16406: What's New In the IBM Problem Determination Tools

Francisco Anaya

fanaya@us.ibm.com

@fmanaya54

@IBMDebugTool

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

SHARE in Seattle 2015

Agenda

- z/OS Tools
  - What are the z/OS Tools
- COBOL Explorer in Fault Analyzer
- API
  - Why are z/OS Tools providing APIs
- Code Coverage Eclipse plugin Tech Preview
- Load Module Analyzer plugin Tech Preview

z/OS Tools Introduction

Not more PD Tools but z/OS Tools

#SHAREorg

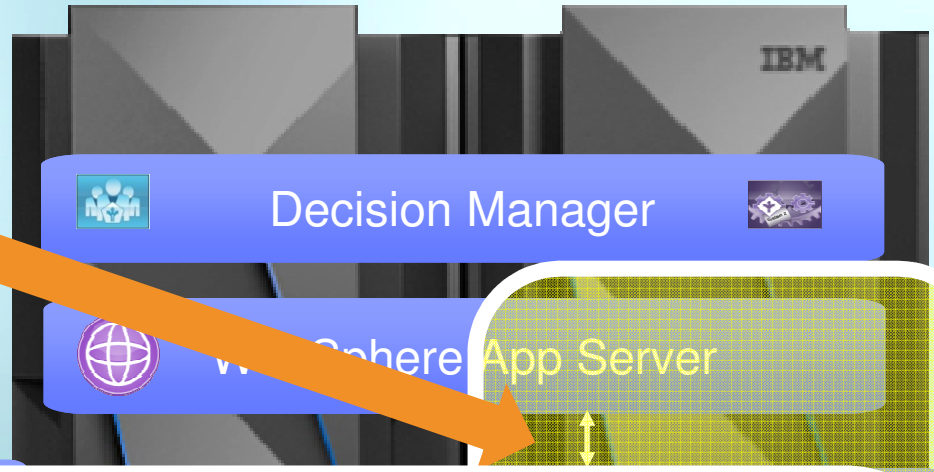
SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Insert Custom Session QR if Desired.

SHARE in Seattle 2015

WebSphere on z Portfolio enables Systems of Engagement

z/OS Tools help you do what you need to do more efficiently & affordably



What is z/OS Tools?

CICS Tools

- CICS Configuration Manager for z/OS
- CICS Deployment Assistant for z/OS
- CICS Interdependency Analyzer for z/OS
- CICS Performance Analyzer for z/OS
- CICS VSAM Recovery for z/OS
- CICS VSAM Transparency for z/OS

❑ CICS Modernization Solution Pack for z/OS
❑ CICS Optimization Solution Pack for z/OS

Problem Determination Tools for z/OS

- Application Performance Analyzer for z/OS
- Debug Tool for z/OS
- Fault Analyzer for z/OS
- File Manager for z/OS
- Workload Simulator for z/OS and OS/390
- Data Set Commander for z/OS
- Hourglass

❑ Problem Determination Modernization Solution Pack
❑ Problem Determination Solution Pack
❑ Problem Determination Testing Solution Pack

Employee

z/OS Tools Key Usage Scenarios

Understand the potential saving for free

Make adjustments for improved performance

Enhance applications to reduce resource consumption

Implement Automated Problem Diagnostic

Simplify management of infrastructure, application and data

Improve efficiency of development and quality of applications

Prepare Enterprise Data for Open World

Discover and exploit existing assets to realise additional value

Understand the potential impact of new mobile workload

IBM CICS tools: Customer ROI from Discovery, Optimization, and Enablement Outcomes

August 2014

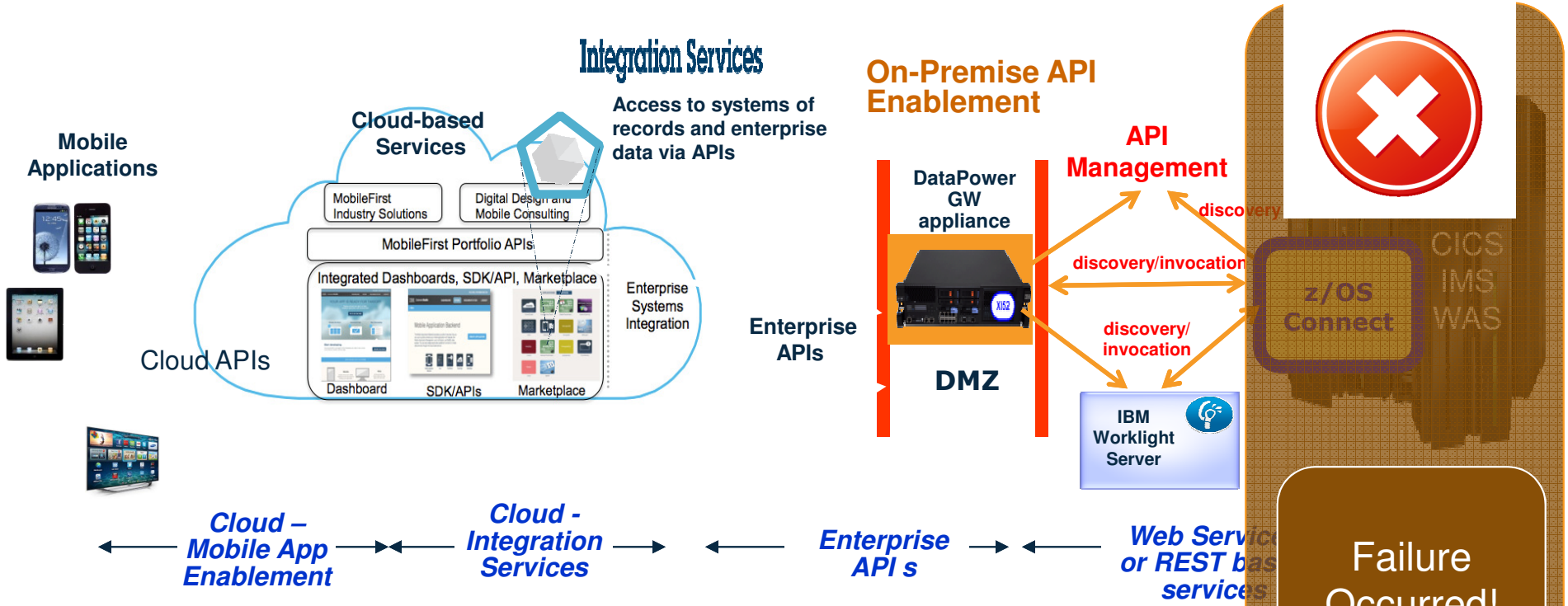
Branham
BRANHAM GROUP INC.

- From 10% to 25% CPU savings for applications that were converted to threadsafe.
- Upwards of 80% time savings to identify, code and test applications when converting to threadsafe.
- Upwards of 90% time savings to identify and validate typical performance issue related changes.
- An average of 66% less time to administer CICS Service Definition Changes, which happen on monthly, weekly and daily basis.
- A reduction from weeks to hours for the discovery and documentation of existing CICS topologies.
- A 50% reduction in the time it takes to identify application code and files that are no longer used and can be decommissioned.
- A 75% reduction in time to recover from a major issue related outage from days to hours.

IBM Fault Analyzer
COBOL Explorer

(an interactive post-mortem debugger)

End-to-End Architecture for Mobile and Cloud Application invoking z Services using APIs



Mobile Application Development
- Invokes services and APIs

Cloud Integration Enablement:
- Enabling integration with APIs as well as technical services

API Enablement:
- Consumability by internal and external developers (creation and look up)
- Entitlement Management (securing, workload enforcement)
- Usage monitoring & Analytics

Servic
- Enables in
applica
applica
protocols (REST)

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

SHARE in Seattle 2015

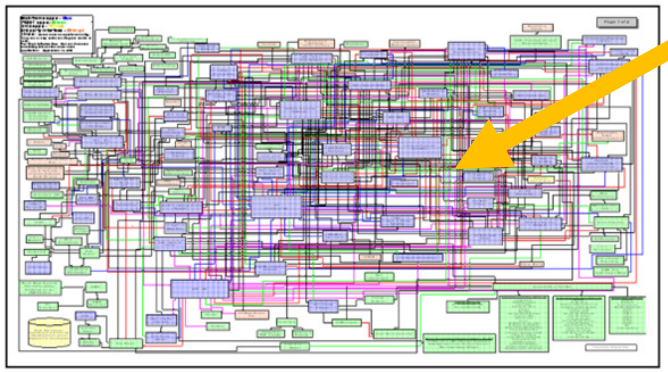
Approaches for Problem Diagnosis

1. Analysing what went wrong based on the information available in the environment where the error occurred.

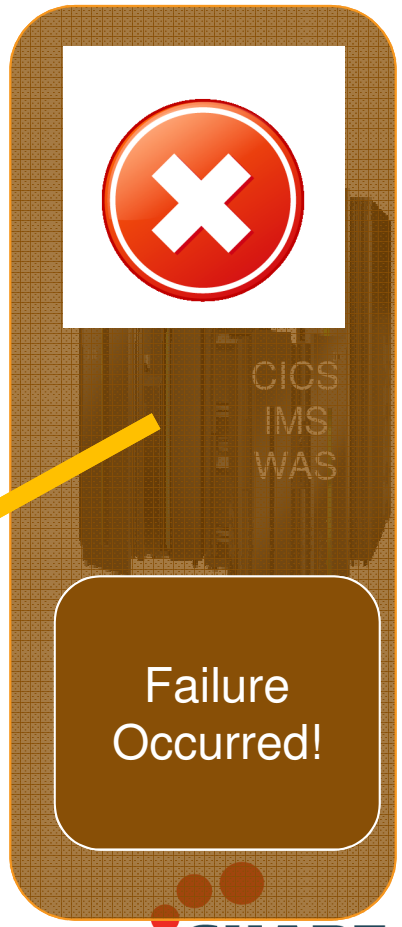
2. Create a replica of failing environment in a development environment and continue with further analysis.

What can be understood from the failing environment?

1. Abend analysis – what caused a program to fail at the source line level?
2. Data error – Identifying records containing erroneous data
3. Performance – resource consumption is too high or response time is not satisfactory



Complex system built on decades of incremental development



Abend Analysis Basics – Analysis and source line mapping

Application is abending. What do I do?

Simple S0CB in COBOL program analysis

Steps without FA:

1. Capture CEEDUMP or MVS dump
2. Determine abend offset into program from dump traceback
3. Obtain a matching compiler listing – need to make sure it really matches!
4. Using the listing, identify the matching source line for the abend offset
5. Identify data fields involved and their offsets into working storage
6. Validate the content of each data field in the dump to determine the one(s) in error
7. Fix the problem

Do I really want to do this?

Abend Analysis Basics – Analysis and source line mapping (continue)

Solution provided by Fault Analyzer report

pthfmd2:2820/KENICHI.HISTORY(F00002)-Report

```
Module COBTSE, program COBTSE, source line # 17: Abend S0CB (Decimal-Divide Exception)

 I B M   F A U L T   A N A L Y Z E R   S Y N O P S I S


A system abend 0CB occurred in module COBTSE program COBTSE at offset X'458'.

A program-interruption code 000B (Decimal-Divide Exception) is associated with
this abend and indicates that:

  The divisor was zero in a signed decimal division.

The cause of the failure was program COBTSE in module COBTSE.  The COBOL source
code that immediately preceded the failure was:

  Source
  Line #
  ------
  000017               DIVIDE NUM1 BY NUM2 GIVING NUM3.

The COBOL source code for data fields involved in the failure:

  Source
  Line #
  ------
  000006       01 NUM1 PIC 99 VALUE ZERO.
  000007       01 NUM2 PIC 99 VALUE ZERO.
  000008       01 NUM3 PIC 999 VALUE ZERO.

Data field values at time of abend:

  NUM1 = 10
  NUM2 = 0  *** Cause of error ***
  NUM3 = 23
```

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

Gives you an explanation of failure and pinpoint erroneous source line #

Displays source information involving in the error

Showing you precisely what data needing the fix

What is COBOL Explorer?

- COBOL Explorer takes debugging your (abended) COBOL program to the next level by allowing you to work with a summary view of the program which starts by revealing only source lines you need to know about. You can then 'expand' sections of the program to investigate further, or create another summary view to pursue something different.

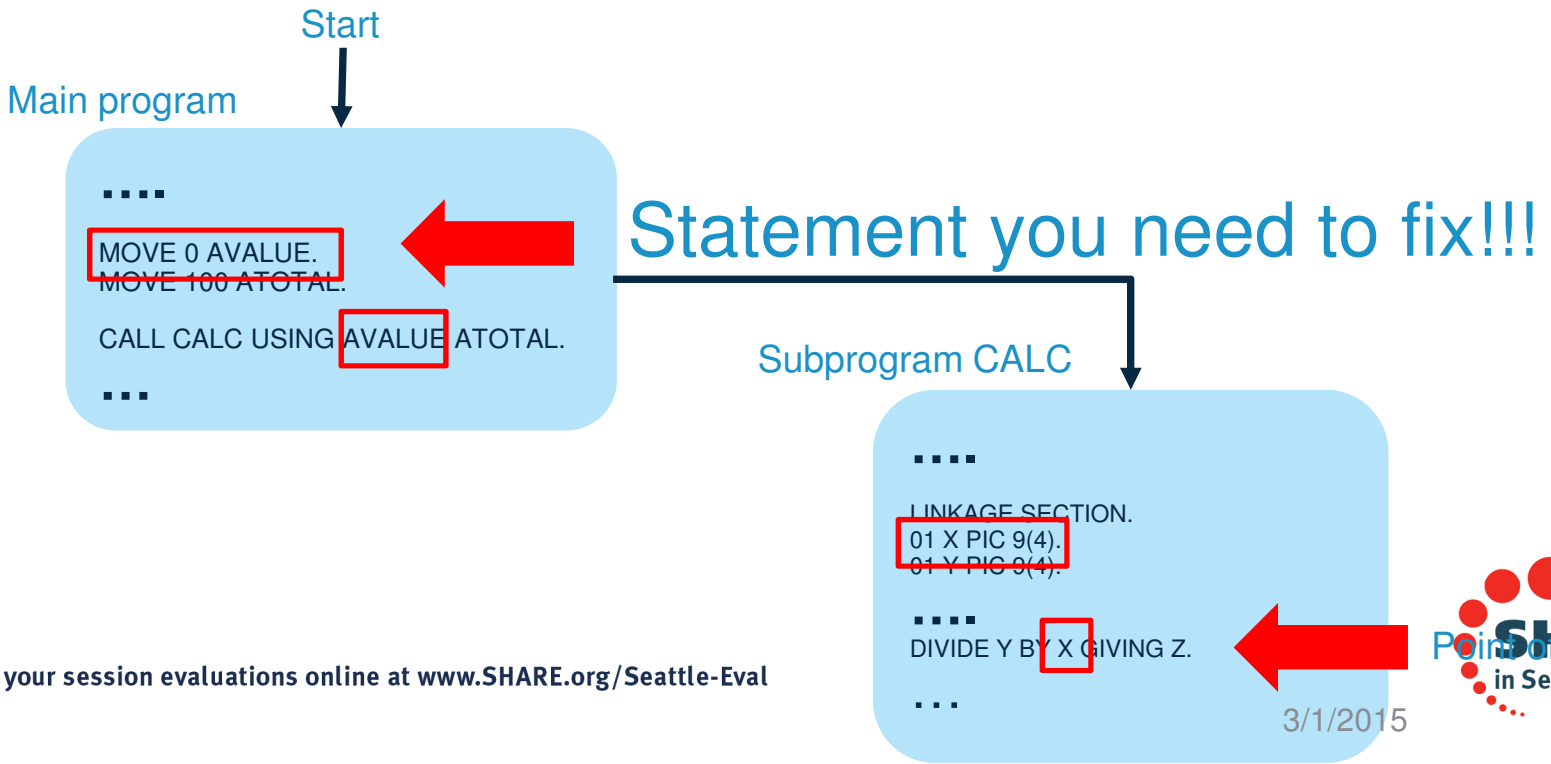
- Query your program to see all lines where a variable is used, all places a procedure is called, or the value of variables on any executable source line.
- IBM Fault Analyzer V13, PTF UI18641 introduced this new feature, documented in the latest User's Guide and Reference SC19-4116-04

Replaces the traditional source (editor style) viewer with a collapse/expand interface.

COBOL Explorer: Post-mortem debugger

- Want to better understand the execution path which led to a failure from a dump?
- Want to see how value of a variable changed which contributed to a failure?

- COBOL Explorer is a new feature of Fault Analyzer which allows you to navigate your application's execution path from a point of failure interactively. It allows examination of a variable and how the value changed which led to failure.



