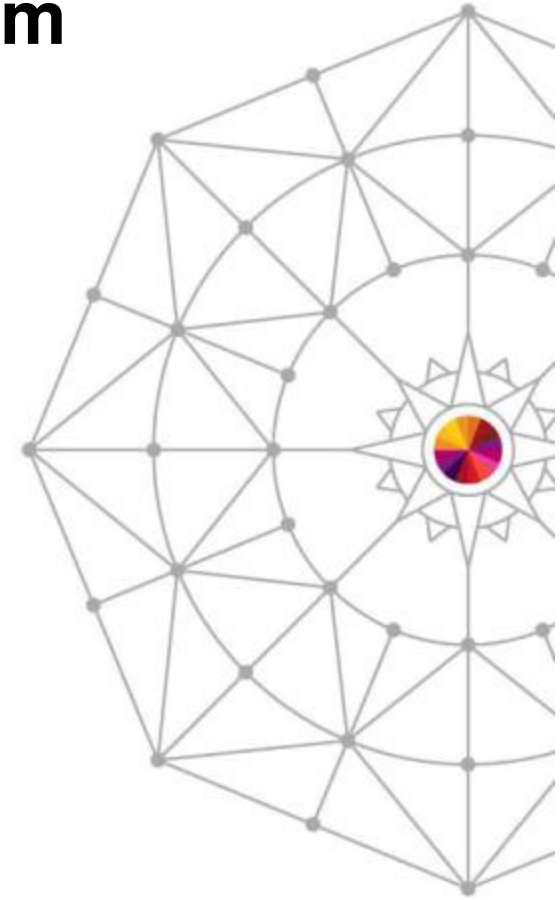
#SHAREorg

SHARE

What's New In the IBM Problem Determination Tools

Francisco M Anaya
IBM Problem Determination Tools Architect
Randy Campbell
IBM Debug Tool Developer

March 10, 2014
Session 14621



SHARE in Anaheim

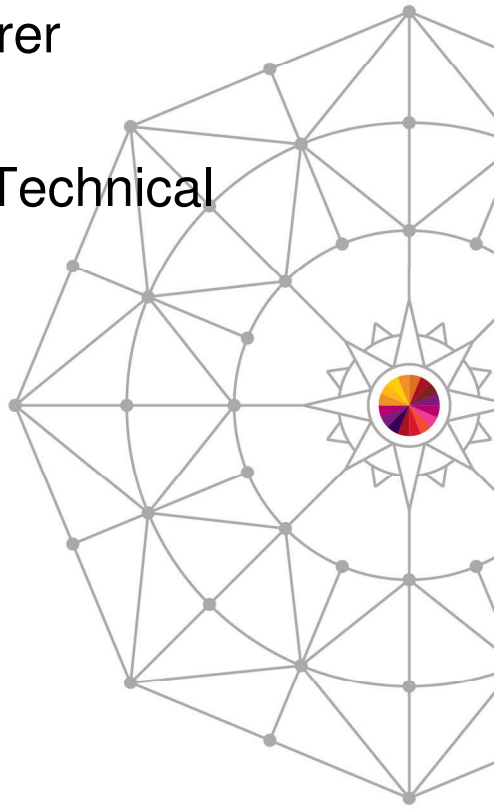
Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.
IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.
NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Agenda

- What are the IBM Problem Determination Tools
- Functional Alignment with CICS Tools and z/OS Explorer
- Solution Packs
- Problem Determination Tools for Multi-platforms V1.0 Technical Preview
- New Debug Tool Code Coverage
- New Fault Analyzer Web Interface
- COBOL V5 Support
- Playback Support for RD/z
- JCL Instrumentation Eclipse plug-in
- IMS private message region enhancement



IBM Problem Determination Tools Products

#SHAREorg

SHARE

Products

Application Performance Analyzer for z/OS
A non-intrusive application performance analyzer that aids developers in the design, development and maintenance cycles.

Debug Tool for z/OS
IBM Debug Tool for z/OS helps examine, monitor, and control the execution of application programs.

Fault Analyzer for z/OS
Helps developers analyze and fix application and system failures. Fault Analyzer gathers information about an application and the surrounding environment.

File Manager for z/OS
Provides comprehensive, user-friendly tools for working with z/OS data sets, DB2®, CICS® or IMS™ data, or MQ queues. These tools include the familiar browse, edit and copy.

Workload Simulator for z/OS and OS/390
Enables you to conduct stress, performance, regression, function and capacity planning tests, while eliminating the need for large amounts of terminal hardware.

Problem Determination Tools Studio and Plug-ins
Provides easy access through a graphical user interface (GUI) to the z/OS Problem Determination Tools.

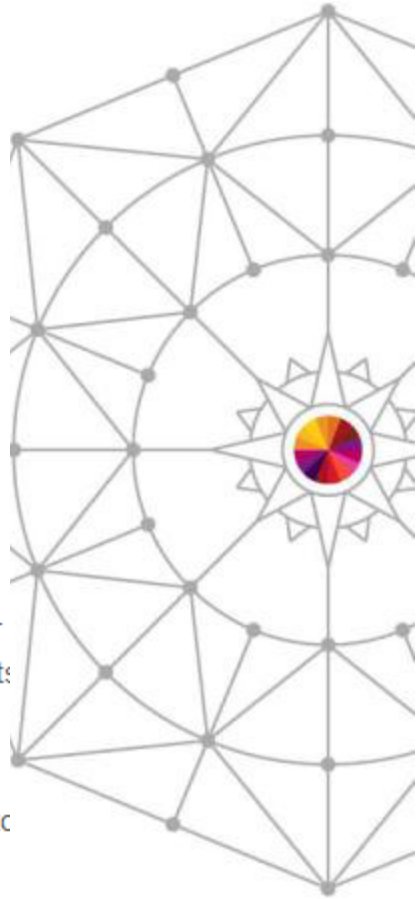
Problem Determination Solution Pack
Convenient package to help address your problem analysis needs.

