMaintenance** and **ProjectMaintenance** procedures were not covered for the same reason as above.

__6.    Go to EmployeeMaintenance line 212.

```
// Process keys
If *INKC;     //F3 = exit program
  Return;
Elseif *INKD; //F4 = return to main screen
  Return;
EndIf;
```

Notice that line 215 is the only uncovered line in the if-statement. This was because we pressed F3 instead of F4 to exit from the EmployeeMaintenance screen in Step 1.3.__23. .

> **Tip:**
>
> The editor annotations can be customized to show text symbols instead of colored bars:
>
>     i.   From the Code Coverage Results view, double-click your coverage result to open it in a File report.
>
>    ii.   Double-click the source file in the file level code coverage results.
>
>   iii.   Right-click on the editor annotations and select **Show text instead of colors for coverage results**.



```
X  Dcl-F MSTDSP      WORKSTN;
✓  Dcl-F EMPMST      Usage(*Update:*Delete:*Output) Keyed;
X  Dcl-F PRJMST      Usage(*Update:*Delete:*Output) Keyed;
X  Dcl-F RSNMST      Usage(*Update:*Delete:*Output) Keyed;
```

| | |
|---|---|
| | Add/Remove Breakpoint |
| | Reset coverage statistics |
| ✓ | Show text instead of colors for coverage results |

You have viewed your code coverage results within the editor.


## 2.2  Comparing code coverage results

You can compare results from two different reports to see how your coverage testing is progressing. At this moment we have only generated one code coverage report, so let's generate another one.

__1.     Switch to the 5250-emulation session.

__2.     Enter **CALL RSELABxx/PAYROLLFFF** to invoke the program.

__3.     Type **x** beside **Employee Master Maintenance**.

__4.     Press **Enter**.

__5.     Type **1002** for the Employee Number.

__6.     Type **D** for the Action Code to delete employee 1002**.**

__7.     Press **Enter**.

Since you have never added employee 1002, an error message is displayed at the bottom of the screen.

__8.     Press **F4** to go back to **Maintenance Selection**.
__9.     Type **x** beside **Project Master Maintenance**.
__10.    Press **Enter**.



__11.    Type **101** for the Project Code.
__12.    Type **A** for the Action Code to add project 101.
__13.    Press **Enter**.

\_\_14.    Type any information you like about the project.
\_\_15.    Press **Enter**.

```
                    Project Code   101

      Description   MYPROC_____

          Responsibility              ABC_____
          Project Start Date          010101      (MMDDYY)
          Project Estimated End Date  020101      (MMDDYY)
          Project Completion Date     020101      (MMDDYY)
          Project Estimated Hours     100_____
                                        _
                      Project History

                  Current    Year To      Prior
                   Month      Date         Year
                  0000000   000000000   000000000
```

___16.    Type **101** for the Project Code.
___17.    Type **C** for the Action Code to change the information of project 101.
___18.    Press **Enter**.
___19.    Change some information about the project.
___20.    Press **Enter**.
___21.    Type **101** for the Project Code.
___22.    Type **D** for the Action Code to delete project 101.
___23.    Press **Enter**.
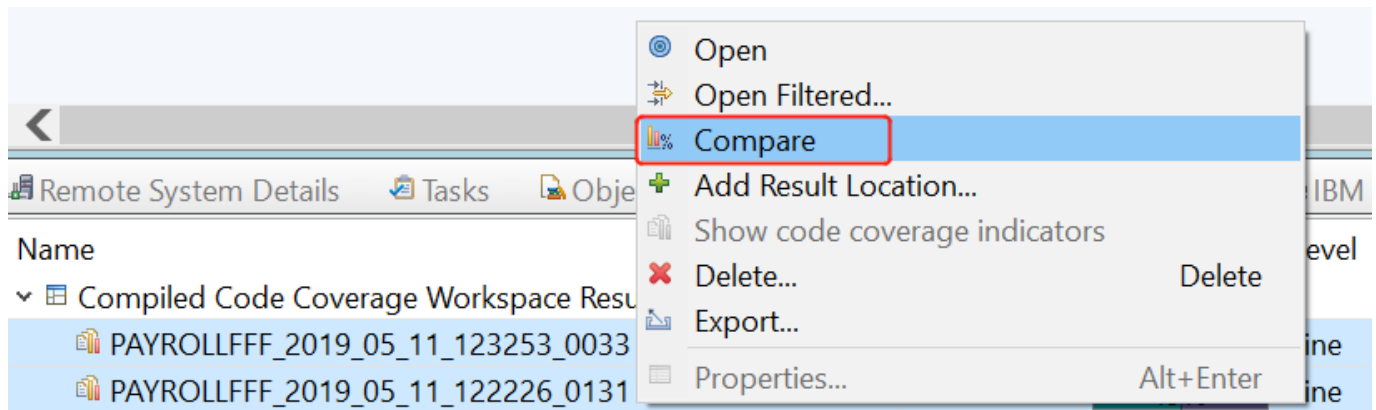___24.    Press **Enter** again to confirm deletion.

          Now exit the program.
___25.    Press **F3**.
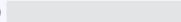
          You have created a second code coverage result.
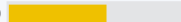
          Now let's compare these coverage results.
___26.    In the Code Coverage Results view, select your two results.
___27.    Right click and select **Compare**.

A comparison report opens in the workbench.