IBM Fault Analyzer
COBOL Explorer Video
(an interactive post-mortem debugger)

Watch the video @
http://www.youtube.com/watch?v=ZXwsaBnfk2Q

z/OS Tools APIs

Insert Custom Session QR if Desired.

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Why do I care about APIs?

- Our tools come with user interfaces. ISPF, Eclipse, WebPages

- Tools divide into two categories
  - Dev/Ops tools – chat directly with live systems
    - Debug Tool, File Management, Configuration Management, Deployment Assistant

  - Collect Data and store for analysis
    - Tools are configured what to collect/how often to collect, Data captured by tool is analyzed for issue/trend
    - Fault Analyzer, Application Performance Analyzer
    - Interdependency Analyzer, Performance Analyzer
    - Debug Tool Code Coverage

APIs allow me to integrate dev/ops tools

- Dev ops tools with user interface can mean replacing existing interface with disruption/adoption
- Existing process can have bespoke steps
- Help you build continuous testing environments
- Build continuous monitoring applications that provides you with data and metrics to operations, QA, development and other stake holders during the different stages of the delivery cycle.
- What if we could drive the tools from the existing process

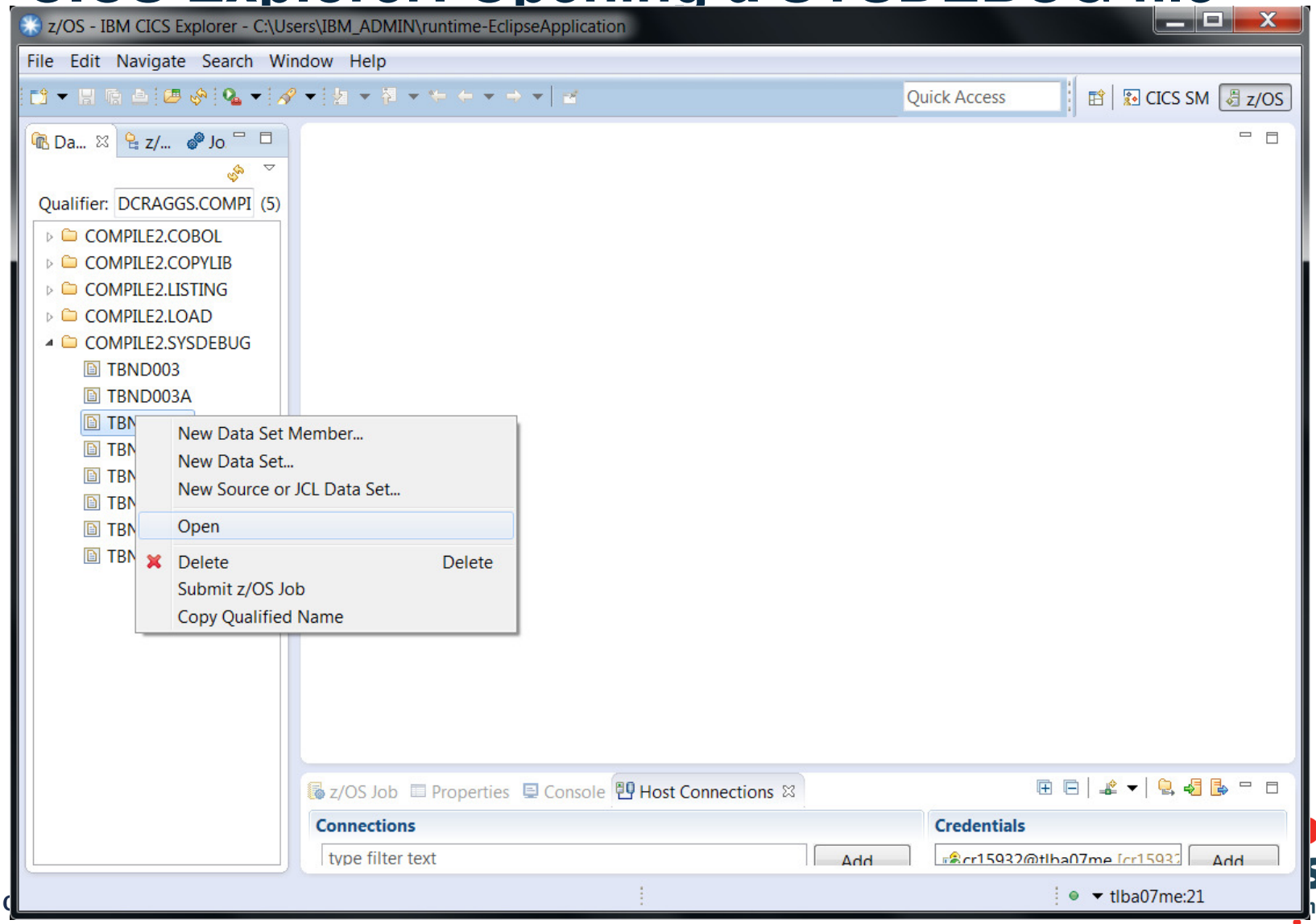
Dev ops APIs

- Debug Tool
  - Managing debug configurations
  - Code Coverage reports
  - Extracting source from debug files
  - Authenticating credentials
- Fault Analyzer – retrieving abend reports occurred during test runs
- Configuration Manager
  - moving change packages as part of source code promotion
  - Powertools for CSD and BAS.
  - (what makes good demo here)
- Deployment Assistant – cloning and provisioning CICS regions and systems

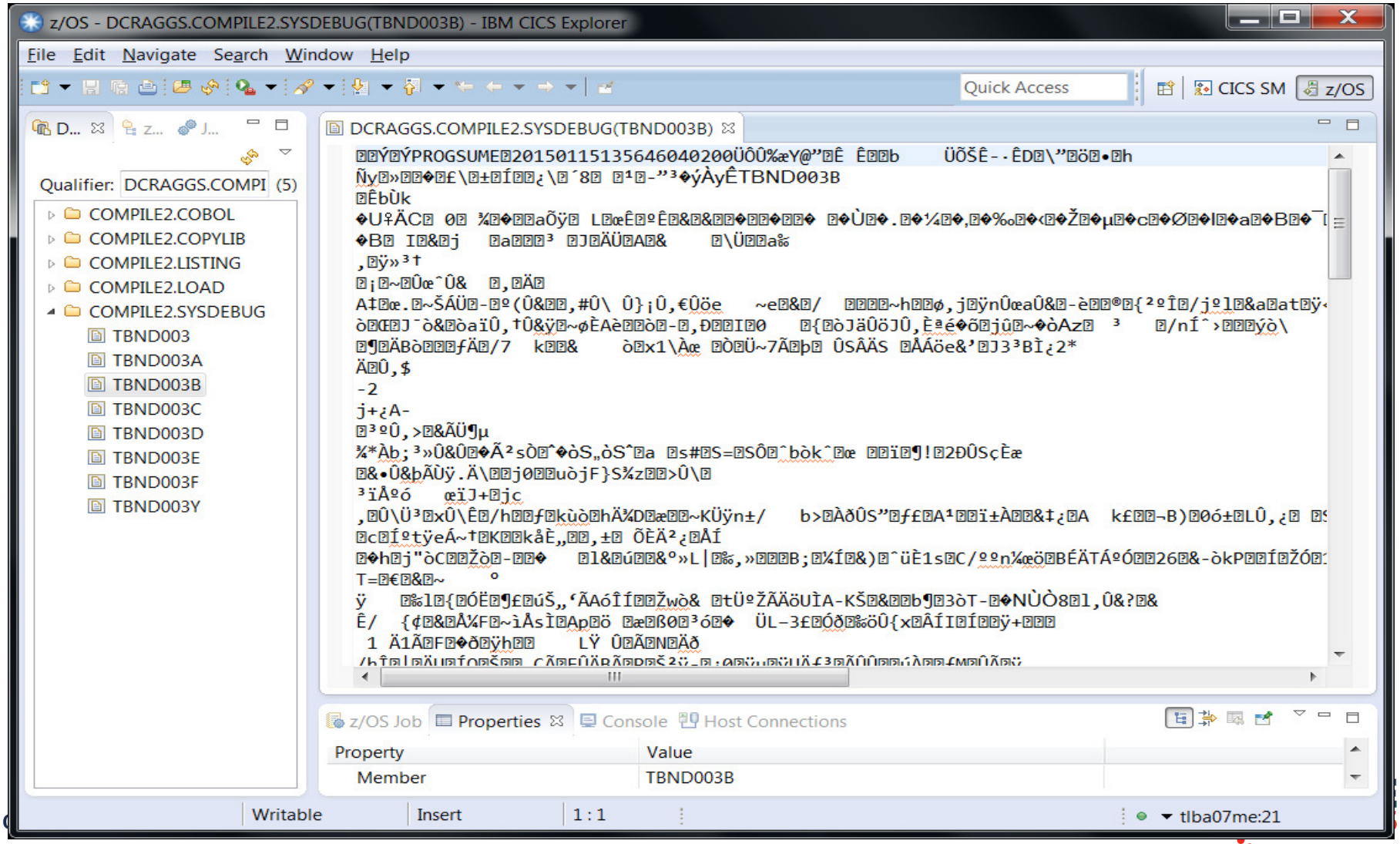
Extracting the expanded source from a COBOL SYSDEBUG File using APIs

- COBOL SYSDEBUG files are created by the compiler when the TEST(SEPARATE) compile option is specified
- The SYSDEBUG file contains the debug data needed for a debug Session as well as the expanded source used during compilation
- The contents are encrypted

CICS Explorer: Opening a SYSDEBUG file



CICS Explorer: Showing Encripted SYSDEBUG file



API 1 - Authenticate

This API provides an authentication method for the host system.

REST URI : /CodeCoveragereportGen/CCrest/reports/csectSrc

POST request : username=*myusername*&password=*mypassword*

API 2 – Extract Source

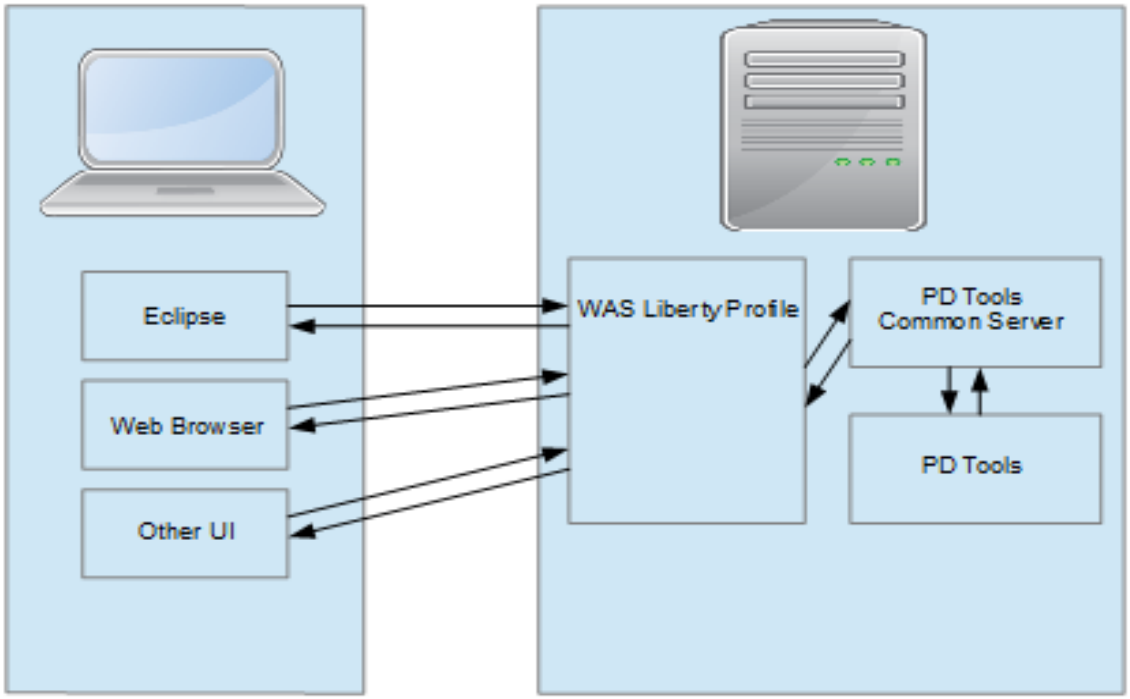
This API will return an XML document containing the extracted source code.

REST URI : /CodeCoveragereportGen/CCrest/reports/csectSrc

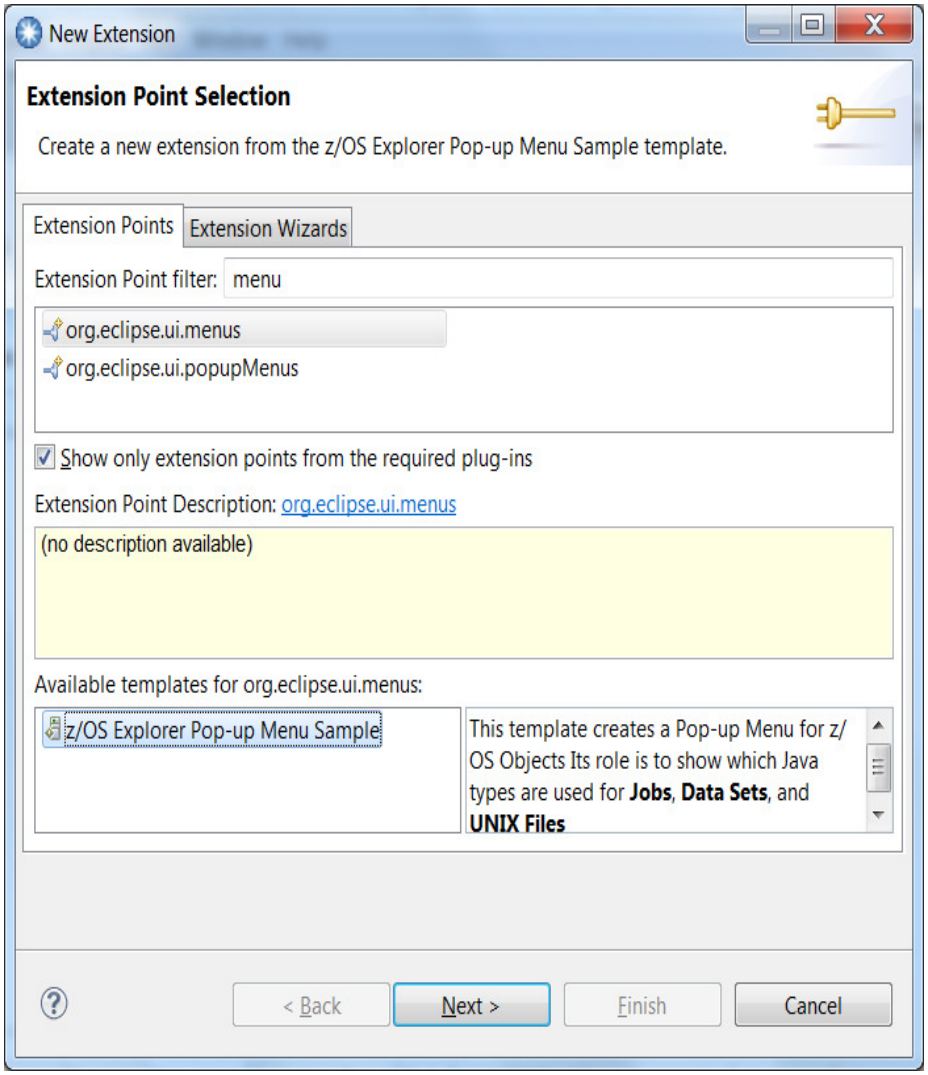
POST request : src=

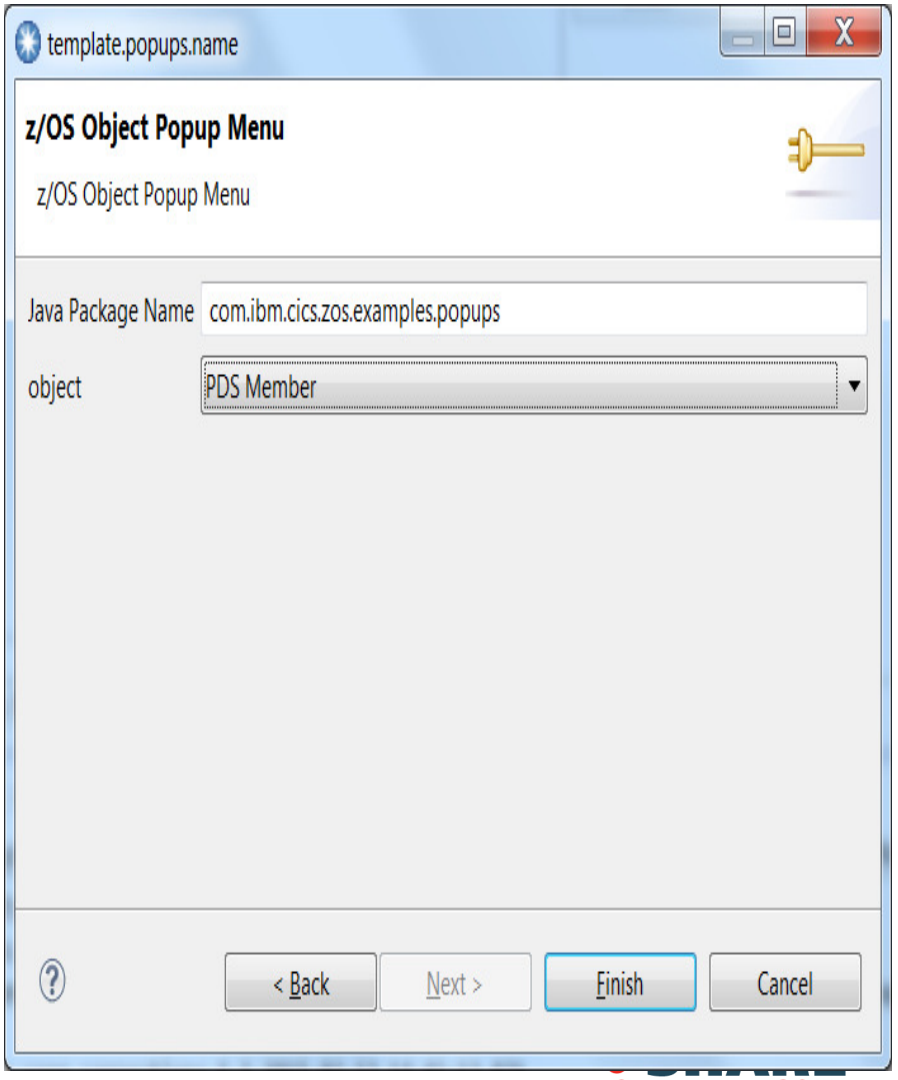
```
<COMPILATIONUNIT>
    <PROGRAMDSNAME>dataset_name</PROGRAMDSNAME>
    <PROGRAMDSTYPE>dataset_language</PROGRAMDSTYPE>
    <CSECT>
        <EXTNAME>member_name</EXTNAME>
    </CSECT>
</COMPILATIONUNIT>";
```