Problem Determination Testing Solution Pack
Convenient package to help address your testing needs.

Data Set Commander for z/OS
Extends interactive and batch capabilities for performing operations on partitioned data sets and their members.

Hourglass
Allows sites to alter the Date/Time returned to a z/OS application when a time request is made (SVC 11 or PC Time Requests).

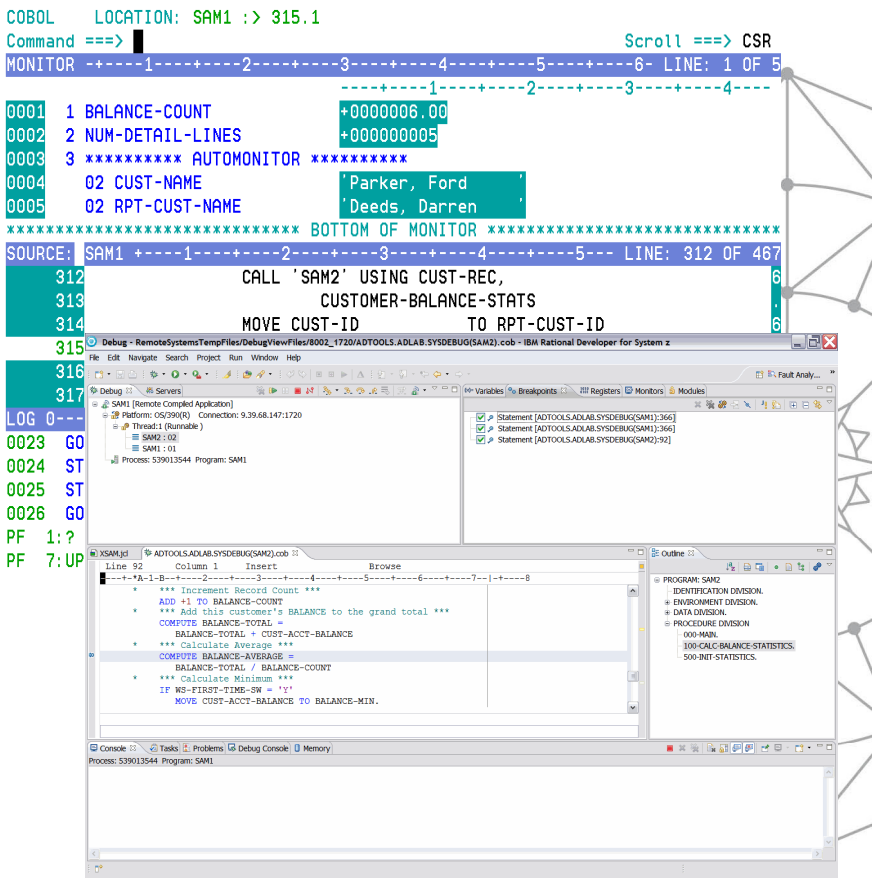
Migration Utility for z/OS
Generates IBM COBOL applications from programs written in the CA Easytrieve® languages.



SHARE in Anaheim

IBM Debug Tool Version 13 (Announce: October 1, GA: October 18)

- IBM Debug Tool can help you increase debugging efficiencies and reduce application development cycle times.
- Program testing and analysis aid that helps you examine, monitor, and control the execution of application programs on z/OS (CICS/DB2/IMS/ COBOL/PLI/ASM,C/C++/ASM/JAVA w Toolkit)
- Code Coverage Facilities
- Proven 3270-based interface
- Eclipse based GUI
- Support for RD/z

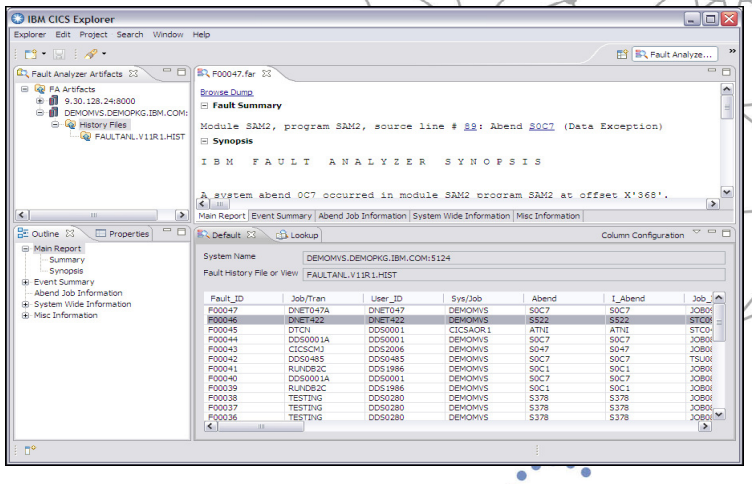


IBM Debug Tool Version 13

What's new?