__28.	Expand the report tree.

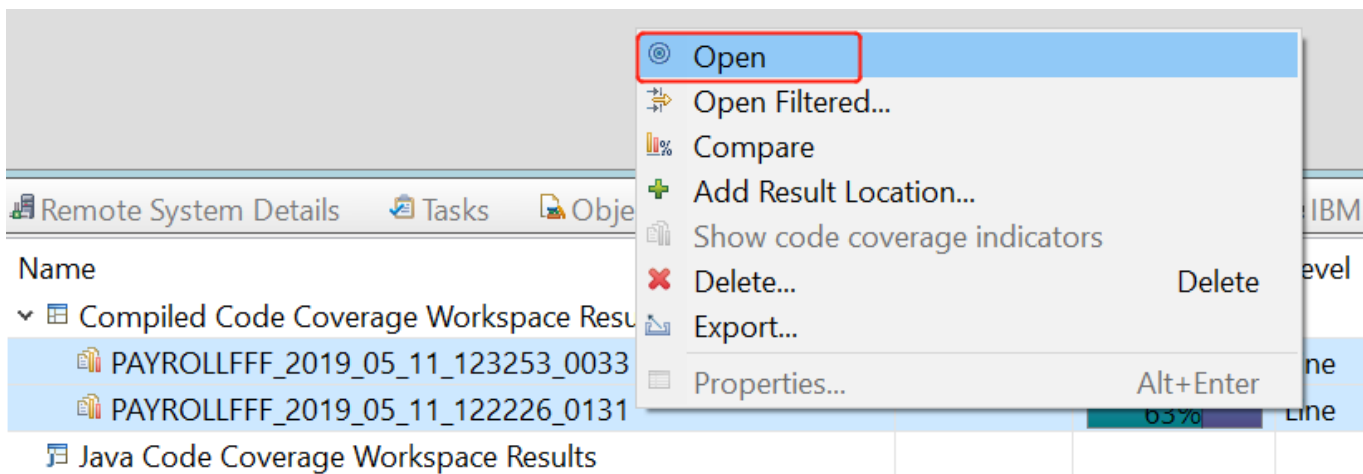| Name | Coverage | Lines Covered | Uncovered Lines | Total Lines |
|------|----------|---------------|-----------------|-------------|
| PAYROLLFFF.RPGLE | 63% (+14) ↑ | 120 → 153 | 123 → 90 | 243 |
| PAYROLLFFF | 57% (0) ○ | 4 | 3 | 7 |
| MAIN | 89% (+10) ↑ | 22 → 25 | 6 → 3 | 28 |
| REASONMAINTENANCE | 0% (0) ○ | 0 | 45 | 45 |
| EMPLOYEEMAINTENANCE | 50% (-41) ↓ | 42 → 23 | 4 → 23 | 46 |
| PROJECTMAINTENANCE | 88% (+88) ↑ | 0 → 45 | 51 → 6 | 51 |
| ISADDNEWRECORDREQUEST | 100% (0) ○ | 3 | 0 | 3 |
| ISADDPREVIOUSLYDELETEDRECORDREQUEST | 100% (0) ○ | 3 | 0 | 3 |
| VALIDATEFILETOMAINTAINSELECTION | 77% (0) ○ | 10 | 3 | 13 |
| VALIDATEACTIONCODE | 87% (0) ○ | 13 | 2 | 15 |
| VALIDATEADD | 78% (-11) ↓ | 8 → 7 | 1 → 2 | 9 |
| VALIDATECHANGE | 78% (0) ○ | 7 | 2 | 9 |
| VALIDATEDELETE | 90% (+10) ↑ | 8 → 9 | 2 → 1 | 10 |
| DISPLAYERROR | 100% (+100) ↑ | 0 → 4 | 4 → 0 | 4 |

The comparison report displays the coverage differences between the two selected code coverage runs. Here the 50% number on the EMPLOYEEMAINTENANCE procedure is the coverage result for the newer session. The number "-41" inside the bracket is the delta between the newer session and the older session. This means the older session has a coverage rate of 91% on EMPLOYEEMAINTENANCE. The same interpretation applies to the annotations in the "Covered" and "Total" columns.

In our second test run above, the steps in Employee Master Maintenance only cover an error scenario which tries to delete a non-existed employee. This is what causes the -41% difference in EMPLOYEEMAINTENANCE and the 100% difference in DISPLAYERROR.

## *2.3 Merging code coverage results*

We have taken 2 separate paths through our application exploring 2 different major functions.  It is fairly common with test cases to cover distinct subsets of an application.  It would be useful to see the combined coverage of all the scenarios we have tested. You can combine two or more results as one file report. By merging you consolidate information from different launches. The following steps will guide you to merge the two results generated in previous sections.

__1.     In the Code Coverage Results view, select your results.

__2.     Right click and select **Open**.



The merged report opens in the workbench.

__3.  Expand the report tree.

## Code Coverage Merged Report

Code coverage report for 'merged_20190511_130242', analyzed 11-May-2019 1:02:42 PM

Off ⬤— On   Show below :  80  %   Refresh                       (?)   Files   Modules

| Name | Coverage | Lines Covered | Uncovered Lines | Total Lines | Messages |
|---|---|---|---|---|---|
| ☑ PAYROLLFFF.RPGLE | 72% | 175 | 68 | 243 | |
| PAYROLLFFF | 71% | 5 | 2 | 7 | |
| MAIN | 89% | 25 | 3 | 28 | |
| REASONMAINTENANCE | 0% | 0 | 45 | 45 | |
| EMPLOYEEMAINTENANCE | 93% | 43 | 3 | 46 | |
| PROJECTMAINTENANCE | 88% | 45 | 6 | 51 | |
| ISADDNEWRECORDREQUEST | 100% | 3 | 0 | 3 | |
| ISADDPREVIOUSLYDELETEDRECORDREQUEST | 100% | 3 | 0 | 3 | |
| VALIDATEFILETOMAINTAINSELECTION | 77% | 10 | 3 | 13 | |
| VALIDATEACTIONCODE | 87% | 13 | 2 | 15 | |
| VALIDATEADD | 89% | 8 | 1 | 9 | |
| VALIDATECHANGE | 78% | 7 | 2 | 9 | |
| VALIDATEDELETE | 90% | 9 | 1 | 10 | |
| DISPLAYERROR | 100% | 4 | 0 | 4 | |
| Summary (Elapsed time: 243.756 sec) | 72% | 175 | 68 | 243 | |

Note that the procedure REASONMAINTENANCE is highlighted in red. This means we have a hole in our test bucket and need more testing on it.