Components

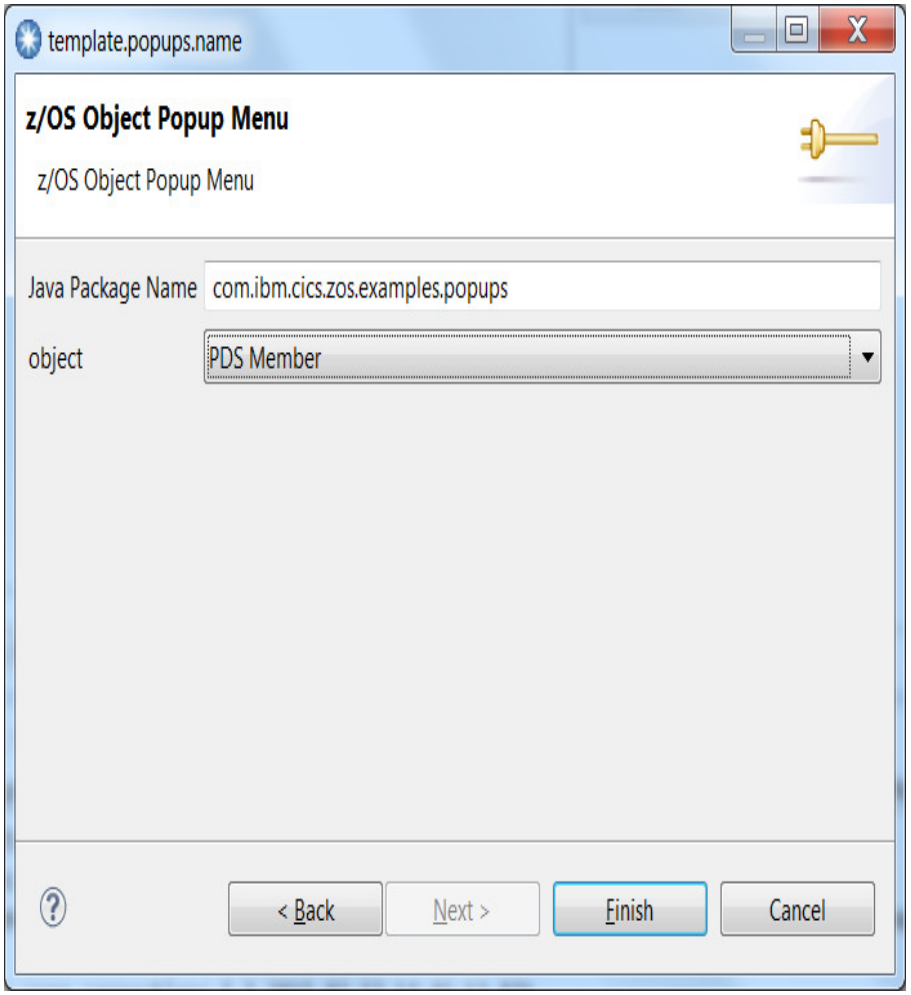

Eclipse
Web Browser
Other UI
WAS Liberty Profile
PD Tools
Common Server
PD Tools

Creating an extension in CICS Explorer





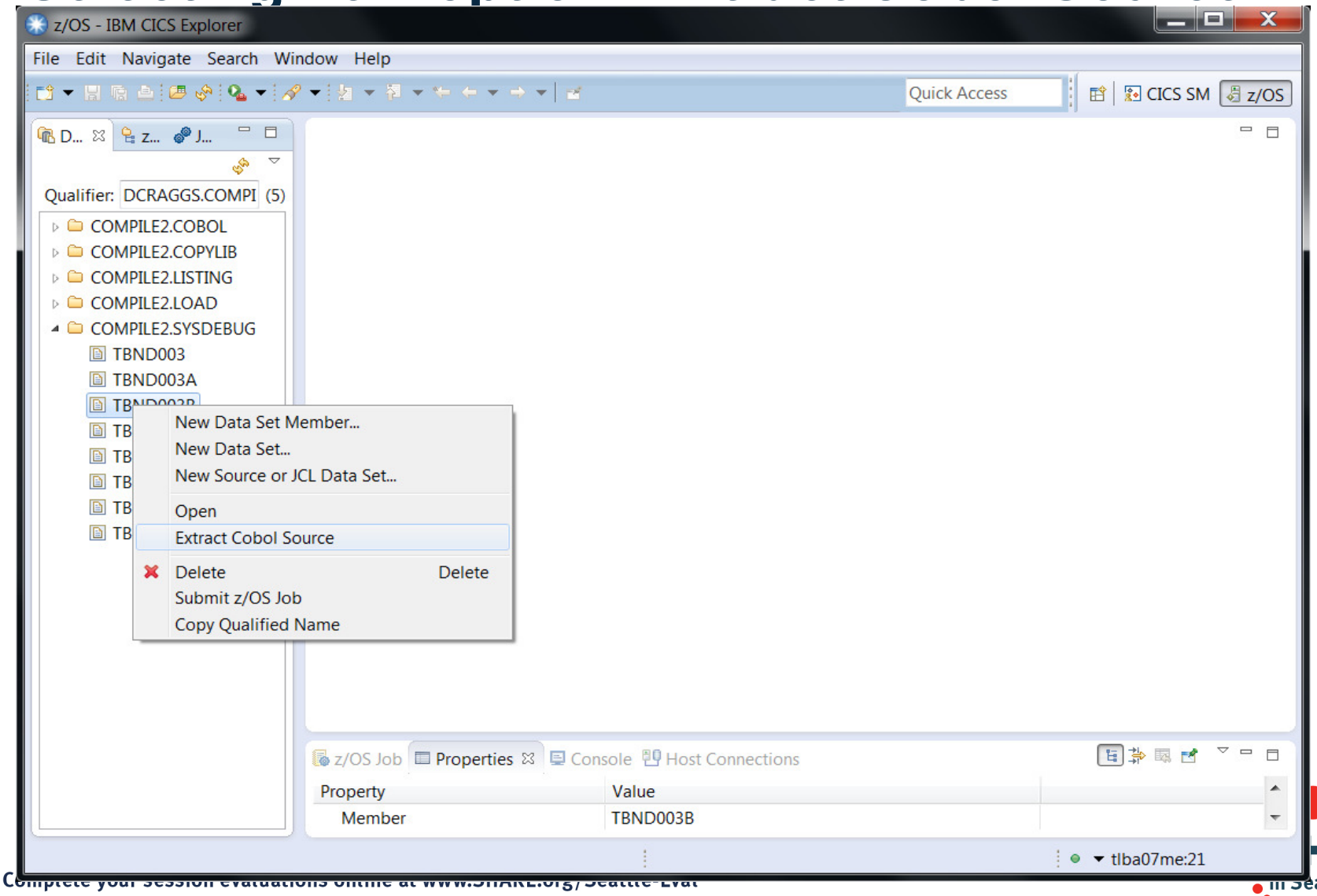
Creating an extension in CICS Explorer



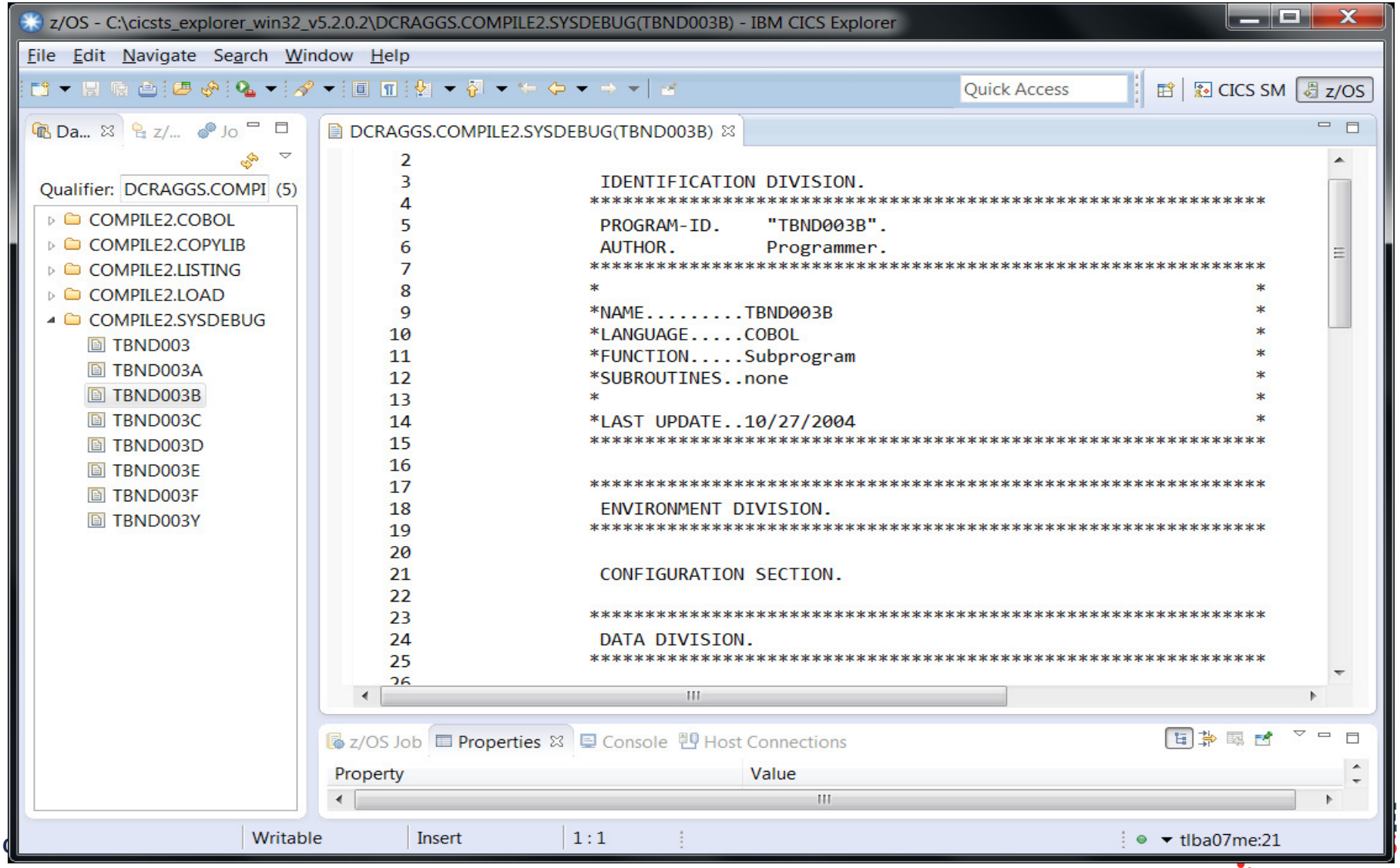

```
@Override
public Object execute(ExecutionEvent event) throws ExecutionException {

    Shell currentShell = PlatformUI.getWorkbench().getActiveWorkbenchWindow().getShell();
    ISelection currentSelection = HandlerUtil.getCurrentSelection(event);
    Object selectedObject = null;
    if(currentSelection instanceof StructuredSelection){
        selectedObject = ((StructuredSelection)currentSelection).getFirstElement();
    }
    MessageBox box = new MessageBox(currentShell);
    box.setText("Hello PDS Member");
    if(selectedObject != null){
        box.setMessage(selectedObject.toString());
    }
    box.open();
    return null;
}
```

Selecting new option: Extract Cobol Source



Extracted source from SYSDEBUG file



Code updates ….

```
public static String callApi(String callUrl, String params) throws IOException {
    URL url = new URL(callUrl);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("POST");
    conn.setDoOutput(true);
    DataOutputStream out = new DataOutputStream(conn.getOutputStream());
    out.writeBytes(params);
    out.flush();
    out.close();
    BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
    String inLine;
    StringBuffer response = new StringBuffer();
    while ((inLine = in.readLine()) != null) {
        response.append(inLine);
        response.append("\n");
    }
    in.close();
    return response.toString();
}
```

Code updates ….

```
String rest1 = "http://localhost:9080/CodeCoveragereportGen/CCrest/reports/authenticate";
String rest2 = "http://localhost:9080/CodeCoveragereportGen/CCrest/reports/csectSrc";

try {
    callApi(rest1, "username=myuser&password=mypass");
    PrintWriter pw = new PrintWriter(getDSN(selectedObject.toString()));
    pw.println(removeXML(callApi(rest2, "src=" + createPost(selectedObject.toString()))));
    pw.close();
} catch (IOException e) {
    e.printStackTrace();
}

IWorkbenchPage page = PlatformUI.getWorkbench().getActiveWorkbenchWindow().getActivePage();
File file = new File(getDSN(selectedObject.toString()));
IFileStore ifs = EFS.getLocalFileSystem().getStore(file.toURI());
try {
    IDE.openEditorOnFileStore(page, ifs);
} catch (PartInitException e) {
    e.printStackTrace();
}
```

Code Coverage and Load Module Analyzer
Tech Preview

Technical preview version of the Problem Determination Tools	Technical preview of the Problem Determination Tools, including the **Debug Tool Code Coverage and Load Module Analyzer plugins**. Available for download as a stand-alone Studio or as P2 and Installation Manager compatible plug-in packages.

http://www-01.ibm.com/support/docview.wss?uid=swg24039125

Insert Custom Session QR if Desired.

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

SHARE in Seattle 2015

Code Coverage Facilities in Debug Tool

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Insert Custom Session QR if Desired.