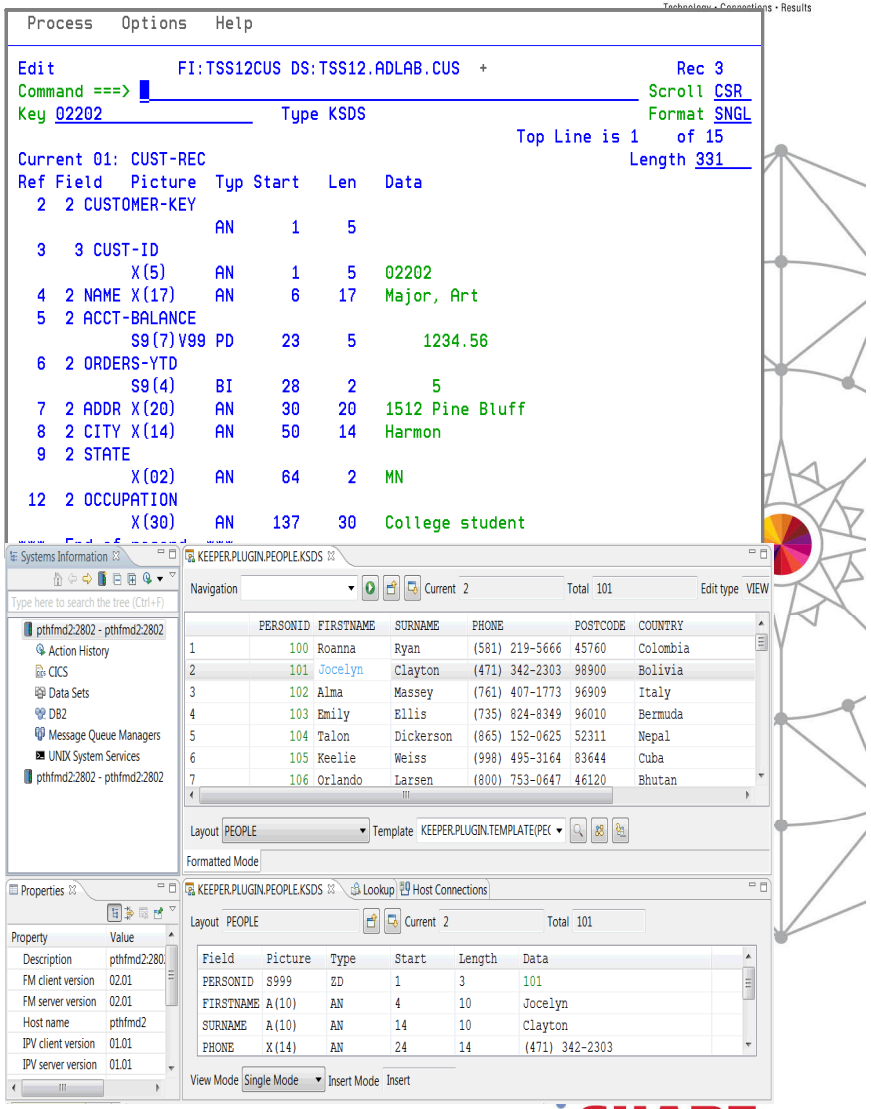
- Smooth debugging mixed apps JAVA/COBOL/PLI
- Improved Code Coverage. Support for E PL/1
- Automatic start IMS MPP regions, dynamic routing of transactions
- IMS Startup Cross Reference table for WASz and Web started transactions
- RDz Playback
- Support for DB2 V11, IMS V13, CICS TS 5.1, z/OS 2.1, WASz 8.5,, COBOL V5.1, Enterprise PLI V4.4, C/C++ for z/OS 2.1, RDz 9.0
- JCL Instrumentation plugin

```
File  View  Services  Help

Synopsis                                                    Line 12 Col 1 80
Command ===> _____________________________________________ Scroll ===> CSR
JOBNAME: DNET246S  SYSTEM ABEND: 0C7          DEMOMVS   2007/10/30  20:00:12
  Source
  Line #
  000088        *    *** Add this customer's BALANCE to the grand total ***
  000089             COMPUTE BALANCE-TOTAL =
  000090                BALANCE-TOTAL + CUST-ACCT-BALANCE

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

  Source
  Line #
  000059             05  CUST-ACCT-BALANCE      PIC S9(7)V99  COMP-3.
  000066             05  BALANCE-TOTAL        PIC S9(7)V99  COMP-3.

Data field values at time of abend:

  BALANCE-TOTAL     = 10948.44
  CUST-ACCT-BALANCE = X'7C7B5B6C50'  *** Cause of error ***

*** Bottom of data.
```



IBM File Manager Version 13 (Announce: October 1, GA: October 18)

- **IBM File Manager** allows you to manage production, test, and development data across multiple formats and storage media.
- Create, edit, copy, browse, extract, print, and compare enterprise data (VSAM/DB2/IMS,CICS/MQ)
- Proven 3270-based interface and free graphica user interface.



What's new?

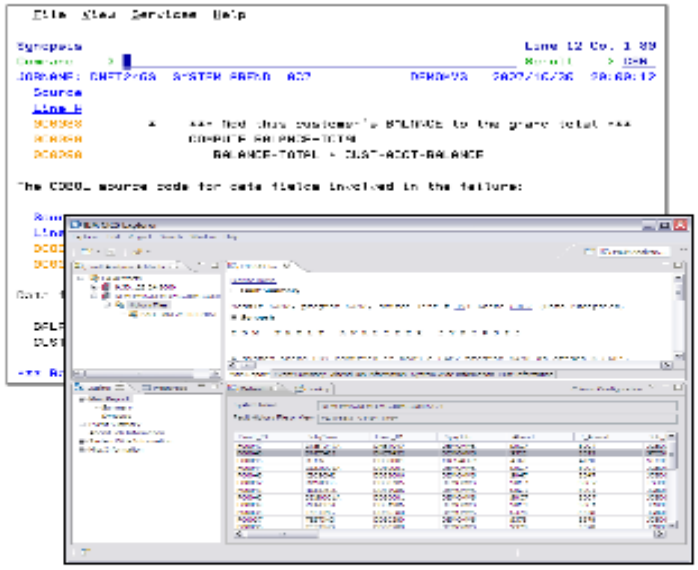
- IMS GUI
- FM DB2 enhancements
- Batch template and update support
- UNICODE and various CCSIDs support
- Support for DB2 V11, IMS V13, CICS TS 5.1, MQ8.5, z/OS2.1, PL/I 4.4,
- DB2 Large Object (LOB) including XML columns
- A range of other enhancements to address key customer requirements

IBM Fault Analyzer Version 13 (Announce: October 1, GA: October 18)

- **IBM Fault Analyzer** improves developer productivity and decreases deployment costs by helping to analyze and correct application failures quickly (CICS/DB2/IMS/MQ/COBOL/PLI/ASM/ C/C++/ASM/JAVA).
- Develop and test new and existing applications more productively, helping to reduce costs along the way.
- Proven 3270-based interface and free graphical user interface.

What's new?