__4.  Now click on the selection button between **off** and **on** to make **on** selected.

__5.  Change **Threshold** to 90%.

__6.  Click **Refresh**.

## Code Coverage Merged Report

Code coverage report for 'merged_20190511_130242', analyzed 11-May-2019 1:02:42 PM

Off —⬤ On   Show below :  90  %   Refresh

__7.   Expand the report tree.

| Name | Coverage | Lines Covered | Uncovered Lines | Total Lines |
|------|----------|---------------|-----------------|-------------|
| PAYROLLFFF.RPGLE | 72% | 175 | 68 | 243 |
| PAYROLLFFF | 71% | 5 | 2 | 7 |
| MAIN | 89% | 25 | 3 | 28 |
| REASONMAINTENANCE | 0% | 0 | 45 | 45 |
| PROJECTMAINTENANCE | 88% | 45 | 6 | 51 |
| VALIDATEFILETOMAINTAINSELECTION | 77% | 10 | 3 | 13 |
| VALIDATEACTIONCODE | 87% | 13 | 2 | 15 |
| VALIDATEADD | 89% | 8 | 1 | 9 |
| VALIDATECHANGE | 78% | 7 | 2 | 9 |

You have updated the threshold report to show only the procedures with at most 90% coverage rate.

## 2.4  Export code coverage results to PDF

Export code coverage result report to a PDF file.

__1.   From the **Code Coverage Results** view, right click the result generated in Section 1.3.

__2.   Right click and select **Export**.
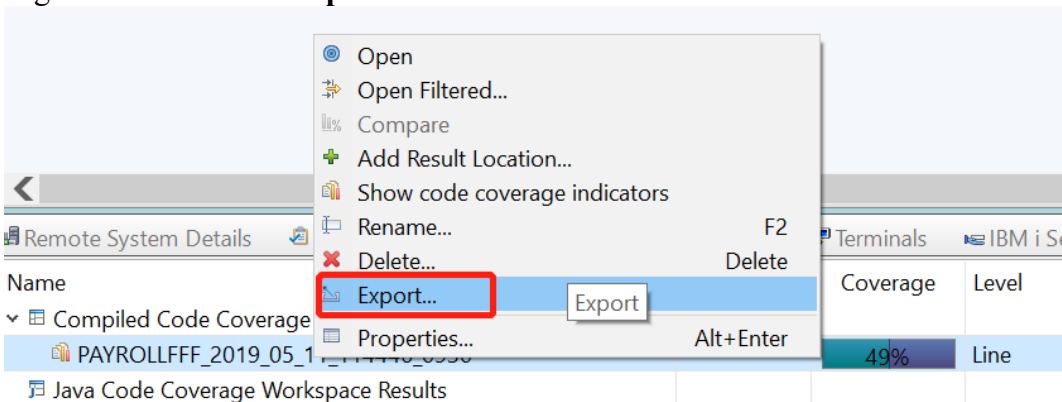


__3.   In the Export format and location section, select **PDF** as its export format.

4. Browse to select a desired location.
5. Change the properties of PDF
   You have the option to change properties of PDF. Click **Settings** button.

Export Format:

PDF

Destination folder:

C:\WLAB01    Browse...

Enter the result file name (leave it blank to use the default name):

ccresult_20190514_160934    .pdf

Filter:

☐ Regular expression

Settings

☐ Open PDF upon completion
☐ Add Location to Code Coverage Results View

Change the properties of PDF in this dialog.

PDF Properties ✕

Report Title:

Report Type: file

Hide Empty Module: true

Include Source: false

Line Marker: false

Threshold Percentage:

Message Type: E,W

Sort By: name

Sort Order: ascending

Reset to default

OK    Cancel

__6.	Click **OK,** then click **Finish**.

# File Overall Summary

**Report Info**

Report: ccresult_20190513_150918.pdf

Type: LINE

Generation Date: 13-May-2019 3:09:22 PM

Filter String:

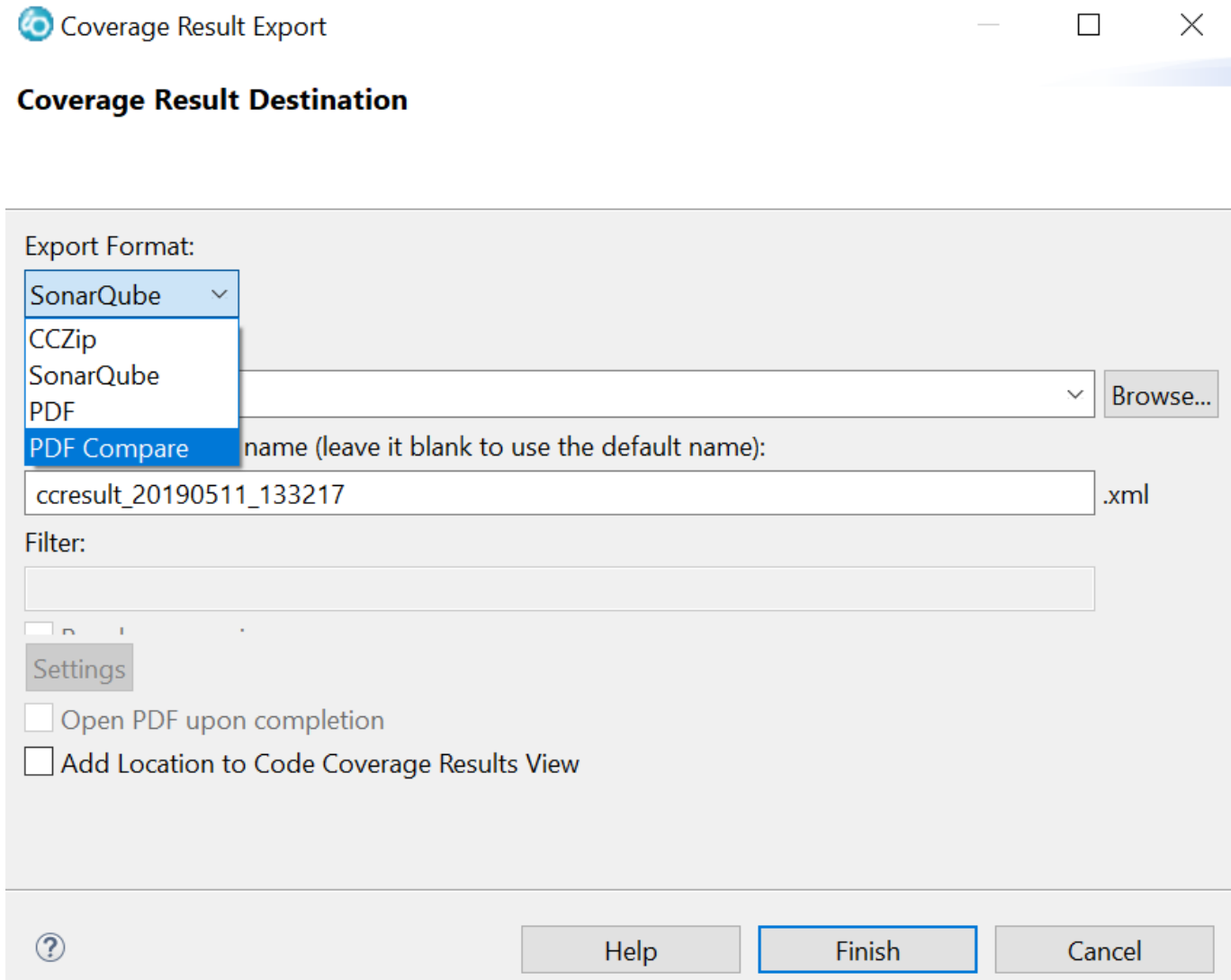| Overall Summary | Code Coverage Summary |
|---|---|
| Total Number of Files: 1 | File Coverage: 100% |
| Total Number of Functions: 13 | Function Coverage: 77% |
| Total Number of Lines: 243 | Line Coverage: 49% |
| Total Number of Statements: 243 | Statement Coverage: 49% |