SHARE in Seattle 2015

Code Coverage Facility

- Uses Debug Tool backend
  - Plus
    - Can be driven by RDz
    - Highly customizable using XML
    - Allow parallel debug and code coverage session
    - Can be run in batch fully separated from UI or 3270
    - Perform better
    - Results presented in XML
    - APIs for extraction of results
    - ISPF, Eclipse and Web facility for setup and report creation
    - Rollup support
  - Minus
    - Not integrated with RD/z testing facilities yet

Code Coverage Facility Design Goals

- Use the same setup needed for a debug session
- Allow the developer to have a CC data gathering session in parallel with a debug session.
  - When use in this mode if the logical flow of the program is altered by using command like GOTO, or JUMPTO, or modifying the contents of a variable the report will contain and indicator.
- Enhance quality by proper unit test validation.
  - Help the developer with making sure that the proper unit test has been performed by showing whether new or modified code have been executed.
  - Validation of proper unit test before checking code into source library

Code Coverage Facility Design Goal

- Allow for the code coverage data gathering to be run unattended in batch
- Allow for the code coverage data gathering to be launched in several modes
  - Debug Tool MFI (3270) , RD/z, or IBM Problem Determination Tools Studio.
- Provides an ISPF and Eclipse interface for the selection and creation of reports

Code Coverage Facility Design Goal

- Provide different medium for the reports.
  - Online reports using ISPF
  - Reports in PDF format
  - Reports in XML for further transformation if required
  - Online reports using a Web or mobile app.
- Provide APIs to retrieve the reports for home grown tooling.
- Support all environments supported by Debug Tool
- Support for applications written in COBOL, PL/1, and C

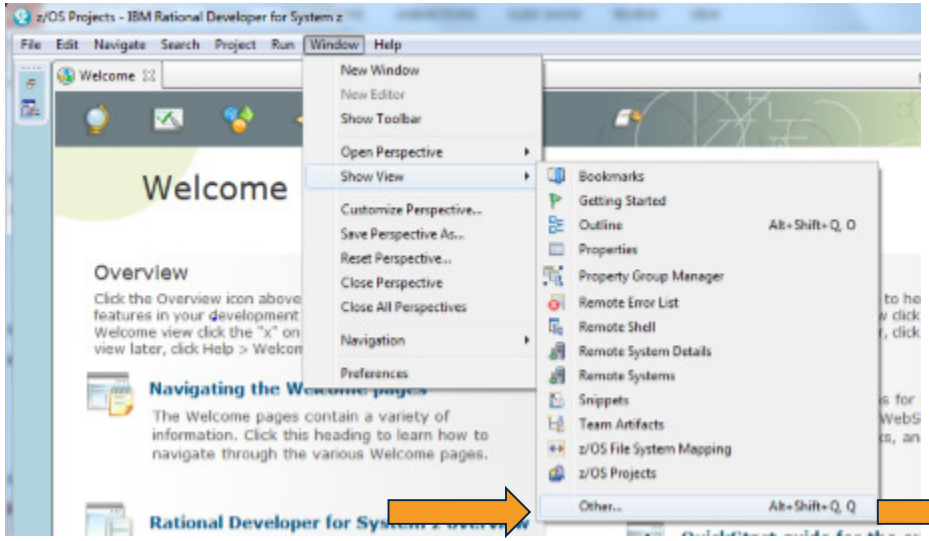
Steps to capture and display the Code Coverage

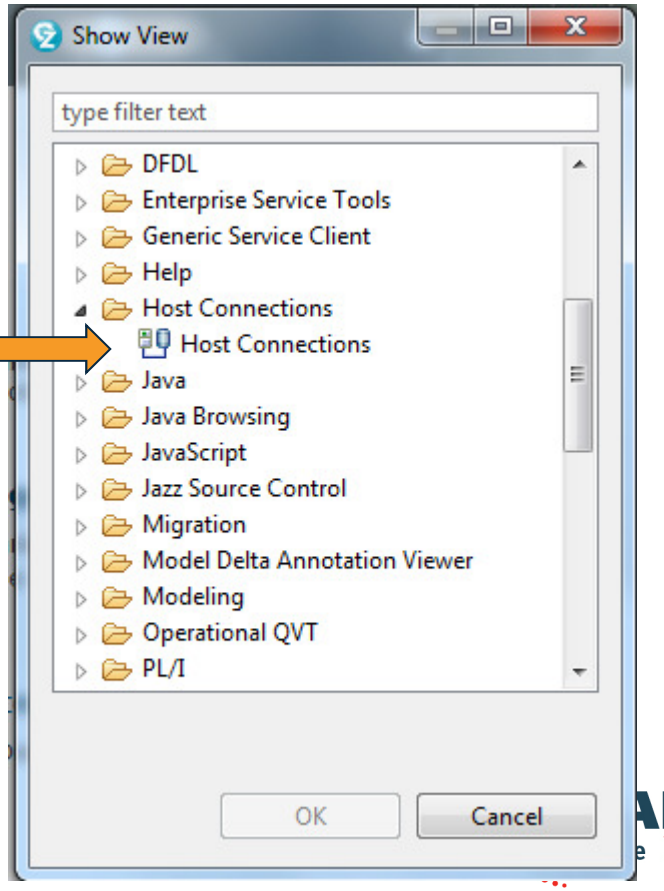
- Create Options File
- Define new EQAOPTS with location of CC output data and location of Option file.
- Specify new environment variable EQA_STARTUP_KEY in TEST runtime option
- Launch Debug Tool
- Provide Selection Criteria
- Create Code Coverage Report
- View Code Coverage Report
- Export Report (.xml, .pdf)

Installing Code Coverage Plug-in into RD/z

- You can download the Code Coverage Plug-in from the IBM PD Tools Website: http://www-01.ibm.com/software/awdtools/deployment/pdtplugins/
- The package is:
  - IBM Debug Tool Utilities (DTSP, Code Coverage, Load Module Analyzer, and JCL Wizard) 13.5.0.0
- You need to use Installation Manager to install this package into RD/z
- If you have previously installed the DTSP plug-in you need to un-install it first and then install this new package which includes DTSP as well.

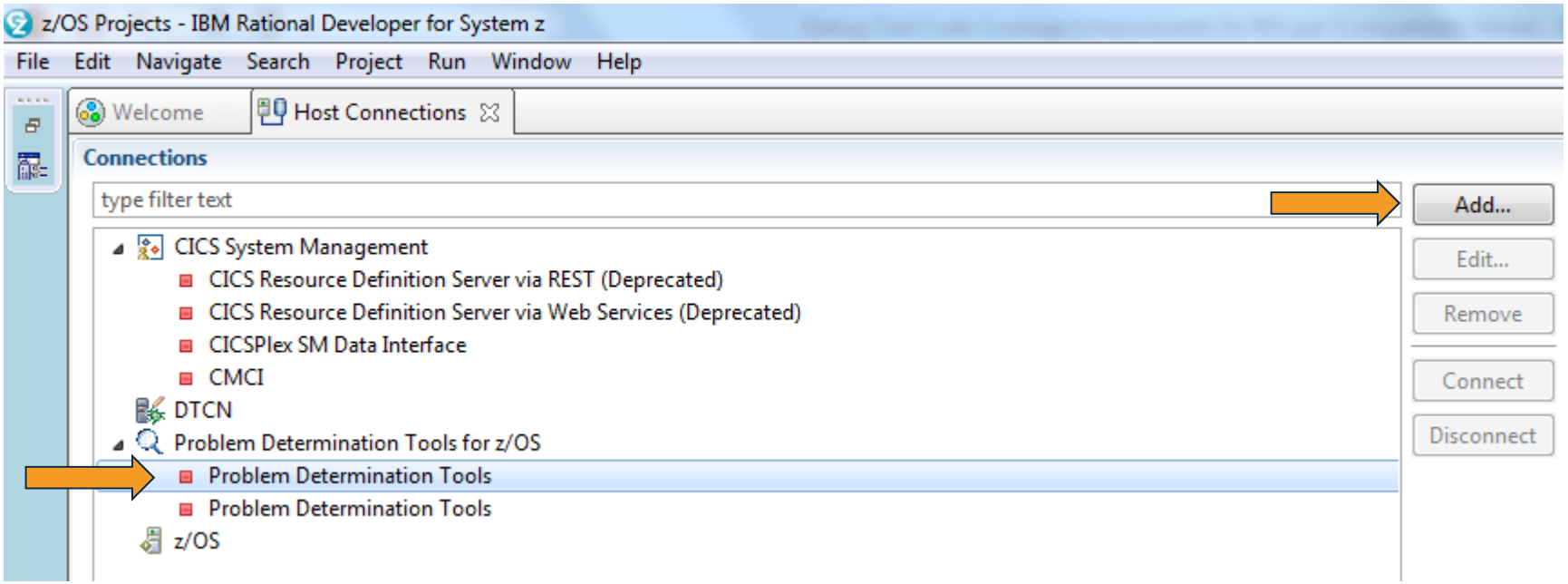
Connecting: Opening Host Connection view





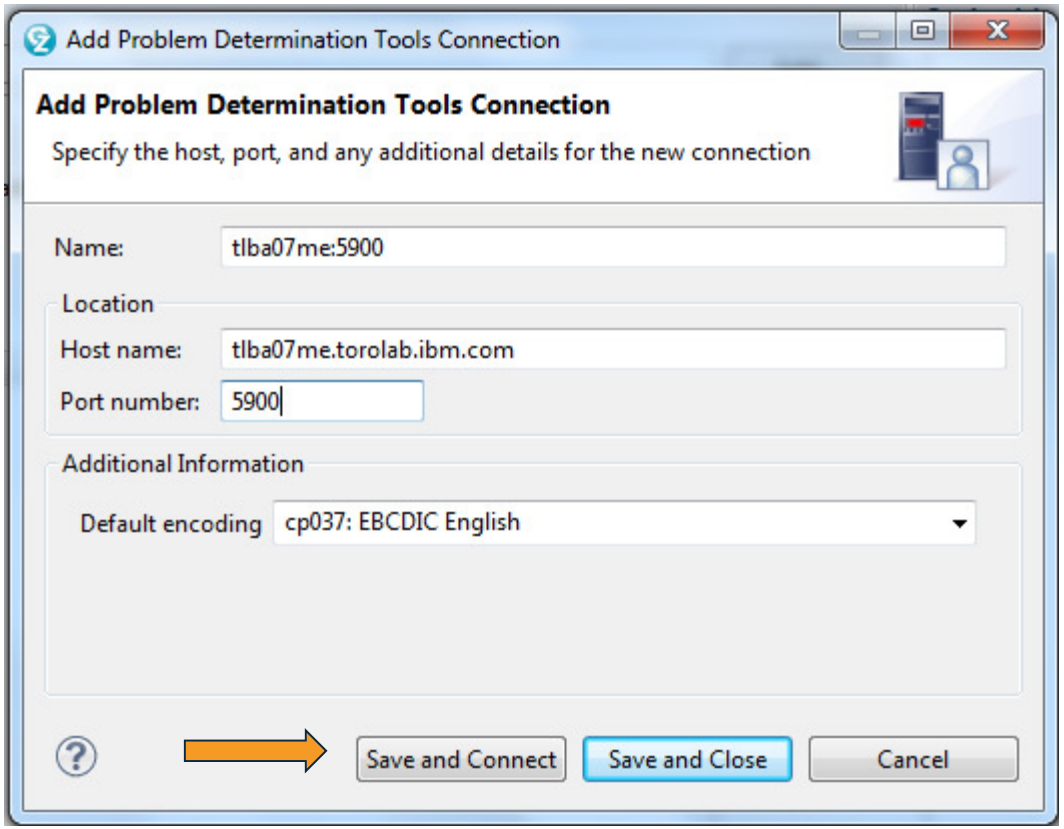
The host connection view allow you to connect
To PDTOOLS Common Server

Establishing a connection



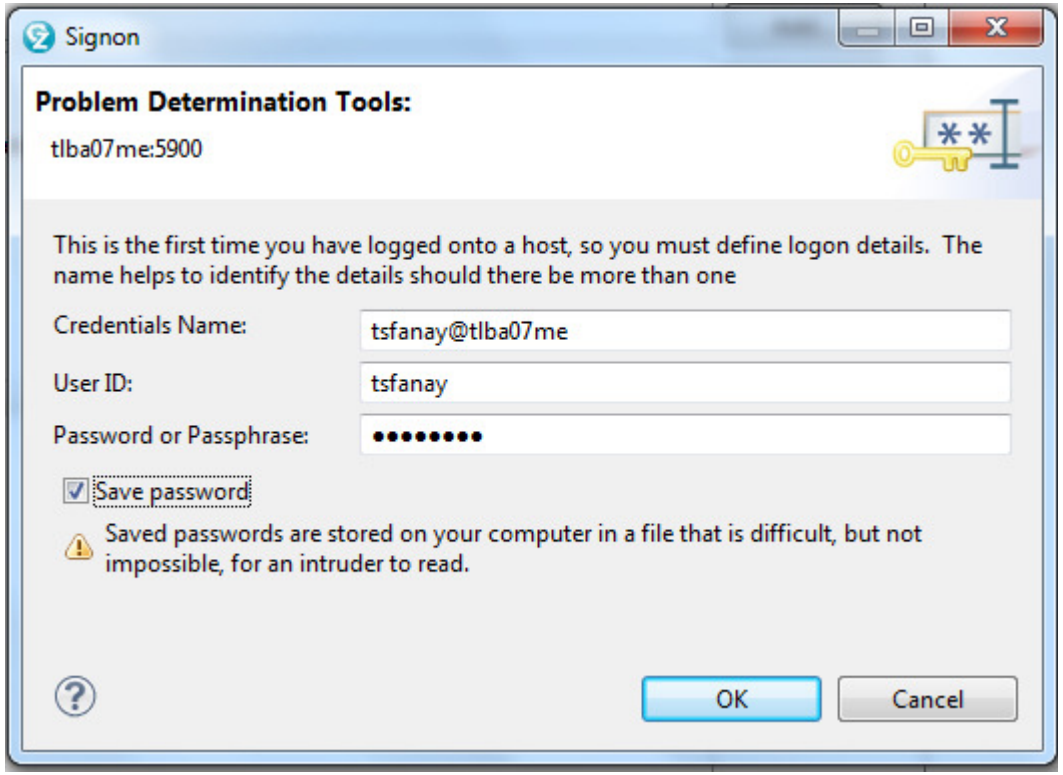
After expanding Problem Determination Tools for z/OS select a Problem Determination Tools Connection type and press Add

Configure your connection for your z/OS system



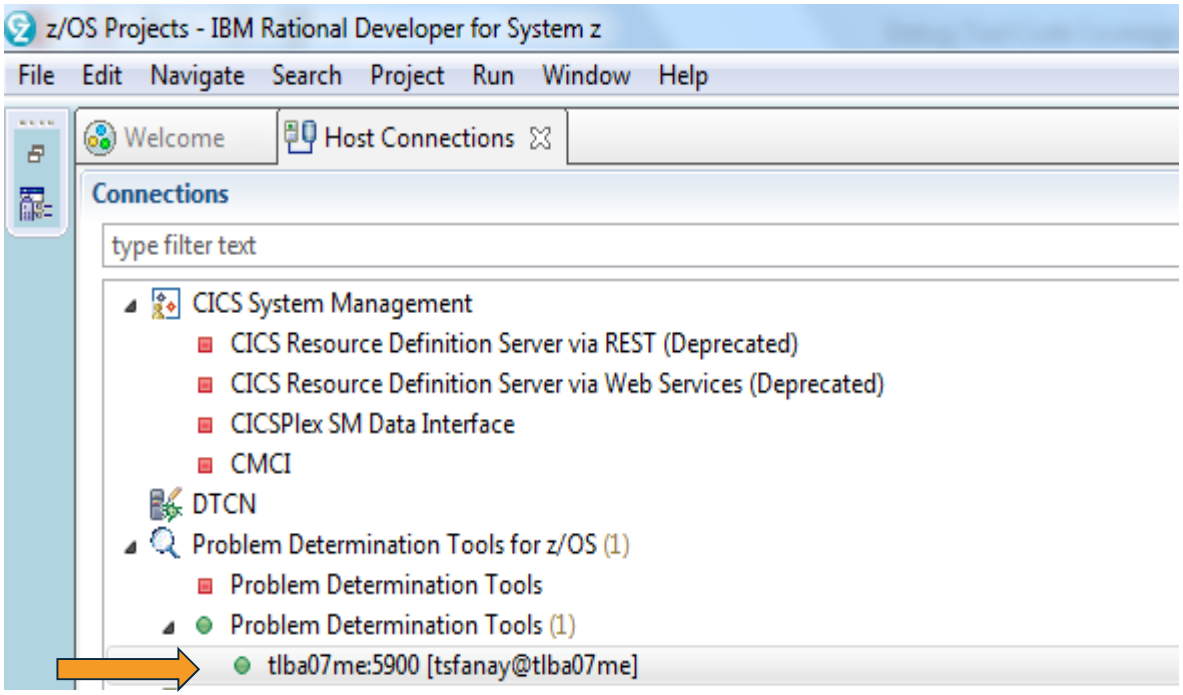
In this example we are choosing Save and Connect which will take us to a pane where you can specify your credentials