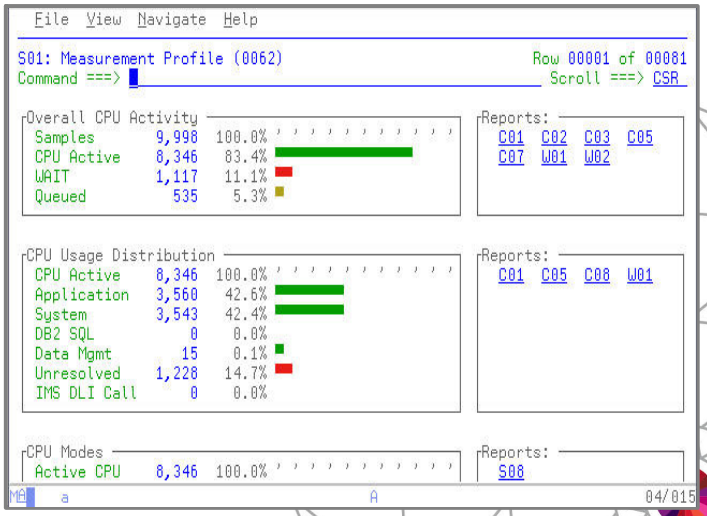
- Optimize
  - Sidefile checking and reporting
  - Customer specific module and abend descriptions
  - ISPF Virtual Relief
- Modernize
  - Enhanced web interface using Dojo
- Enhanced Java support
- Support for DB2 V11, IMS V13, CICS TS 5.1, z/OS 2.1, COBOL V5, PL/I V4.4, Java V7
- A range of key customer requirements

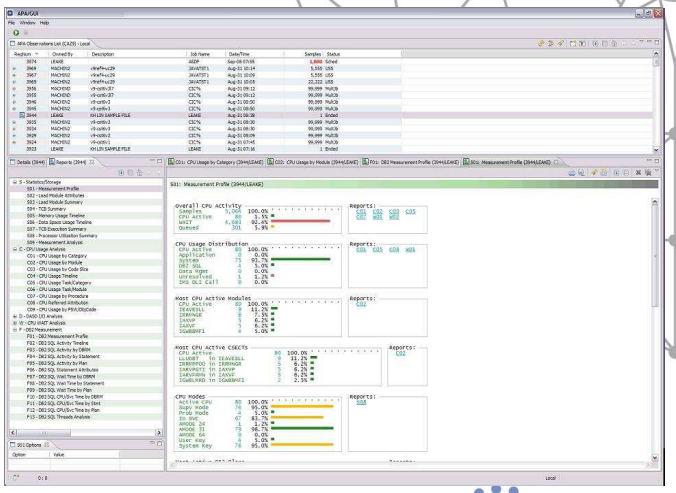


Helps to identify the cause, analyze the failure, and fix the problem

*IBM Application Performance Analyzer Version **13*** (Ann: OCT 1, GA: OCT 18)

- **IBM Application Performance Analyzer** helps maximize the performance of your applications and improve the response time of your online transactions and batch turnaround.
- Identify constraints and improve the entire application's performance no matter where the problem resides (CICS/ IMS/DB2/MQ/COBOL/PLI/ ASM/JAVA)
- Proven 3270-based interface and free graphical user interface.





What new?

- Enhanced support for WebSphere and Java
- Sampling enhancements to improve accuracy of reports
- WAS Phase 2 to support simultaneous sampling of servant regions, CICS and DB2 activity
- Support for DB2 V11, IMS V13, CICS TS V5.1, z/OS 2.1, WASz V8.5, COBOL V5.1, PL/I V4.4, Java V7
- A range of key customer requirements

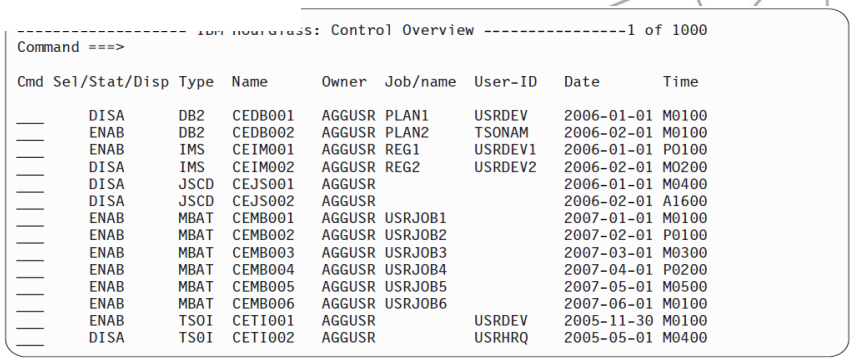
IBM HourGlass V7.1 (Announce: October 1, GA: November 6)

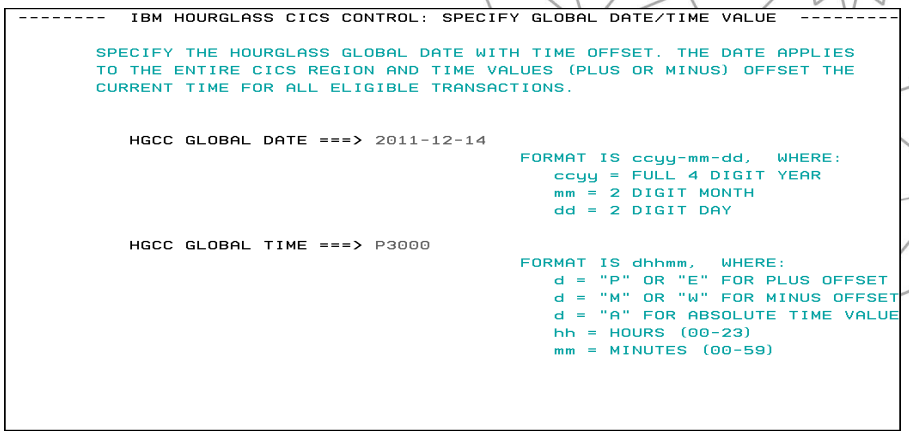
IBM® HourGlass is a leading z/OS® clock simulator for testing mainframe applications. The features and flexibility of HourGlass promote more accurate and reliable application testing with the ability to coordinate full functionality across time zones. HourGlass can simulate past, present or future dates and times without changing application code or computing environments.