A code coverage report in the PDF format has been generated and saved to the specified location.

Export code coverage results comparison report to a PDF file.

__7.     From the **Code Coverage Results** view, select your two results generated in section 2.2.

__8.     Select **Export**.
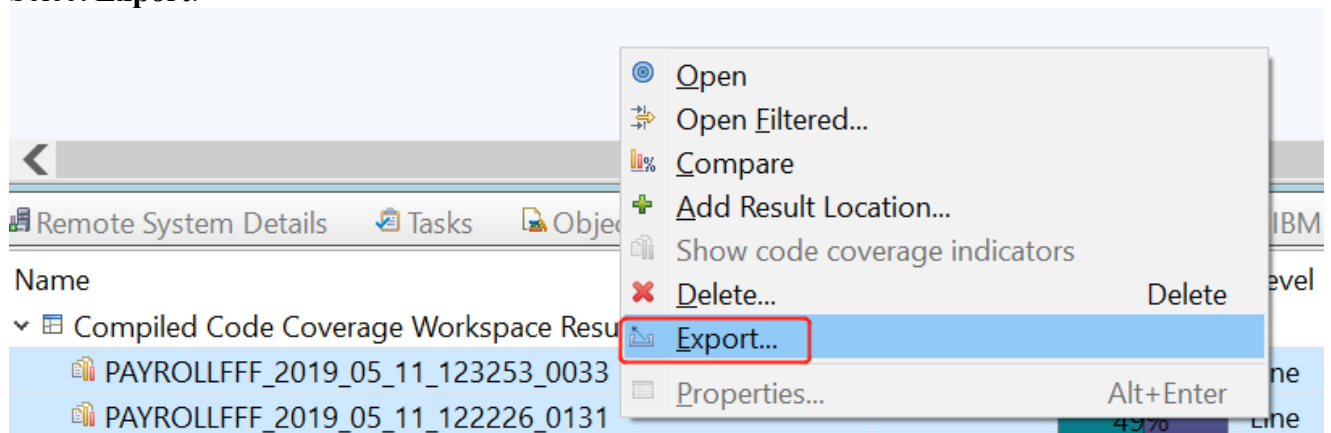


__9.     In the Export format and location section, select **PDF Compare** as its export format.

___10.    Browse to select a desired location.
___11.    Click **Finish**.

# File Comparison Overall Summary

**Report Info**

Report: ccresult_20190513_151324.pdf

Type: LINE

Generation Date: 13-May-2019 3:13:33 PM

Filter String:

| Overall Summary | Code Coverage Summary |
|---|---|
| Total Number of Files: 1 | File Coverage: 100% (0) ○ |
| Total Number of Functions: 13 | Function Coverage: 92% (43) ↑ |
| Total Number of Lines: 243 | Line Coverage: 63% (14) ↑ |
| Total Number of Statements: 243 | Statement Coverage: 63% (14) ↑ |