Specifying credentials



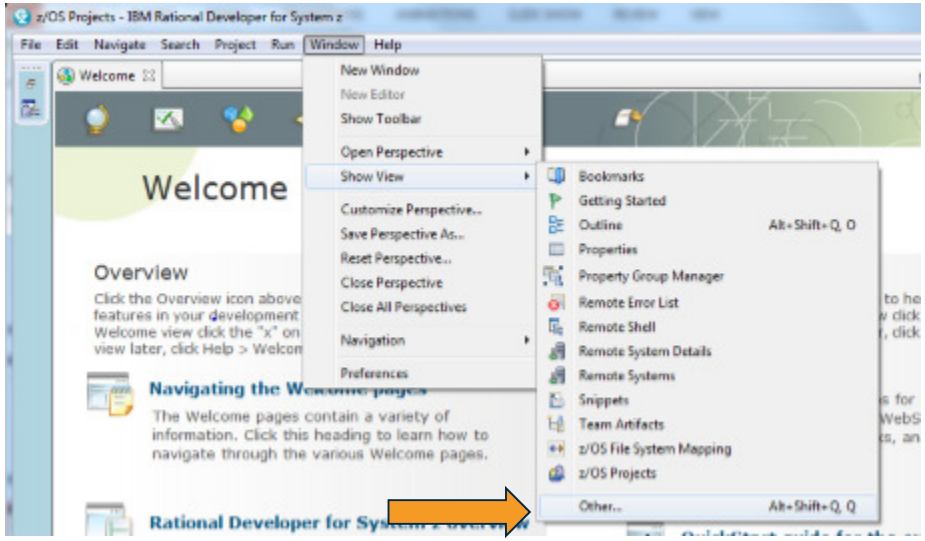
The connection and Credentials are valid for All PD Tools Plug-ins

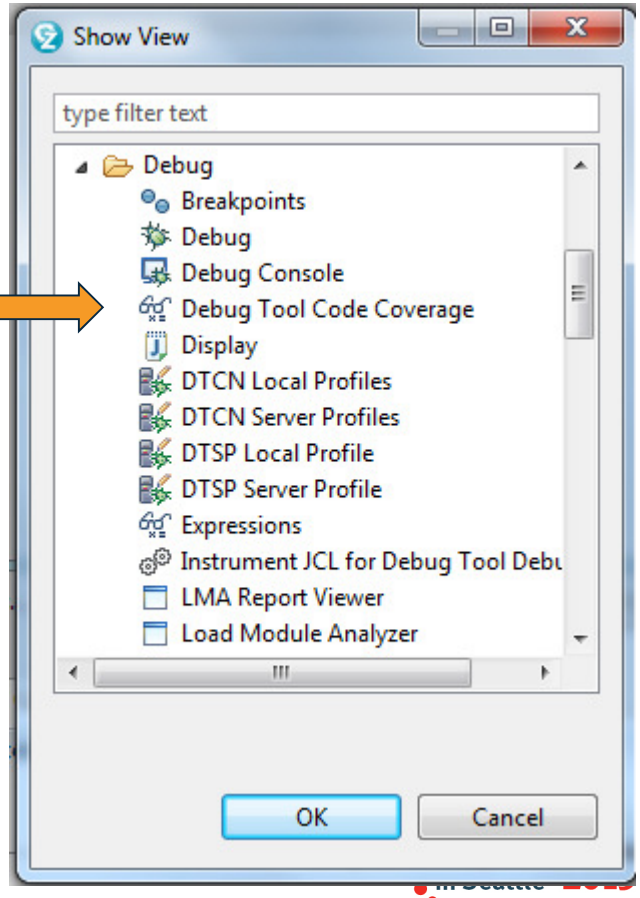
Connected to PD Tools Common Server



Now that we are connected we can then set up the code coverage session using the new Debug Tool Code Coverage Plug-in.

Opening Code Coverage view





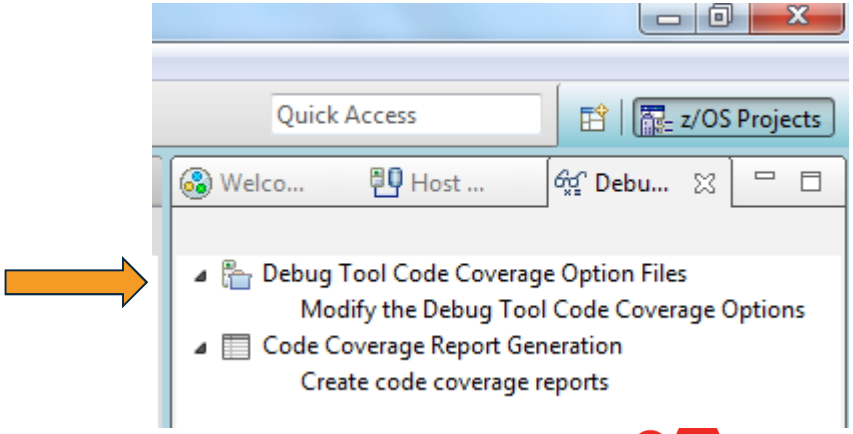
After opening Debug folder, select
Debug Tool Code Coverage and press OK

Code Coverage View

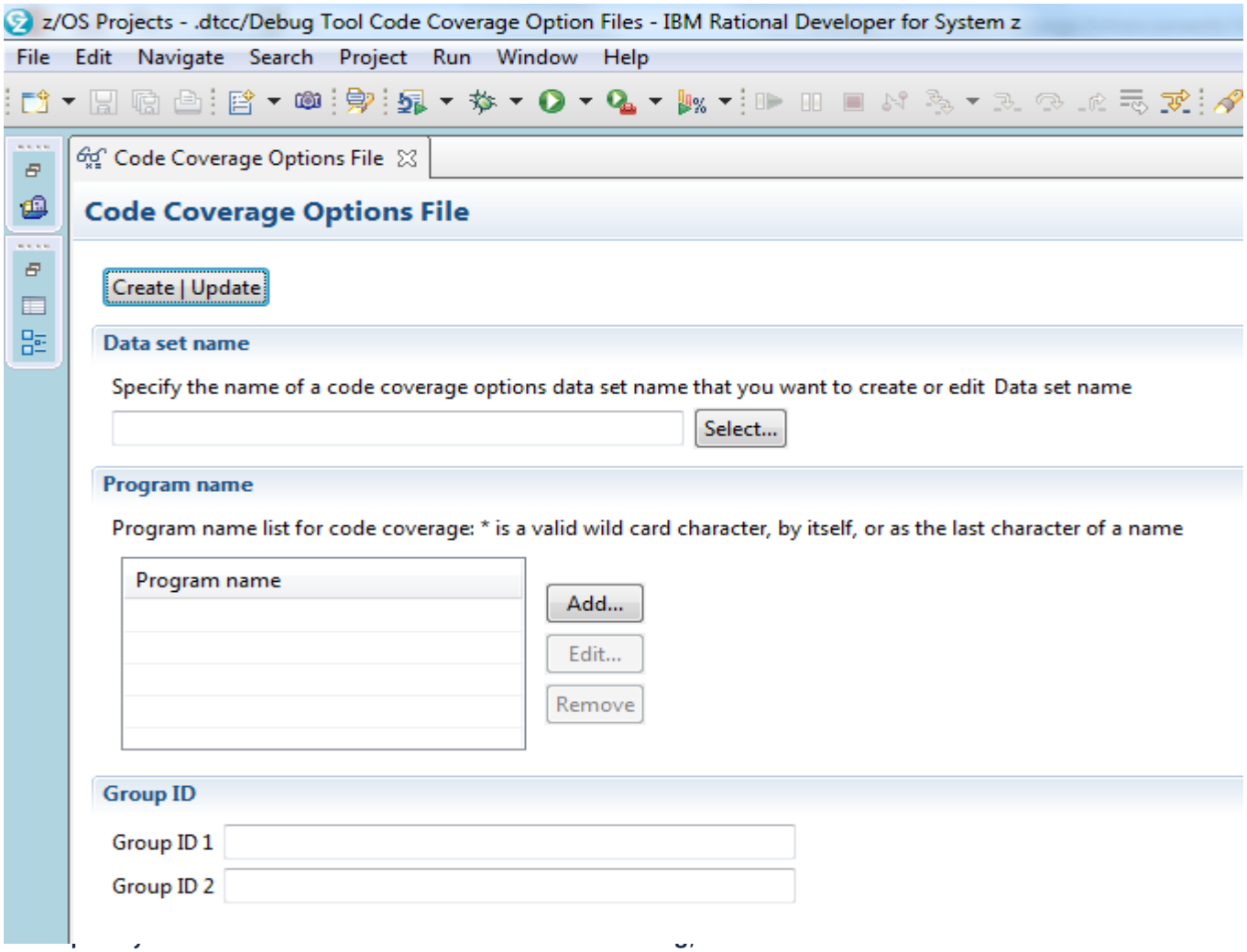
The Code Coverage view has two components:

- Debug Tool Coverage Option Files
  - Where you indicate the options to be use for capturing code coverage Observations
- Code Coverage Report Generation
  - Where you customize and create your reports

Double click on selection indicated with arrow. It will bring the Options Pane



Code Coverage Option File Pane

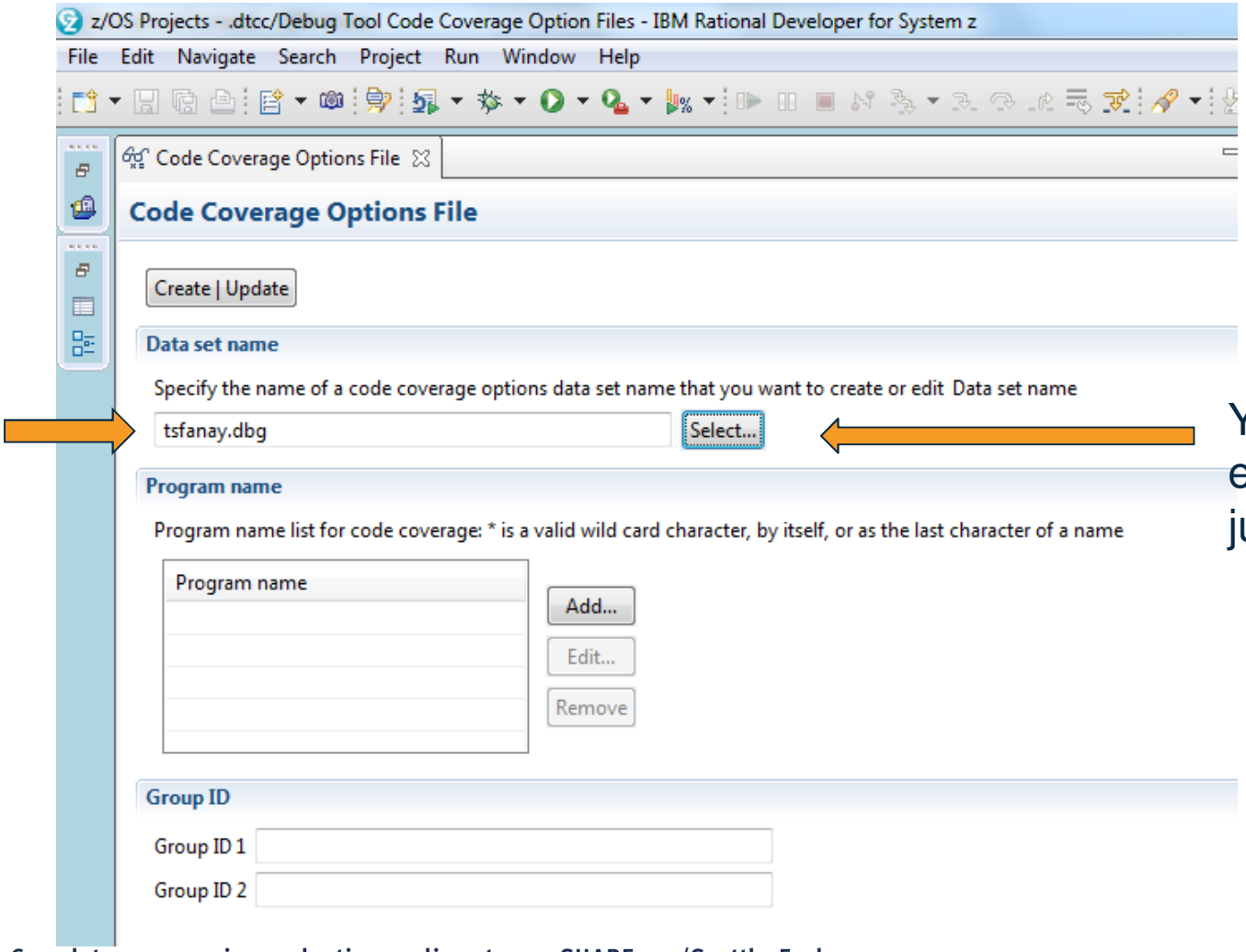


What is the CC Options file?

- Where you specify the programs that the analyst is interested in capturing CC observations:
  - An '*" => all programs in a transaction or batch job
  - A groups of programs with a wild card => ABCD*
  - Specific programs => MYPGM1, MYPGM2
- You can have more than one Options file but only one is use in each run.
- Facilities are available for creating the option file
- You can specify a group if you are interested in grouping results when creating a report
- You can create the Option file and specify the programs manually. It uses an XML like syntax. See below

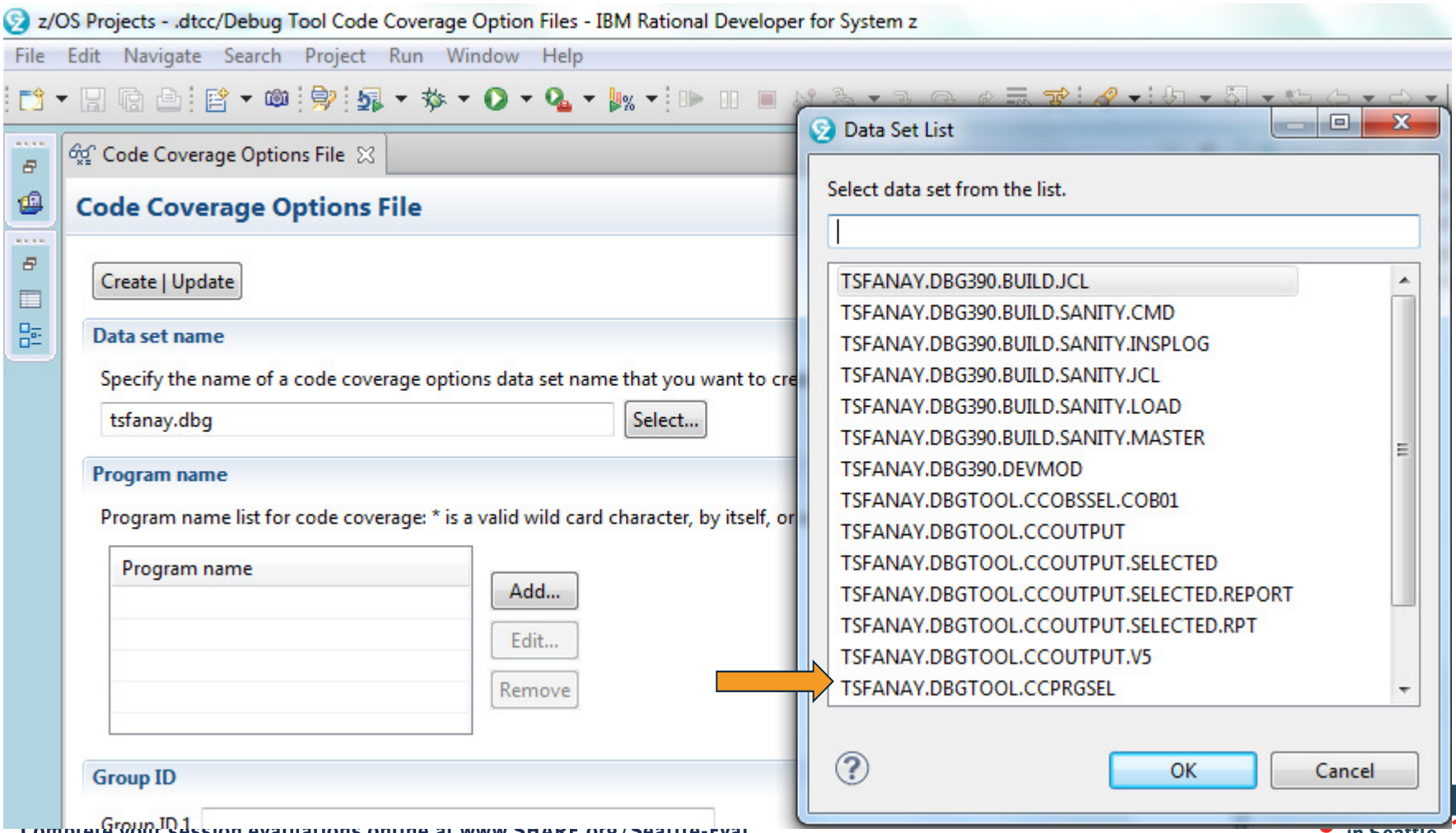
```
<GROUPID1>GROUP11</GROUPID1>
<GROUPID2>CICS1</GROUPID2>
<EXTNAME>ABCD</EXTNAME>
<EXTNAME>EFGI</EXTNAME>
```