- **IBM HourGlass** allows testing of time-sensitive application programs without dedicated individual LPARs or costly IPLs
- Provides support for popular middleware elements such as DB2, IMS, CICS, and COBOL, on the z/OS platform.
- Proven 3270-based interface and batch interface options.

What's new?

- Support for SYSPLEX synchronization
- Additional CICS, IMS invocation commanc options supported
- Dynamic update of repository
- JES exit to ensure time options are valid



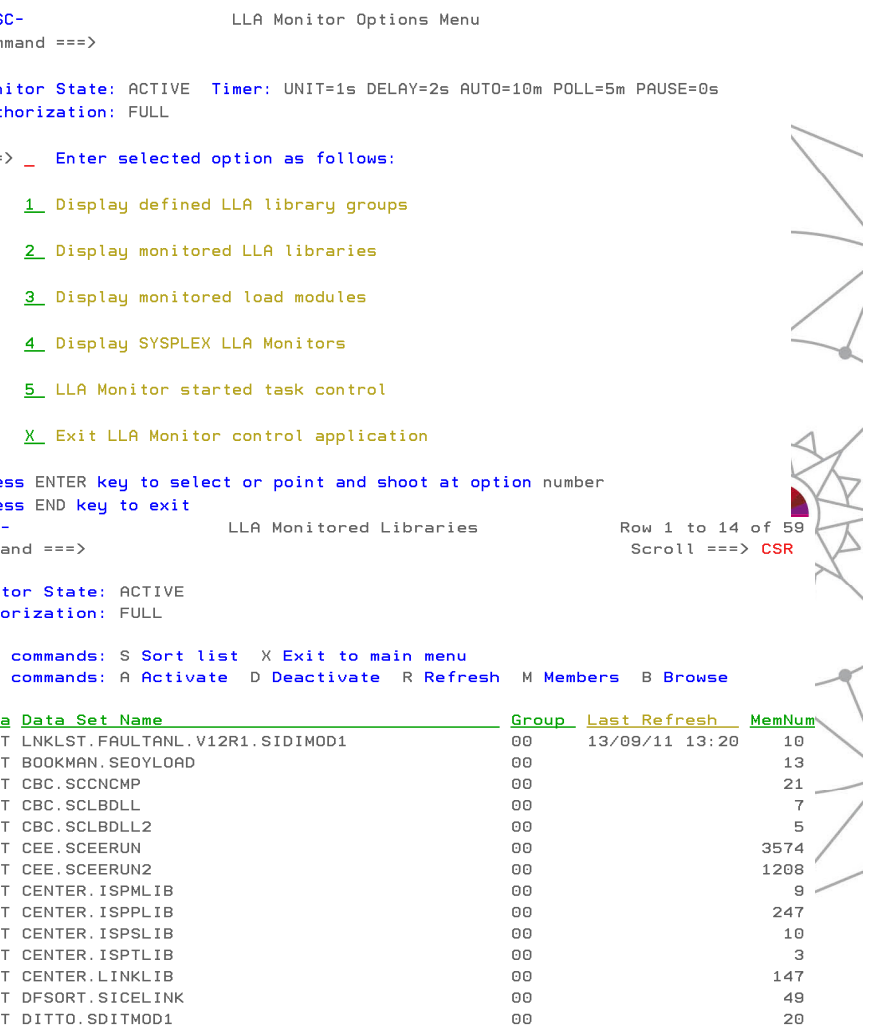


Data Set Commander V8.1 (Announce: October 1, GA: November 6)

- **IBM Data Set Commander** provides efficient tools and shortcuts to improve handling of PDS and PDSE.
- Fully automated facility to manage caching of PDS and PDSE members.
- Fully compatible replacement for IEBCOPY, enhancing capability to manage and manipulate PDS and PDSE.
- Manage PDSE member generations (versions)
- Integrated front end to ISPF. It enhances ISPF functions, including browse, edit, TSO shell, and DSLIST, UNDELETE, and provides many additional capabilities

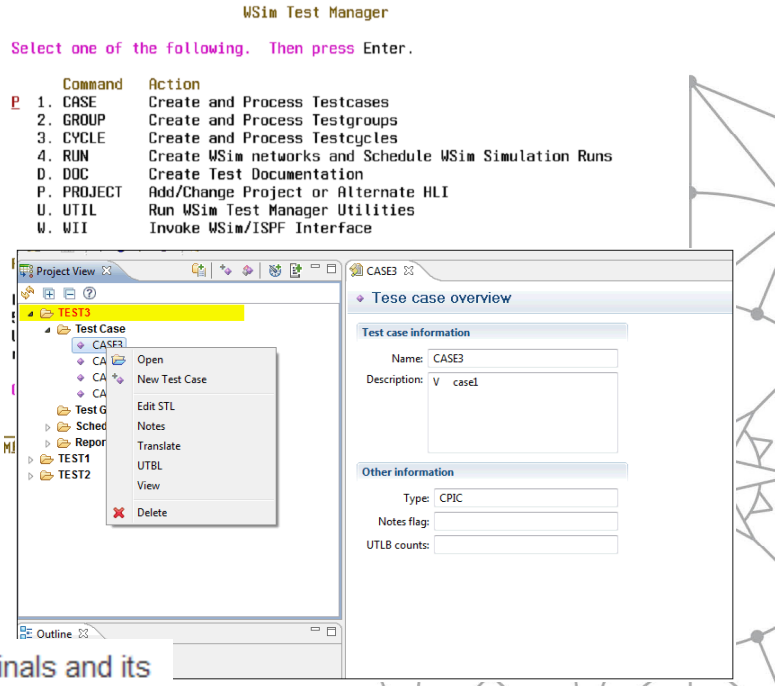
What's new?