○ - No changes between two results   ↑ - Results going up

⊕ - This is a new item between two results   ↓ - Results going down
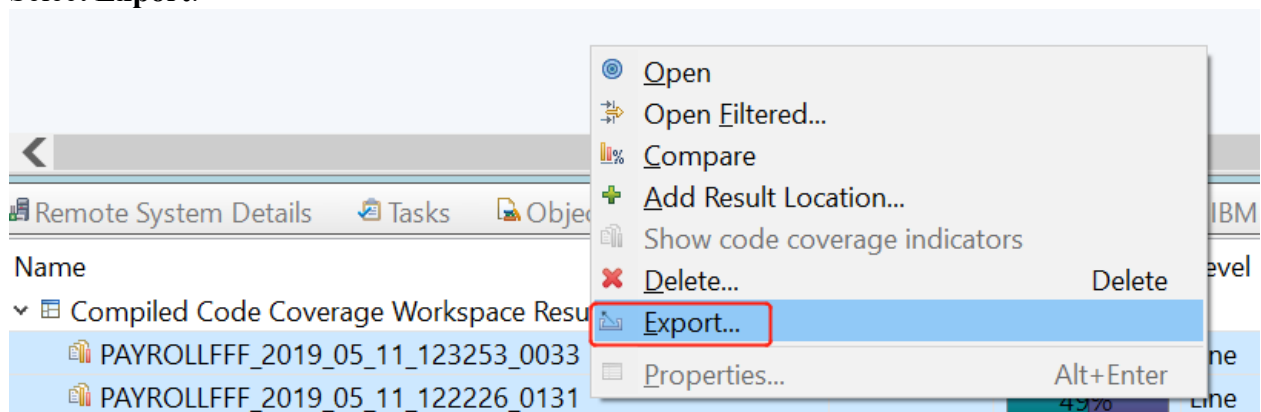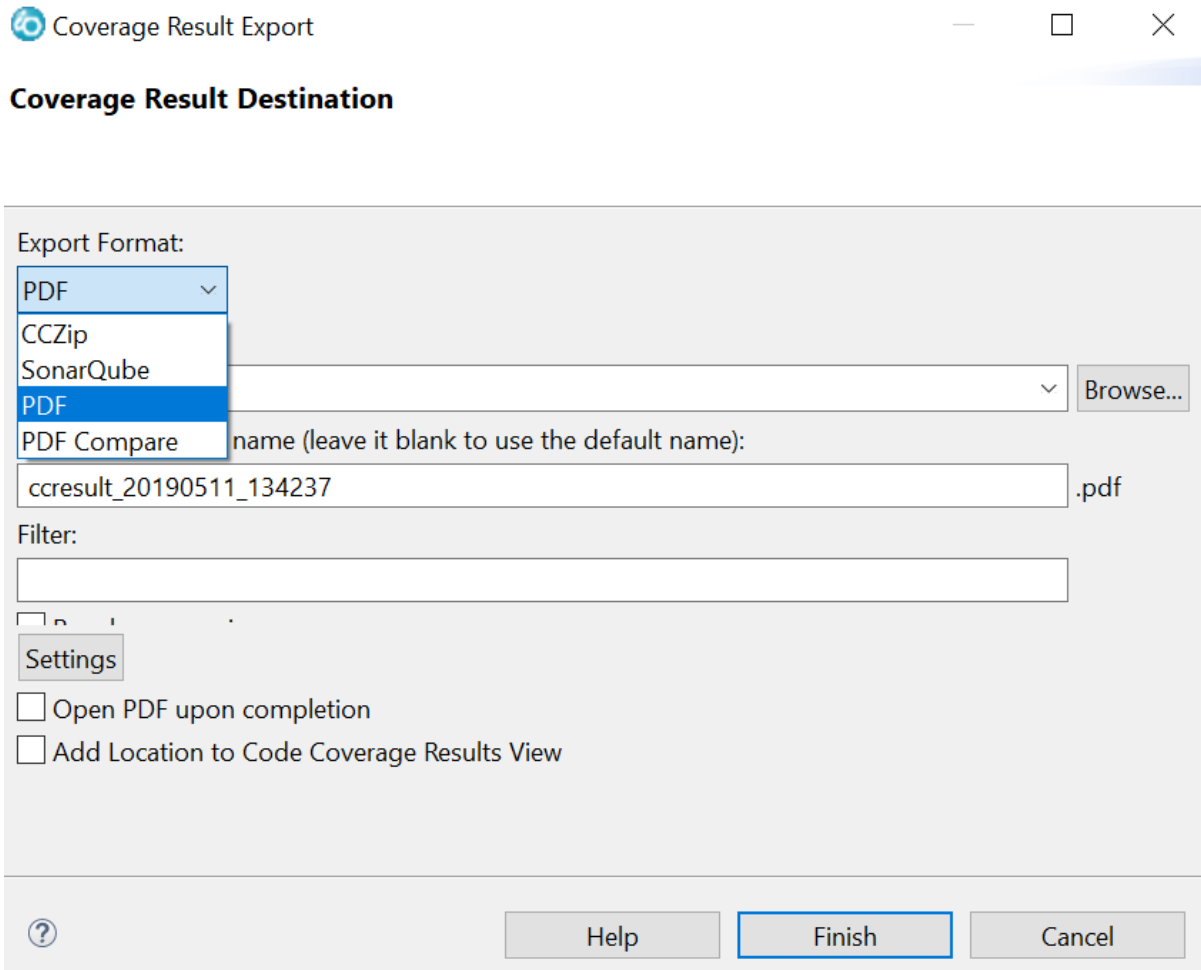
A code coverage comparison report in the PDF format has been generated and saved to the specified location.

Export merged code coverage results report to a PDF file.

__12.   From the **Code Coverage Results** view, select your two results generated in section 2.2.

__13.   Select **Export**.

```
                                        ◉  Open
                                        ⇥  Open Filtered...
◄                                       ▌% Compare
                                        ✚  Add Result Location...
🖥 Remote System Details   🗐 Tasks   📖 Obje    📇 Show code coverage indicators          IBM
Name                                    ✖  Delete...                        Delete    level
   ⌄ ⊞ Compiled Code Coverage Workspace Resu    📊 Export...
         📊 PAYROLLFFF_2019_05_11_123253_0033         Properties...          Alt+Enter   ne
         📊 PAYROLLFFF_2019_05_11_122226_0131                                          Line
```

__14.   In the Export format and location section, select **PDF** as its export format.

___15.    Browse to select a desired location.

___16.    Click **Finish**.

## File Overall Summary

A merged code coverage comparison report in the PDF format has been generated and saved to the specified location.

## 2.5 MODULES vs FILES

Modules would look at the compile programs and list which modules were bound into it. Whereas Files looks just at the source files independently. In this lab, there is only 1 module bound to the program, so there is no difference.

| Name | Coverage | Lines Covered | Uncovered Lines | Total Lines |
|---|---|---|---|---|
| *PGM RSELAB01/PAYROLLFFF | 49% | 120 | 123 | 243 |
| PAYROLLFFF | 49% | 120 | 123 | 243 |
| PAYROLLFFF.RPGLE | 49% | 120 | 123 | 243 |
| Summary (Elapsed time: 105.885 sec) | 49% | 120 | 123 | 243 |

## 2.6 Import and export code coverage results

Now you have tried most of the features in analyzing code coverage reports. It's time to make the results available outside of your workspace.