Specifying the option file

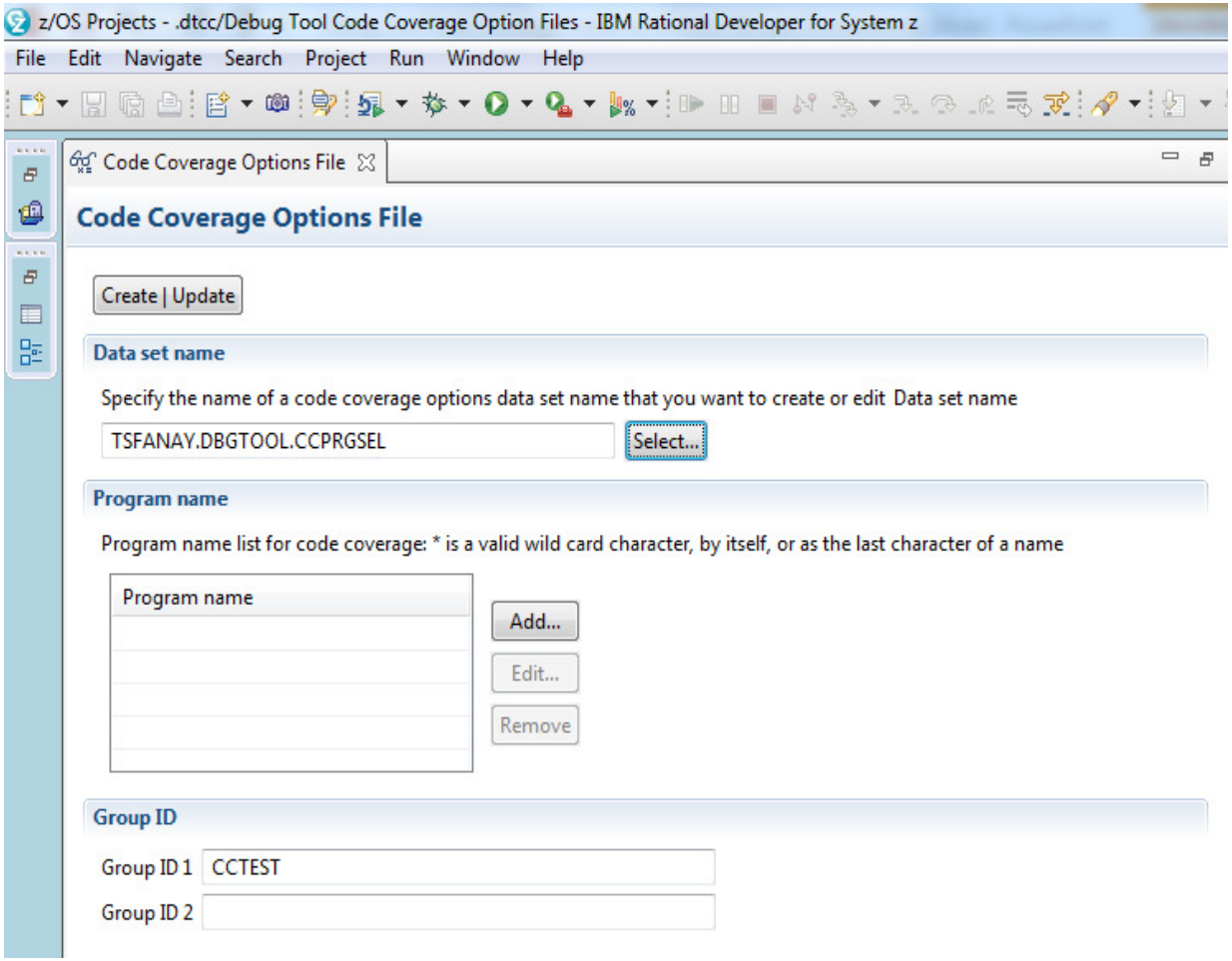


You don't need to type entire name, you could just click on Select

Expanded file selection list



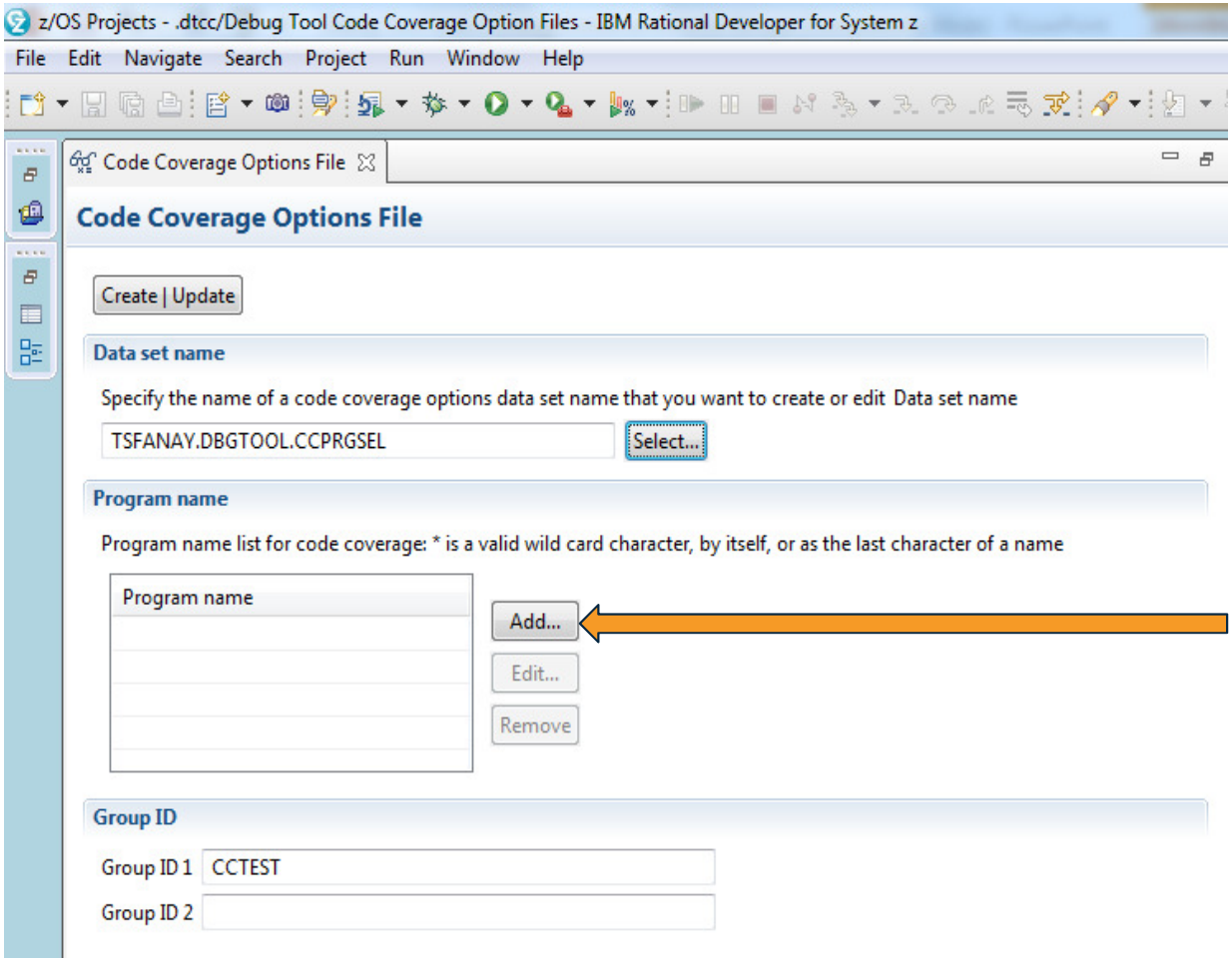
Specifying the option file



Next step is to specify the program(s) :

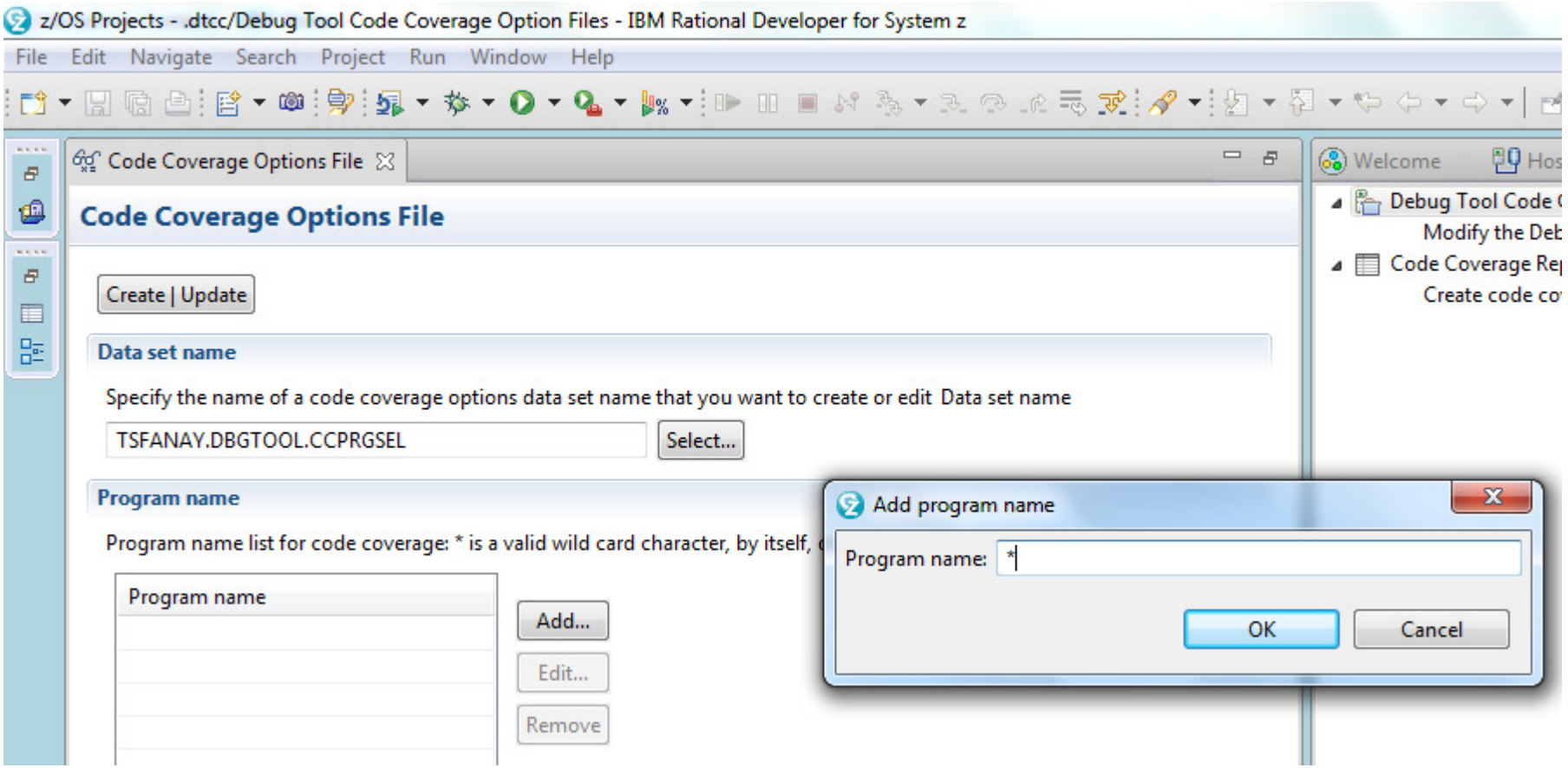
- An * indicates all programs in the job or transaction
- You can use the * as part of a name
  - BAC*
  - ABCDE*
- Or you can specify the full name
  - BAC007
  - ABCD08

Specifying the program name.



Click on Add to specify the program name

Specifying the program name.



For this example we specify *, meaning all executing programs in the job

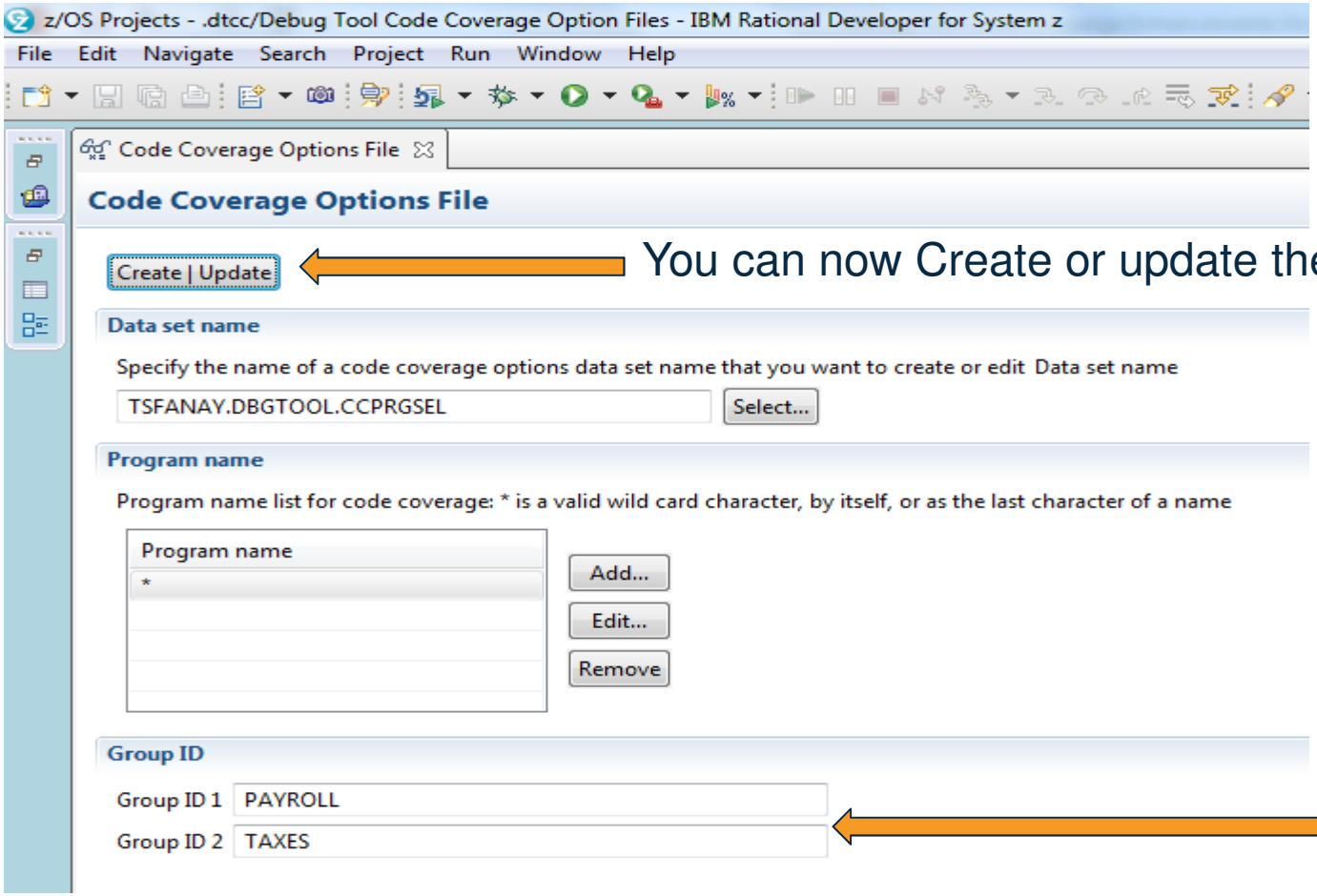
Specifying a Group(s)

What are groups? Groups are optional but they can be use to collect observations from several runs with the same group.