- Automated PDS/PDSE caching (LLA)
- Manage PDSE member generations (versions)



IBM Workload Simulator

- **IBM Workload Simulator** helps eliminate the need for large amounts of terminal hardware and terminal operator time for testing.
- Ideal for stress, performance, regression, function, and capacity planning tests.
- Proven 3270-based interface and new free graphical user interface.



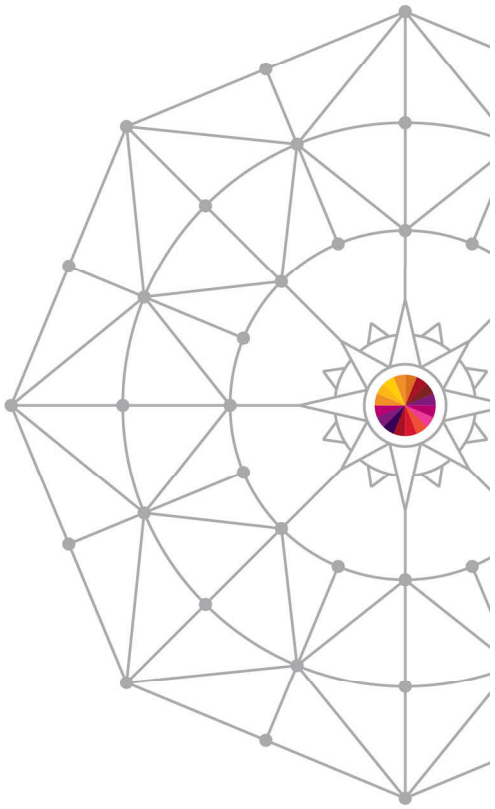
What's new?

- Graphical user interface (GUI) that help you manage testing scripts and testing buckets

IBM® Workload Simulator for z/OS® and OS/390® can simulate a network of terminals and its associated messages. This solution for stress, performance, regression, function and capacity planning testing eliminates the need for large amounts of terminal hardware and operator time by providing a powerful analysis with log list, log compare and response time utilities.

- Provides support for SNA, CPI-C (LU 6.2), TCP/IP, Telnet 3270, 3270E and 5250 clients, Telnet line mode network virtual terminal clients, simple TCP and UDP clients, FTP clients and multiple client applications that run on top of TCP/IP.

CICS Tools, PD Tools and z/OS Explorer Functional Alignment



Eclipse-based integration platform for z/OS tools

IBM® Explorer for z/OS® (z/OS Explorer) is an Eclipse-based integration platform for z/OS users that enables the integration of a wide variety of solutions using IBM, vendors, or customer plug-ins. z/OS Explorer is extendable via the IBM repository of compatible products to fulfill each user's roles and responsibilities. For example, z/OS Explorer, powered by IBM product plug-ins from the repository can provide a single workbench with the ability to develop, and test CICS, DB2, WebSphere MQ, IMS, and batch applications and manage related sub-systems.

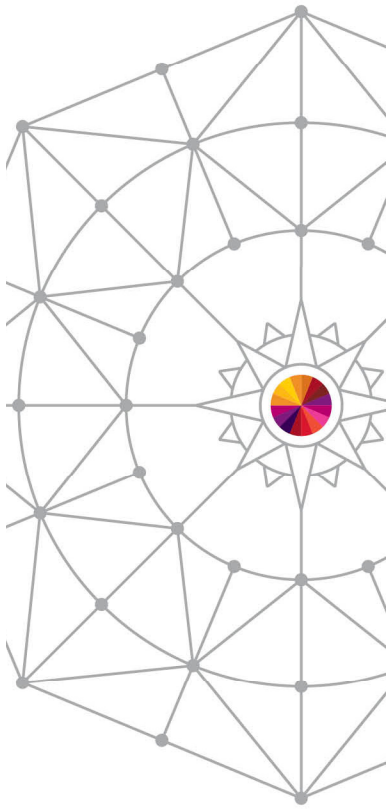
IBM Explorer for z/OS delivers extensible workstation connectivity to key z/OS functions with:

Problem Determination Tools V13 GUIs To Meet Your Needs

- **The GUIs are available with Rational Developer for z, for a full blown application development environment, or in conjunction in a systems environment as part of the PD Tools Studio, CICS Explorer, IMS Explorer on top of a z/OS Explorer base/technology**

- Rational Developer for System z

CICS Explorer, IMS Explorer
or PD Tools Studio
with z/OS Explorer



Solution Packs
Four New Tools Solution Packs

SHARE
Technology • Connections • Results

CICS Optimization Solution Pack for z/OS V5.1

Quickly optimize performance of CICS systems and applications with CICS Interdependency Analyzer, CICS Deployment Assistant, CICS Performance Analyzer, CICS Configuration Manager

CICS Modernization Solution Pack for z/OS V5.1

Understand deployed CICS application and system resources, connect from non-mainframe devices, and create CICS infrastructure to support the increased workload with CICS Interdependency Analyzer, CICS Deployment Assistant, CICS Transaction Gateway for z/OS

IBM Problem Determination Solution Pack for z/OS V1.1

Cost effective problem analysis for z/OS sub-systems and languages with both GUI & 3270 interface with File Manager for z/OS, Fault Analyzer for z/OS, Debug Tool for z/OS, Workload Simulator for z/OS and OS/390, Hour Glass, Data Set Commander

IBM Problem Determination Testing Solution Pack for z/OS V1.1

Reduce testing time, improve application reliability and user diagnosis capabilities, and improving user's ability to regulate and monitor testing activities with Debug Tool for z/OS , Workload Simulator for z/OS and OS/390, Hour Glass

Upgrade from standalone products using trade-ups

Problem Determination Tools for Multi-platforms V1.0 Technical Preview

Why PD Tools for Multiplatforms?

- We have customers beyond z platform - AIX and Linux users.
- Provide a consistent user experience for customers running application on z, AIX and Linux.
- Targeting customers using WebSphere MQ, Java/WAS/Liberty Profile, and TXSeries/WXTR.
- A differentiation from the competition.