First, let's give the results a meaningful name.
__1.    In the Code Coverage Results view, right click on the result that you have created in Section 2.2.
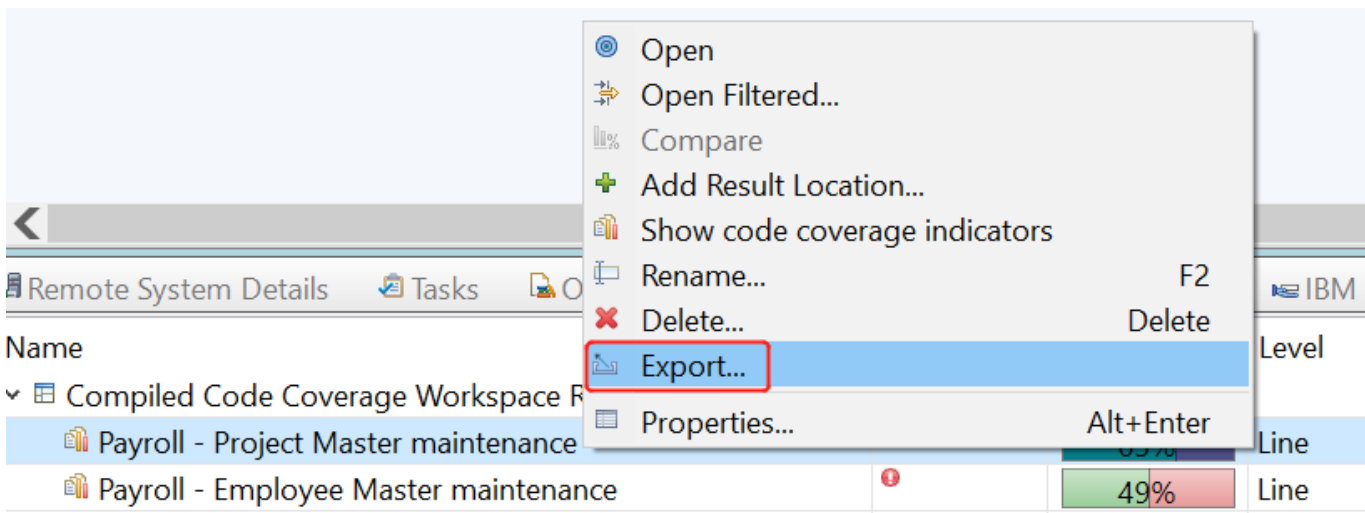__2.    Click **Rename** on the pop-up menu.
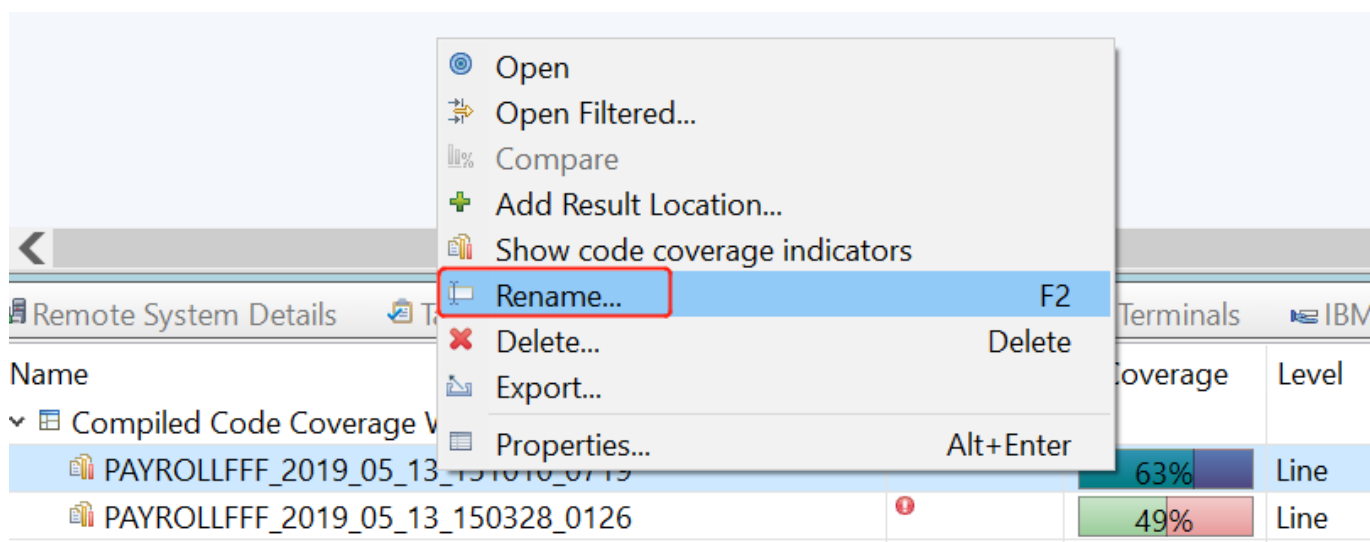
__3.     Enter **Payroll - Project Master maintenance** as the new result name.
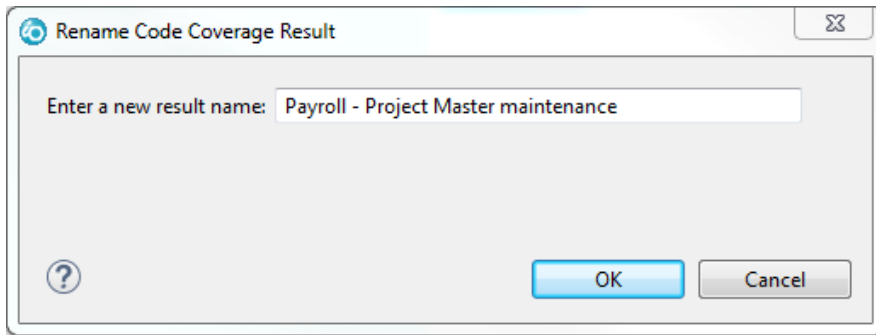


__4.     Click **OK**.

__5.     Rename the other coverage result to **Payroll – Employee Master maintenance**.

To export coverage results to a local directory:

__6.     Right click **Payroll - Project Master maintenance** and select **Export**.



__7.     Select **CCZip** as export format, select a desired destination and enter **payroll-project** as its file name.
         Click **Finish**.