For example, you can have a Group named PAYROLL and a Group named TAXES You can run TRAN1, TRAN2, TRAN3 for example as part of group PAYROLL then you can create a consolidated report of the code coverage for all three transactions by specifying when creating a report the group name. In this example PAYROLL

You can run TRAN4, TRAN5 with group PAYROLL and TAXES and when creating a report if you specify PAYROLL and TAXES as groups then only Code Coverage observations for TRAN4 and TRAN5 are included in the report.
If instead you specify group PAYROLL when creating a report then all transactions are included: TRAN1, TRAN2, TRAN3, TRAN4, and TRAN5

Specifying a Groups PAYROLL and TAXES

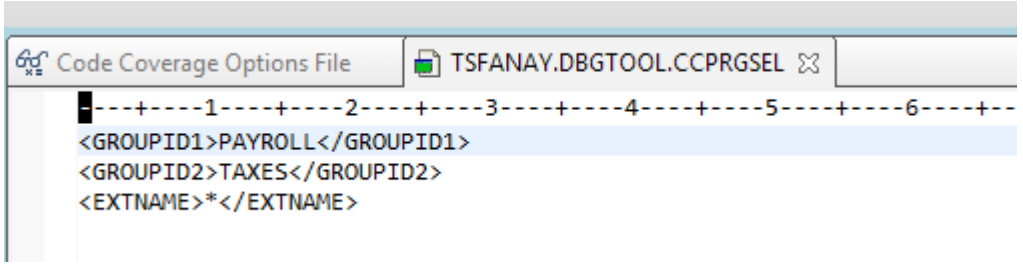


You can now Create or update the file by clicking on

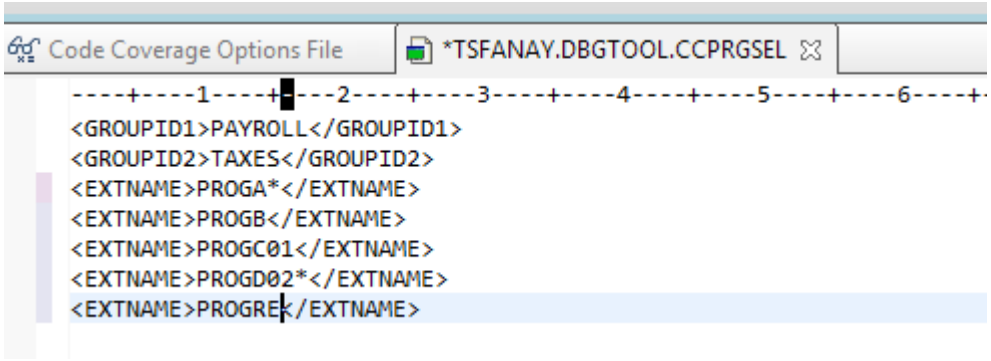
Groups

The option file

You can edit the option file and see the contents after it has been updated by the CC plug-in.



You can also modify the contents manually or programmatically by adding additional program names For example:



Note: You can only have two groups.

Setting up Debug Tool to capture code coverage for your application

Customizing Debug Tool for code coverage capture

The customization required to have Debug Tool capture Code Coverage data is not much different than the one for debugging the same application. Here are the customization steps:

- Define new EQAOPTS commands
  - CCOUTPUTDSN
  - CCOUTPUTDSNALLOC
  - CCPROGSELECTDSN
- Provide location of output file
- Modify TEST runtime option by specifying CC environment variable
  - EQA_STARTUP_KEY

New EQAOPTS

- CCOUTPUTDSN
  - Use to specify location of the output file
- CCOUTPUTDSNALLOC
  - Use to specify the file attributes of the output file. This is optional if you have already allocated the output file.
  - Debug Tool will try to open the output file specified in CCOUTPUTDSN and if it fails it will then use CCOUTPUTDSNALLOC to create the output file.
- CCPROGSELECTDSN
  - Use to specify the location of the option file.
  - The option file is where you specify the program(s) or groups(s) for which Debug Tool need to provide CC observations.

These EQAUOPTS commands are documented in:
IBM Debug Tool Reference and Messages.
Chapter 6. EQAUOPTS command.

The CC output file

- Central repository for CC observations created using:
  - MFI mode
  - RD/z
  - PD Tools Studio
  - WEB/Mobile Facility
- Repository can be by user or global. It is a PDS or a PDSE
  - By user. For example: “&&USERID.DBGTOOL.CCOUT”
  - Global. For example: “ACCOUNT.DBGTOOL.CCOUT”
- You can indicate what repository to use when viewing observations

Modifying the TEST runtime option in your JCL

In order to tell Debug Tool to gather CC observations the EQA_STARTUP_KEY Environment variable is used. Here is an example of specifying the TEST runtime option together with the environment variable using CEEOPTS DD
You can specify the following values in the environment variable:

- CC: The debug session runs in unattended mode
- DCC: The debug session runs in interactive mode

Running un-attended mode for better performance:

```
//CEEOPTS   DD *
TEST(ALL,*,PROMPT,),ENVAR("EQA_STARTUP_KEY=CC")
```

Running a RD/z debug session in parallel with a code coverage gathering session

```
//CEEOPTS   DD *
TEST(ALL,*,PROMPT,TCPIP&9.80.20.3:*),ENVAR("EQA_STARTUP_KEY=DCC")
```

Modifying your JCL

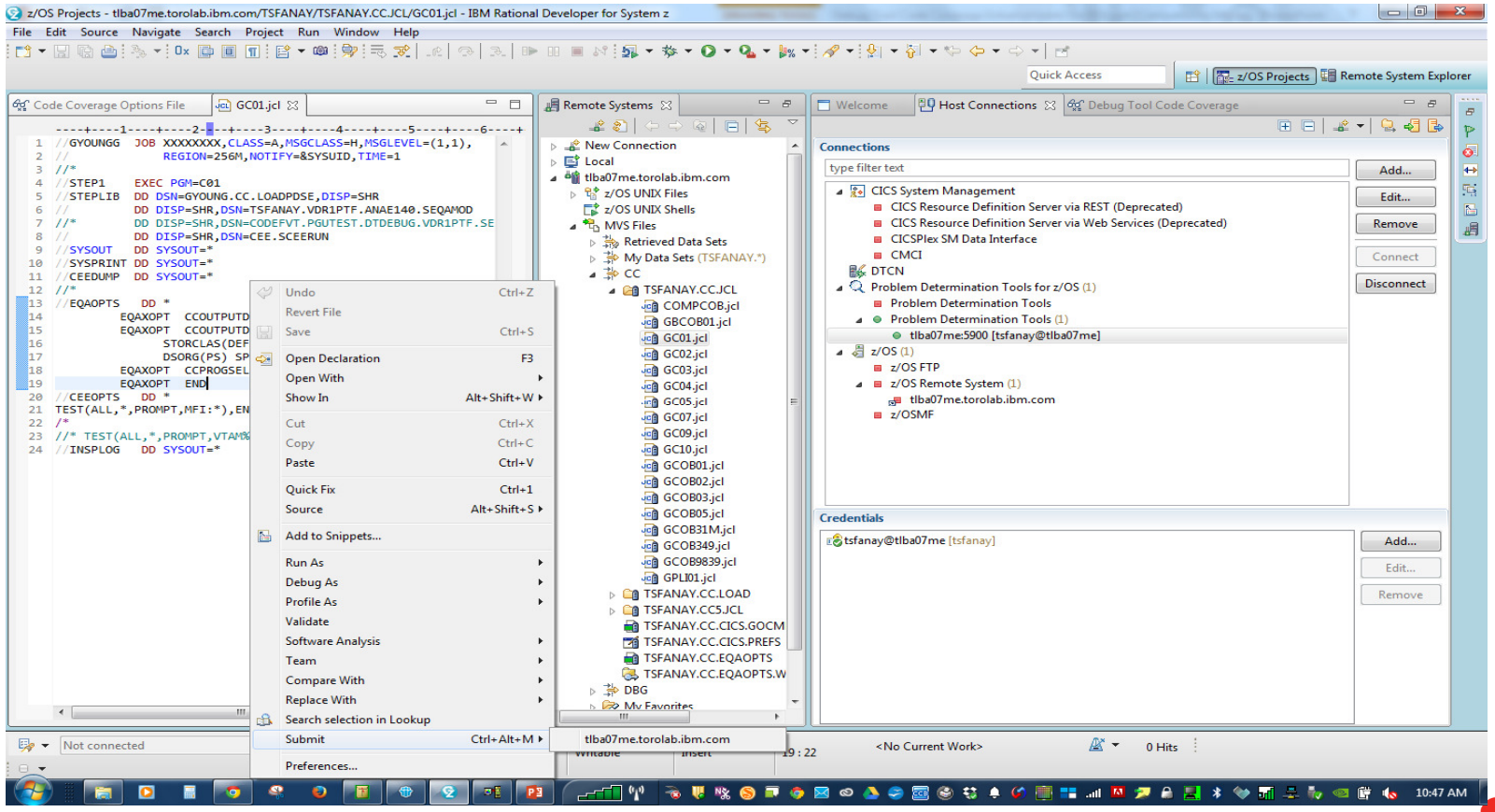
Below is an example of defining EQAOPTS and EQA_STARTUP_KEY environment variable in your JCL

Code Coverage Options File | GC01.jcl

```
----+----1----+----2----+----3----+----4----+----5----+----6----+
1  //GYOUNGG  JOB XXXXXXXX,CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
2  //             REGION=256M,NOTIFY=&SYSUID,TIME=1
3  //*
4  //STEP1    EXEC PGM=C01
5  //STEPLIB  DD DSN=GYOUNG.CC.LOADPDSE,DISP=SHR
6  //         DD DISP=SHR,DSN=TSFANAY.VDR1PTF.ANAE140.SEQAMOD
7  //*        DD DISP=SHR,DSN=CODEFVT.PGUTEST.DTDEBUG.VDR1PTF.SE
8  //         DD DISP=SHR,DSN=CEE.SCEERUN
9  //SYSOUT   DD SYSOUT=*
10 //SYSPRINT DD SYSOUT=*
11 //CEEDUMP  DD SYSOUT=*
12 //*
13 //EQAOPTS   DD *
14          EQAXOPT  CCOUTPUTDSN,'&&USERID.DBGTOOL.CCOUTPUT'
15          EQAXOPT  CCOUTPUTDSNALLOC,'MGMTCLAS(STANDARD)
16                STORCLAS(DEFAULT) LRECL(255) BLKSIZE(0) RECFM(
17                DSORG(PS) SPACE(2,2) CYL'
18          EQAXOPT  CCPROGSELECTDSN,'TSFANAY.DBGTOOL.CCPRGSEL'
19          EQAXOPT  END
20 //CEEOPTS   DD *
21 TEST(ALL,*,PROMPT,MFI:*),ENVAR("EQA_STARTUP_KEY=CC")
22 /*
23 //* TEST(ALL,*,PROMPT,VTAM%GYOUNG:*),ENVAR("EQA_STARTUP_KEY=D
24 //INSPLOG   DD SYSOUT=*
```

You should be ready now to launch Debug Tool so it can collect CC observations in unattended mode. RMC and select submit to start the job

Submit job


z/OS Projects - tlba07me.torolab.ibm.com/TSFANAY/TSFANAY.CC.JCL/GC01.jcl - IBM Rational Developer for System z
File Edit Source Navigate Search Project Run Window Help
Quick Access
z/OS Projects
Remote System Explorer
Code Coverage Options File
GC01.jcl
Remote Systems
Welcome
Host Connections
Debug Tool Code Coverage
Undo Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Quick Fix Ctrl+1
Source Alt+Shift+S
Add to Snippets...
Run As
Debug As
Profile As
Validate
Software Analysis
Team
Compare With
Replace With
Search selection in Lookup
Submit Ctrl+Alt+M
Preferences...
tlba07me.torolab.ibm.com
Connections
Credentials
Not connected
<No Current Work>
0 Hits
10:47 AM

Contents of the Output file generated by Debug Tool

Code Coverage Optio... | Code Coverage Report... | TSFANAY.DBGTOOL.CC...

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
<DTCODECOVERAGEFILE>
<RUNDATE>
<YEAR>2014</YEAR>
<MONTH>12</MONTH>
<DAY>18</DAY>
</RUNDATE>
<RUNTIME>
<HOURS>14</HOURS>
<MINUTES>46</MINUTES>
<SECONDS>56</SECONDS>
</RUNTIME>
<GROUPID1>PAYROLL</GROUPID1>
<GROUPID2>TAXES</GROUPID2>
<USERID>TSFANAY</USERID>
<LOADMODULE>
<MEMBERNAME>COB01</MEMBERNAME>
<COMPILATIONUNIT>
<PROGRAMDSNAME>GYOUNG.CC5.LOADPDSE(COB01)</PROGRAMDSNAME>
<PROGRAMDSTYPE>4</PROGRAMDSTYPE>
<COMPILEDATE>
<YEAR>2014</YEAR>
<MONTH>12</MONTH>
<DAY>01</DAY>
</COMPILEDATE>
<COMPILETIME>
<HOURS>19</HOURS>
<MINUTES>16</MINUTES>
<SECONDS>00</SECONDS>
</COMPILETIME>
<CSECT>
<EXTNAME>COB01A</EXTNAME>
<DBGOV>N</DBGOV>
<EXECUTED>43.1 46.1 47.1 50.1 51.1 52.1 56.1 58.1 63.1 65.1</EXECUTED>
<EXECUTED>73.1 74.1 75.1 78.1 79.1</EXECUTED>
<UNEXECUTED>60.1 70.1</UNEXECUTED>
</CSECT>
</COMPILATIONUNIT>
</LOADMODULE>
```

- Uses XML . Documented In Debug Tool Users Guide Appendix D, XML Tags for Code Coverage.
- It is a repository so more than one run is stored in the same file.
- Example shows the CC Observations for COB01

Non-Batch Environments

- You can gather CC observations in all environments supported by Debug Tool
  - CICS
  - IMS
  - DB2
  - USS
  - TSO
- The setup for above environments is basically the same as for batch and the only differences are those required for debugging an application with Debug Tool.

CICS

Below is an example of using DTCN to define a debug profile where Debug Tool gathers CC observations :

```
DTCN              Debug Tool CICS Control - Menu 2                S07CICPB

 Select Debug Tool options

 Test Option     ==> TEST              Test/Notest
 Test Level      ==> ERROR             All/Error/None
 Commands File   ==> GYOUNG.CC.CICS.GOCMD
 Prompt Level    ==> PROMPT
 Preference File ==> *

 EQAOPTS File    ==> GYOUNG.CC.EQAOPTS

 Any other valid Language Environment options
 ==> ENVAR("EQA_STARTUP_KEY=CC")