PD Tools for Multi-platforms features are available as a technical preview

•Managing data on distributed platforms and analyzing Java dumps

•Features are subject to change at our own discretion.

• Available following this link:

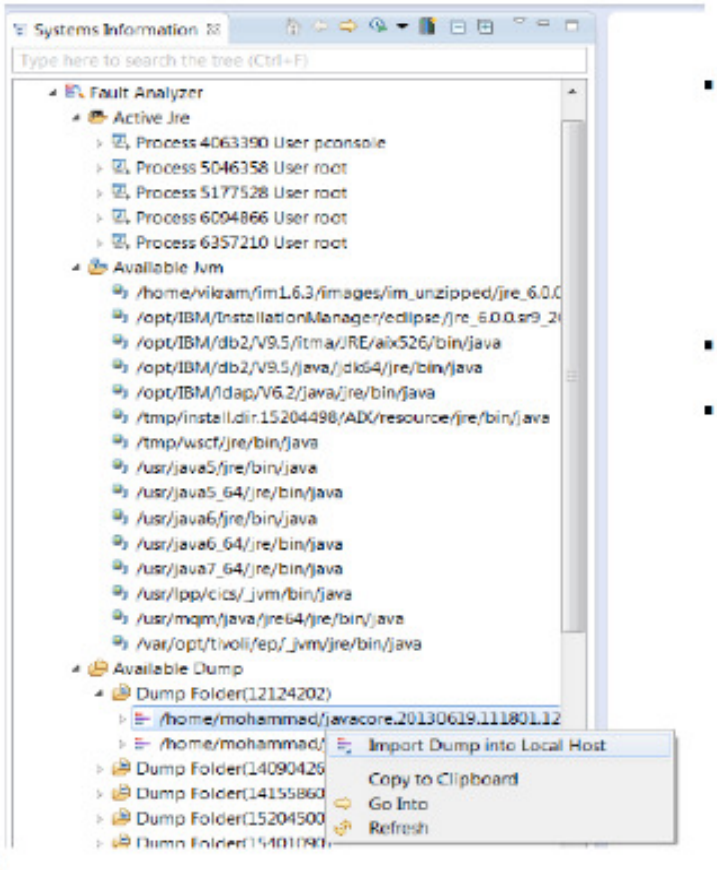
http://www-01.ibm.com/support/docview.wss?uid=swg27040206

Fault Analyzer for Multi-platforms features

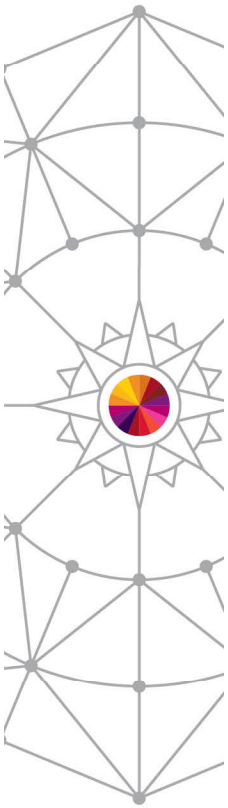
Fault Analyzer for Multiplatforms overview (Java dump analysis)

- Support AIX and Red Hat operating systems on System p.
- Analysis of Java dump files on a local file system.
- Supports analysis of Java dumps from IBM's Java virtual machines. While we only provide support for AIX and Red Hat, Java dumps from any supported platforms should work. Just import Java dumps to your local system.
- Some key features:
  - Importing Java dumps to your local workspace for analysis.
  - Display of Java threads in the dump and stack trace of executing program in each thread.
  - Display of general information about Java dumps (command to start the Java program, environmental values, and system information).
  - Source line display.
  - Discovery of Java run-time environments installed on your system.
  - Discovery of Java dumps on your system.
  - Discovery of Java processes running on your system and instructing to take a dump.

Fault Analyzer for Multi-platforms Overview



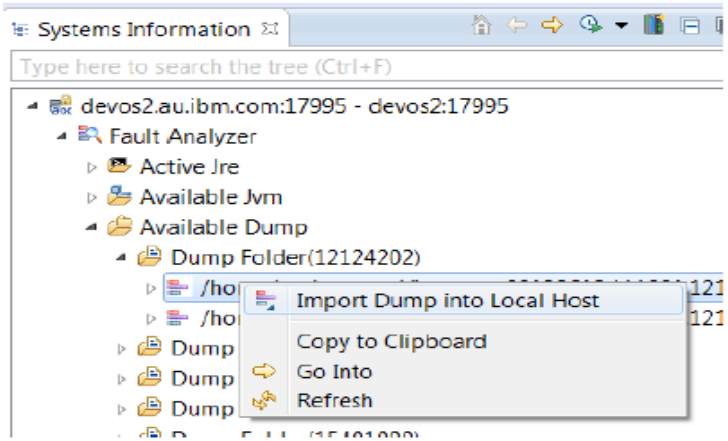
- Discovers running Java processes on your machine:
  - Instruct to take a dump.
  - Browse JAR files associated with the process and let you browse classes.
- Discover JVMs available on your system.
- Discover Java dumps on your machine.
  - Allows you to import Java dumps to your desktop for analysis.