You have exported a coverage result to a specified local directory.

Assume you need to share your exported results with some teammates. To place the files in a shared IFS folder:

__8.     In the Remote Systems view, expand the **Local** connection.
__9.     Expand to the location where you have saved the exported result file.
__10.    Right-click the file and select **Copy**.
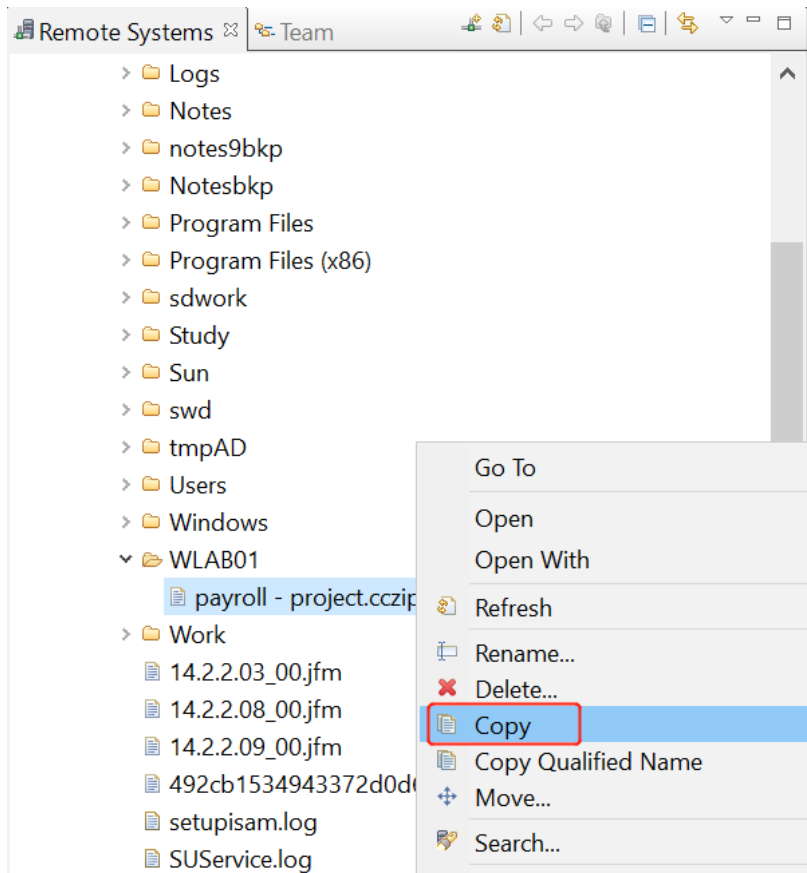
__11.    In the Remote Systems view, expand your remote connection **(common1.frankeni.com)**.