 PF1=HELP 2=GHELP 3=RETURN
```

You need to specify a GO command in a commands file and you need to provide the new EQAOPTS as well

Contents of file:
GYOUNG.CC.CICS.GOCMD

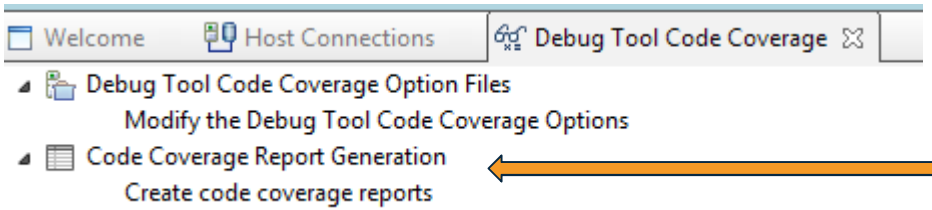
```
----+----1----+----2----+----3----+----
      GO;
```

Contents of file:
GYOUNG.CC.EQAOPTS

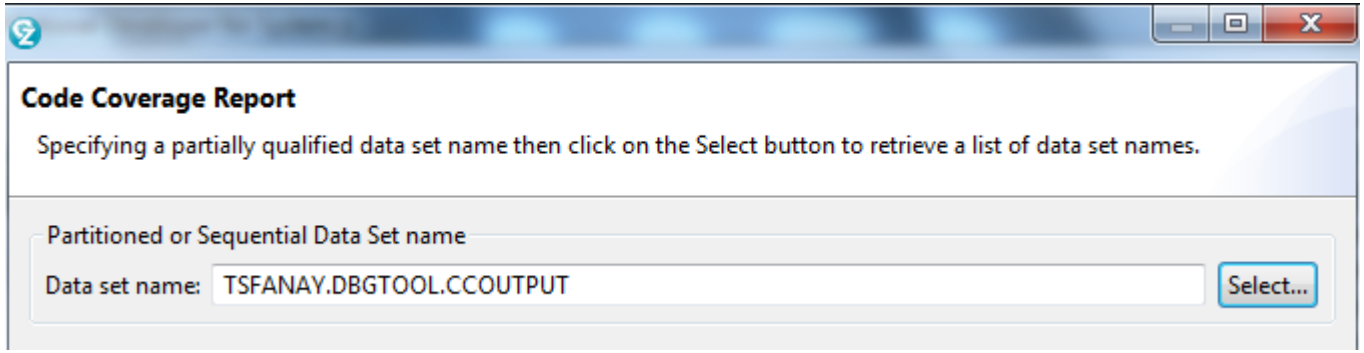
```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
      EQAOPT  CCOUTPUTDSN,'&&USERID.DBGTOOL.CCOUTPUT'
      EQAOPT  CCOUTPUTDSNALLOC,'MGMTCLAS(STANDARD)                    +
            STORCLAS(DEFAULT) LRECL(255) BLKSIZE(0) RECFM(V,B)        +
            DSORG(PS) SPACE(2,2) CYL'
      EQAOPT  CCPROGSELECTDSN,'&&USERID.DBGTOOL.CCPRGSEL'
      EQAOPT  END
```

Viewing the Code Coverage Observations

Go back to the Code Coverage Main View

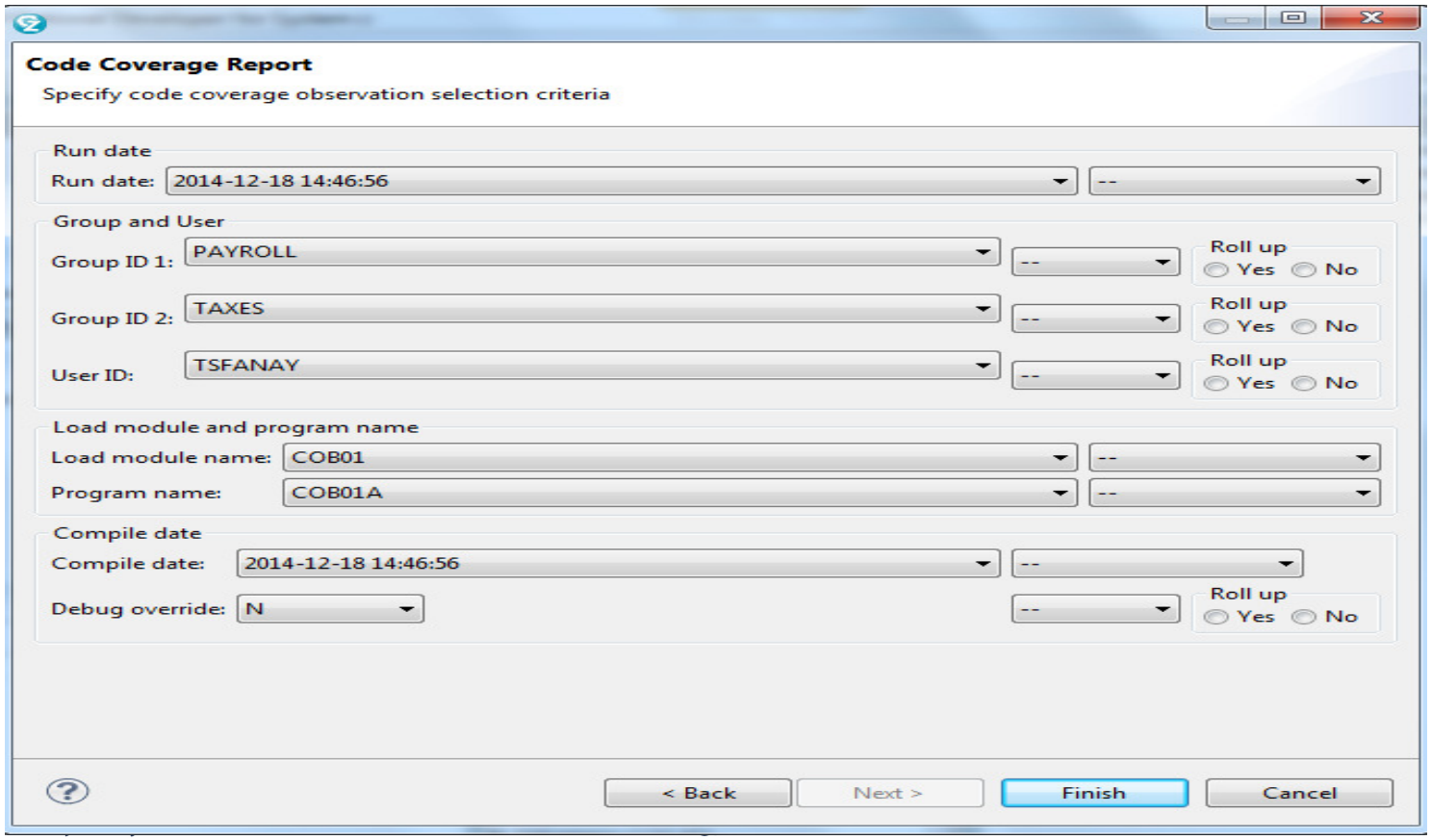


Select Code Coverage Report Generation and the provide location of CC repository (Output file).

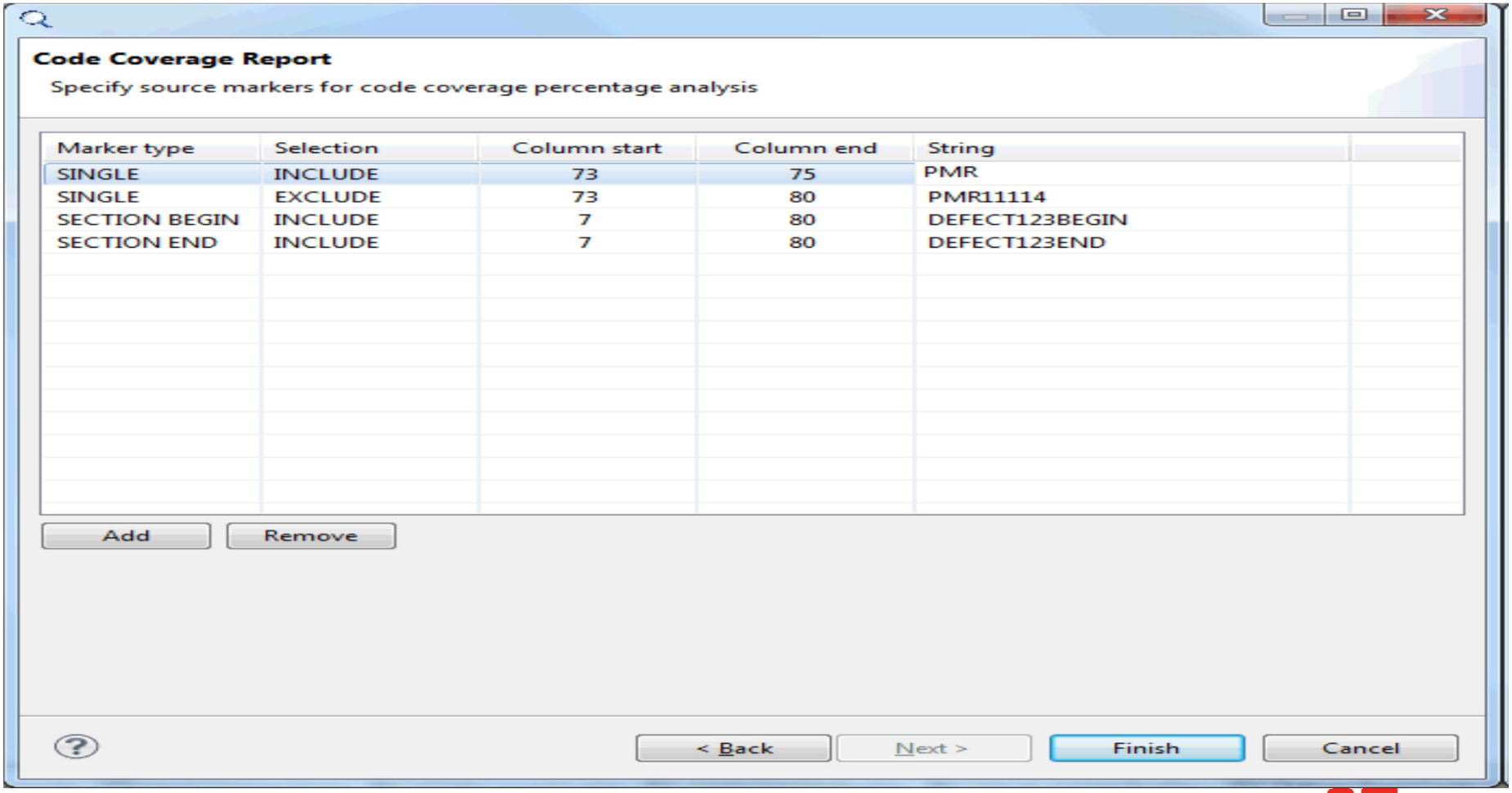


Code Coverage Report

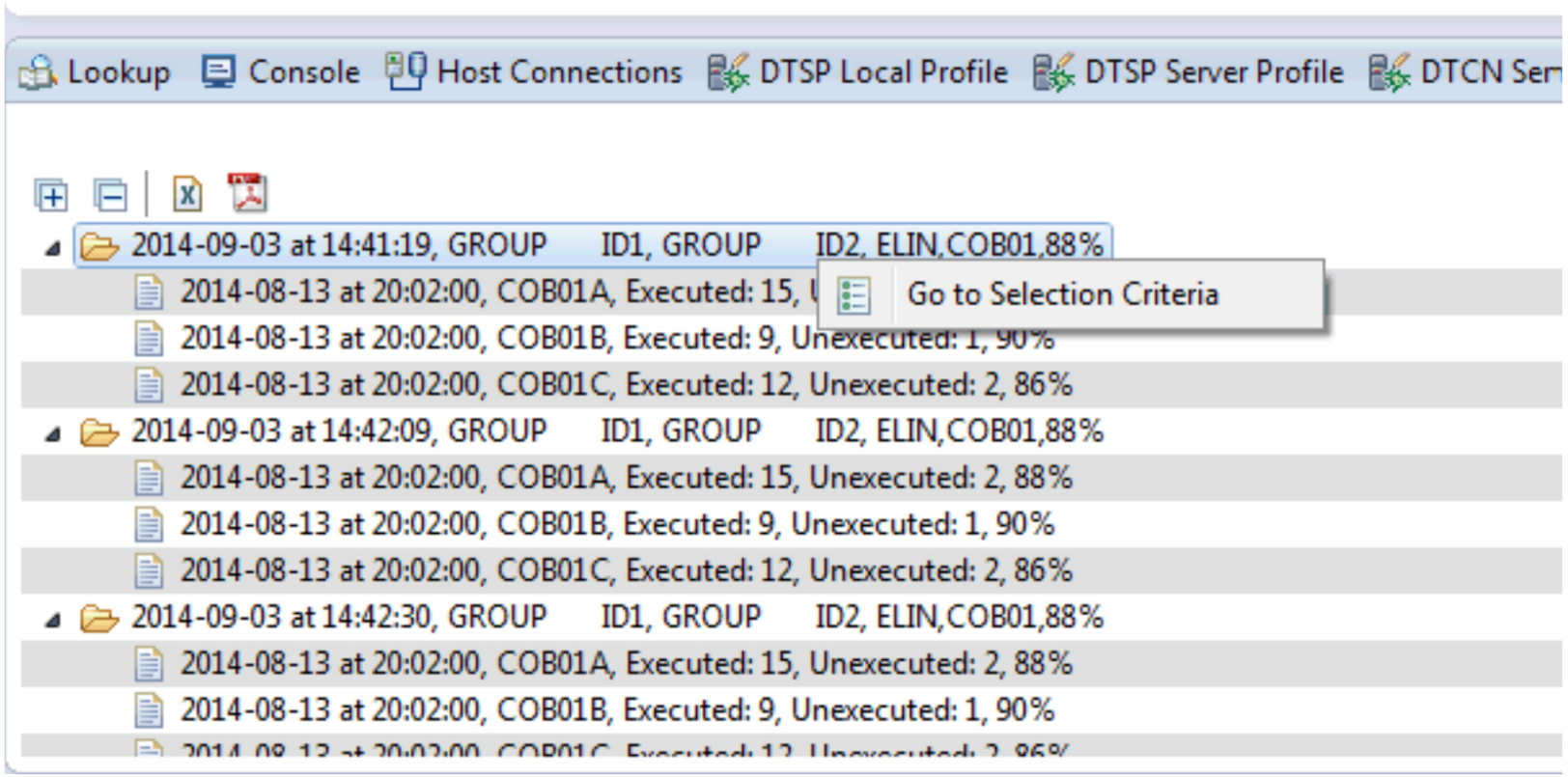
The Code Coverage report view provide all selectable fields in the repository for the creation of the report. It is here that you customize your report to fit your needs.



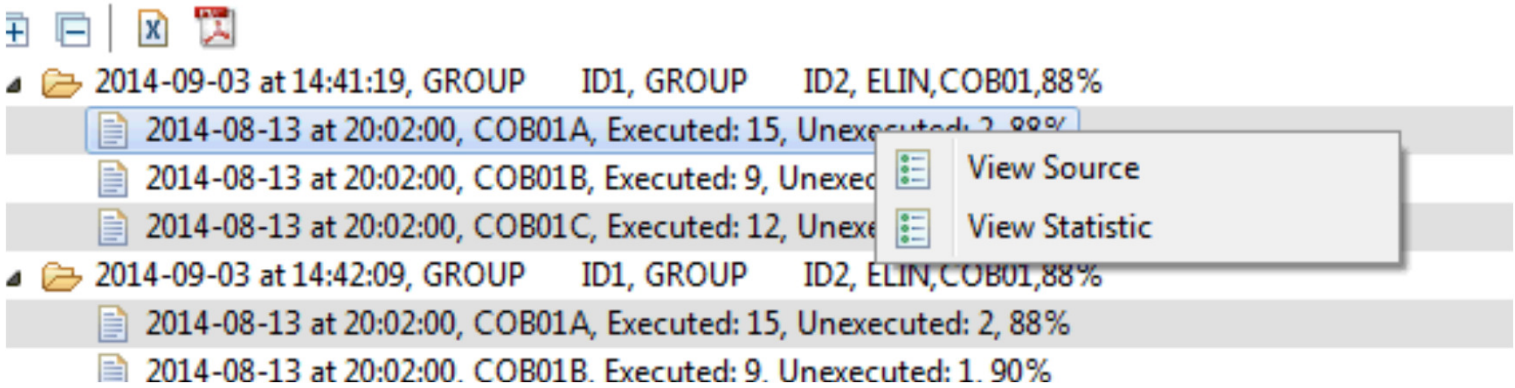
Code Coverage Report

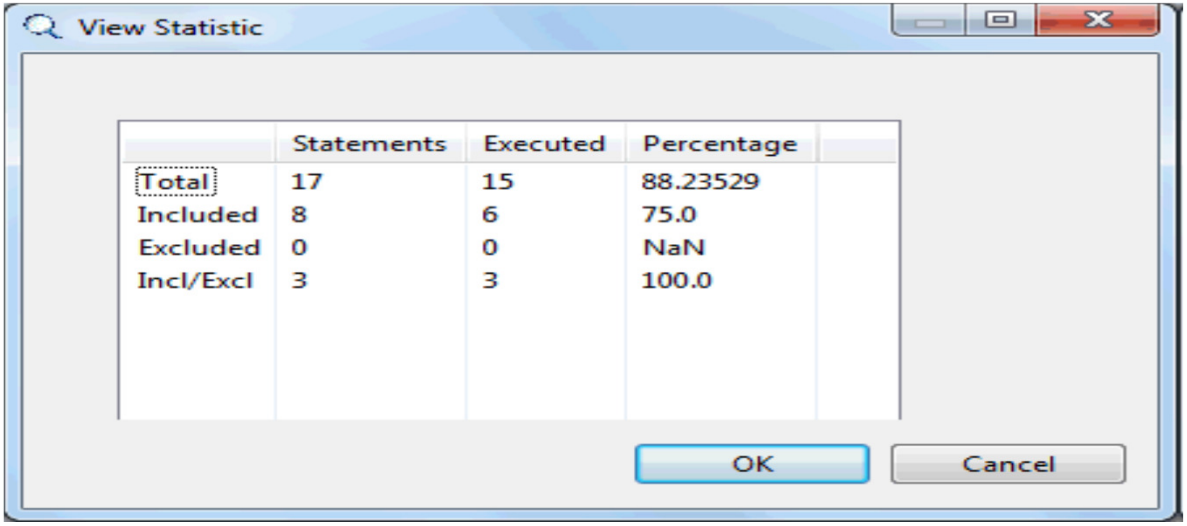


Code Coverage Reports Repository

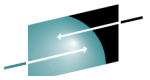


Viewing Code Coverage Reports





Notices and Disclaimers



U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and Disclaimers (con't)

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 in connection with this publication 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Bluemix, Blueworks Live, CICS, Clearcase, DOORS®, Enterprise Document Management System™, Global Business Services ®, Global Technology Services ®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, SoDA, SPSS, StoredIQ, Tivoli®, Trusteer®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.