__12.    Expand **IFS Files**.

__13.    Right-click and **Paste** the file to **Home > WLABxx**. (You may want to create a filter for WLABxx to save time)
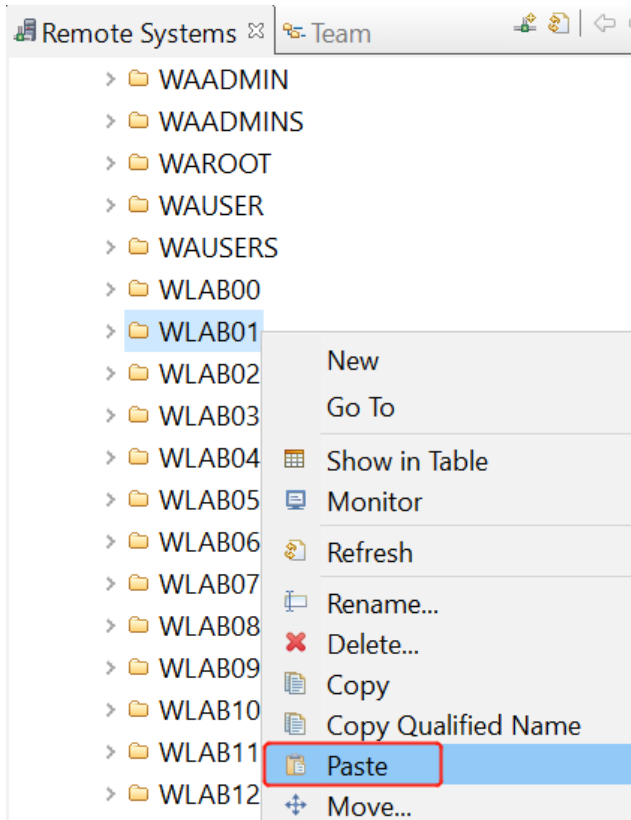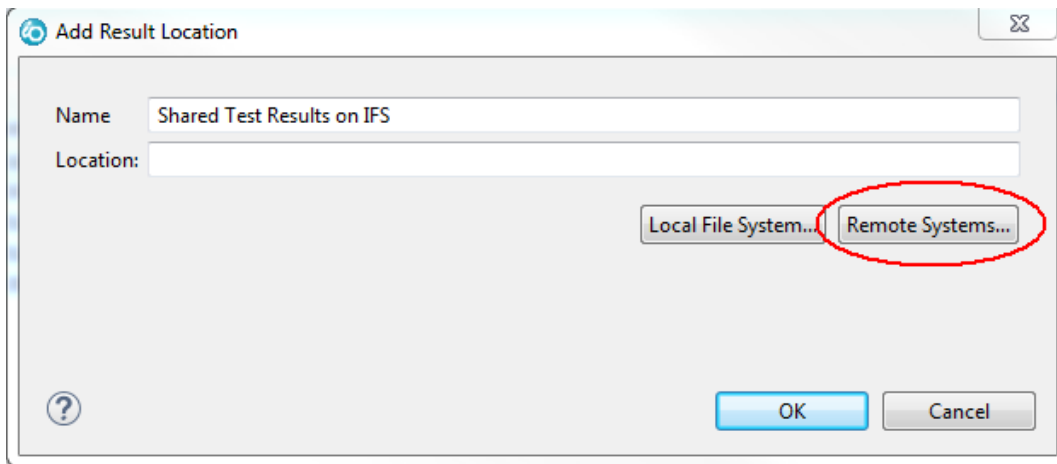
Now your teammates could see the coverage results generated by you. To view and analyze these results in their own workspace:
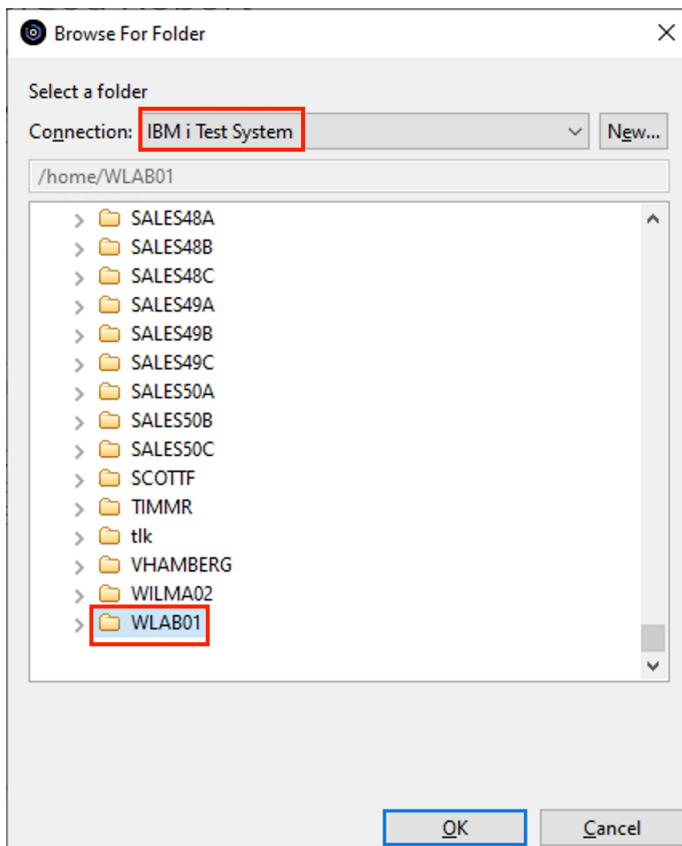
__14.    Right-click on the Code Coverage Results view and select **Add Result Location**.
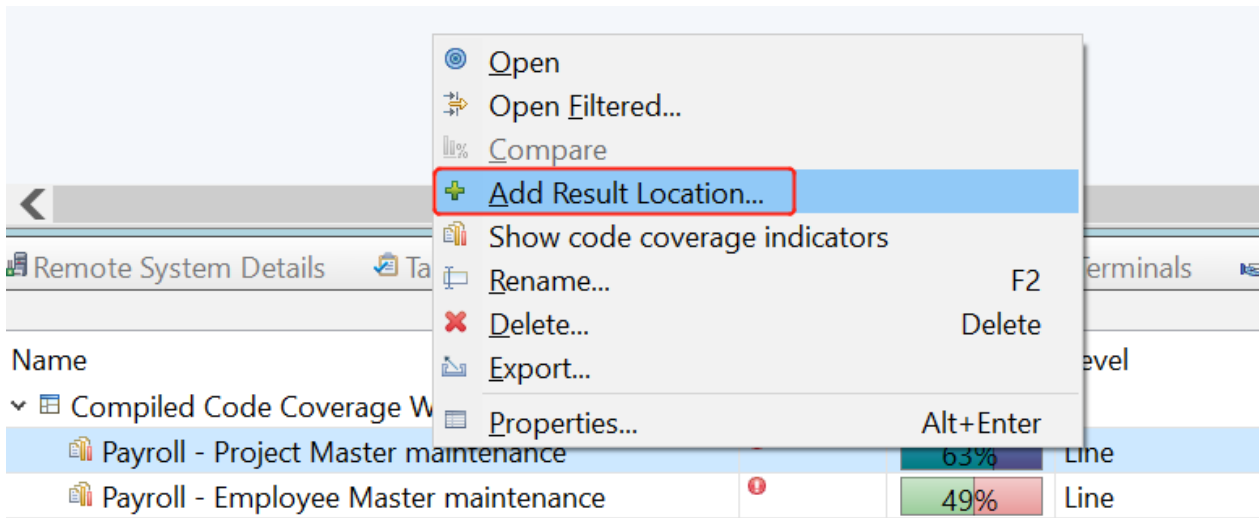
__15.    Enter **Shared Test Results on IFS** as the name and click **Remote Systems**.

__16.    Select your IBM i connection.

__17.    Expand **Home** and select **WLABxx**.



__18.    Click OK.

__19.    Click OK again to close the Add Result Location dialog.



A new result location has been created and any coverage results in the associated IFS folder should appear under this new result location.

# Congratulations!

You have successfully completed the RDi Working with Code Coverage lab exercises.

In this module you learned how to work with the code coverage tools. You started by running code coverage with a service entry breakpoint. You then viewed code coverage results within the editor and in both the HTML and PDF report formats. You also compared two code coverage results and merged them together. Lastly you exported code coverage results to a local directory and then imported it back.

We recommend that you move on to the next lab in the sequence; browse the list of labs on Rational Developer for i - Hands-On Labs at http://ibm.biz/rdi_labs to choose a lab of interest.

More information, material and opportunities to discuss the product can be found at our RDi Hub:

http://ibm.biz/rdi_hub

# Appendix A  Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

# Appendix B  Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

| | | | | | |
|---|---|---|---|---|---|
| IBM | AIX | CICS | ClearCase | ClearQuest | Cloudscape |
| Cube Views | DB2 | developerWorks | DRDA | IMS | IMS/ESA |
| Informix | Lotus | Lotus Workflow | MQSeries | OmniFind | |
| Rational | Redbooks | Red Brick | RequisitePro | System i | |
| System z | Tivoli | WebSphere | Workplace | System p | |

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product and service names may be trademarks or service marks of others.

IBM

Please Recycle

IBM Software