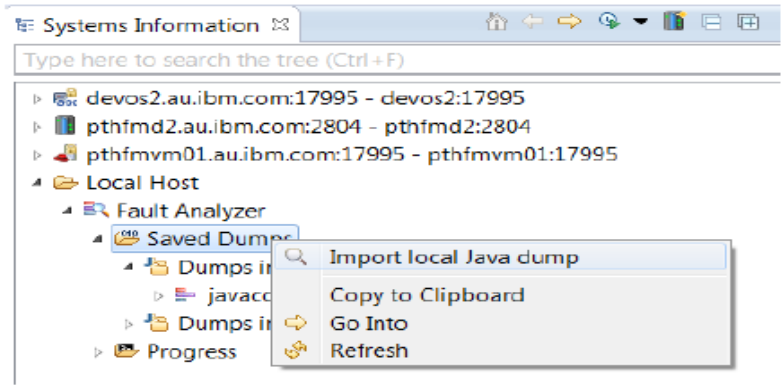


Fault Analyzer for Multi-platforms Overview

Java dump analysis overview

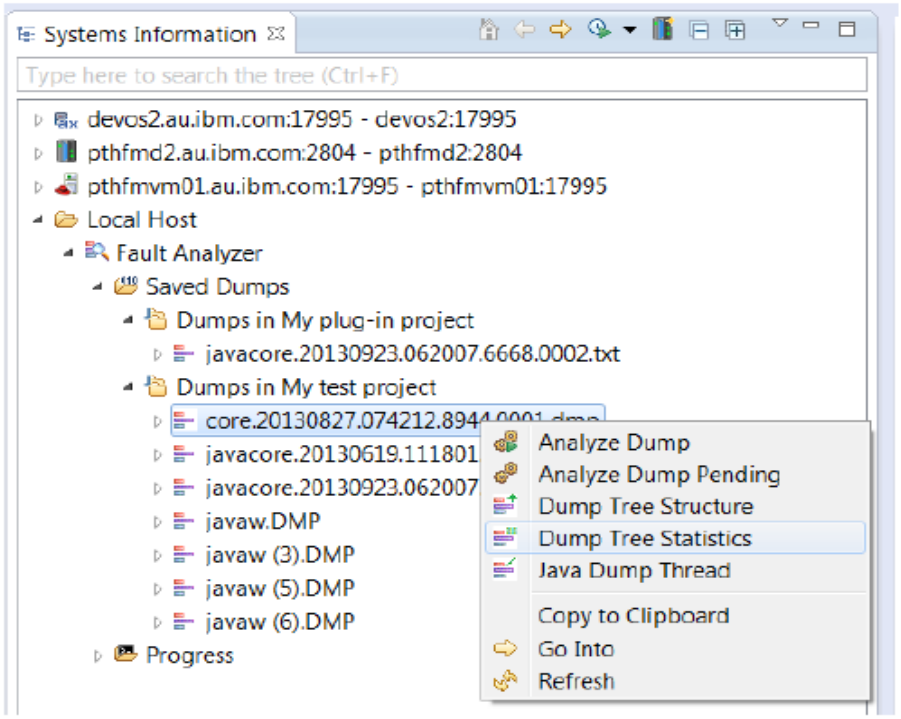
- Java projects in Eclipse knows everything about your Java applications:
  - Source files.
  - JAR files for dependent libraries.
  - Dependencies on other Java projects.
- We place Java dumps into your local Java project and utilize the information available during the analysis process (eg. Source display).
- The plug-in provides method for importing dumps from a remote system (AIX and Red Hat) to your Java project.
- The plug-in provides method for importing a local dump into your Java project.
- Once imported, you can kick off the analysis of Java dumps.





Fault Analyzer for Multi-platforms Overview

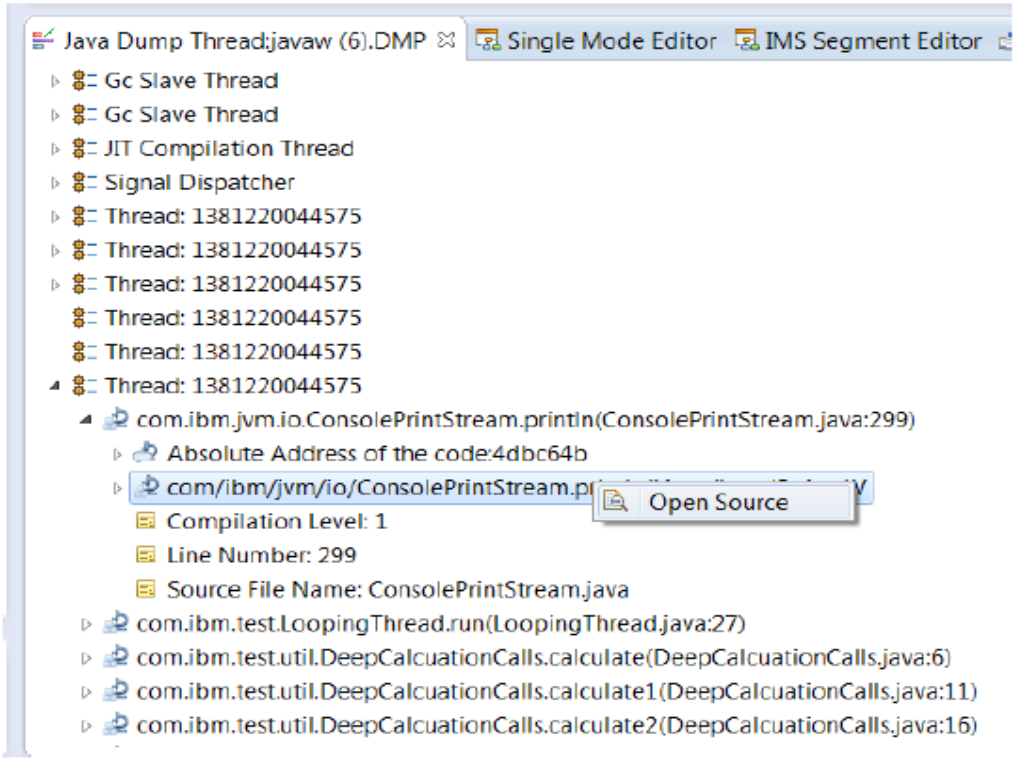
Java dump analysis overview - continue



- From a list of imported Java dumps, you can kick off various analysis function.
- **Java Dump Thread**: looking at currently active threads and the stack trace of threads at the time when the dump was taken.
- **Dump Tree Structure**: a graphical navigator for looking around everything you can discover in the dump (eg. Environmental variables).
- **Analyze Dump:** starts analysis of Java dump and allows you invoke various DTFJ commands.

Fault Analyzer for Multi-platforms Overview

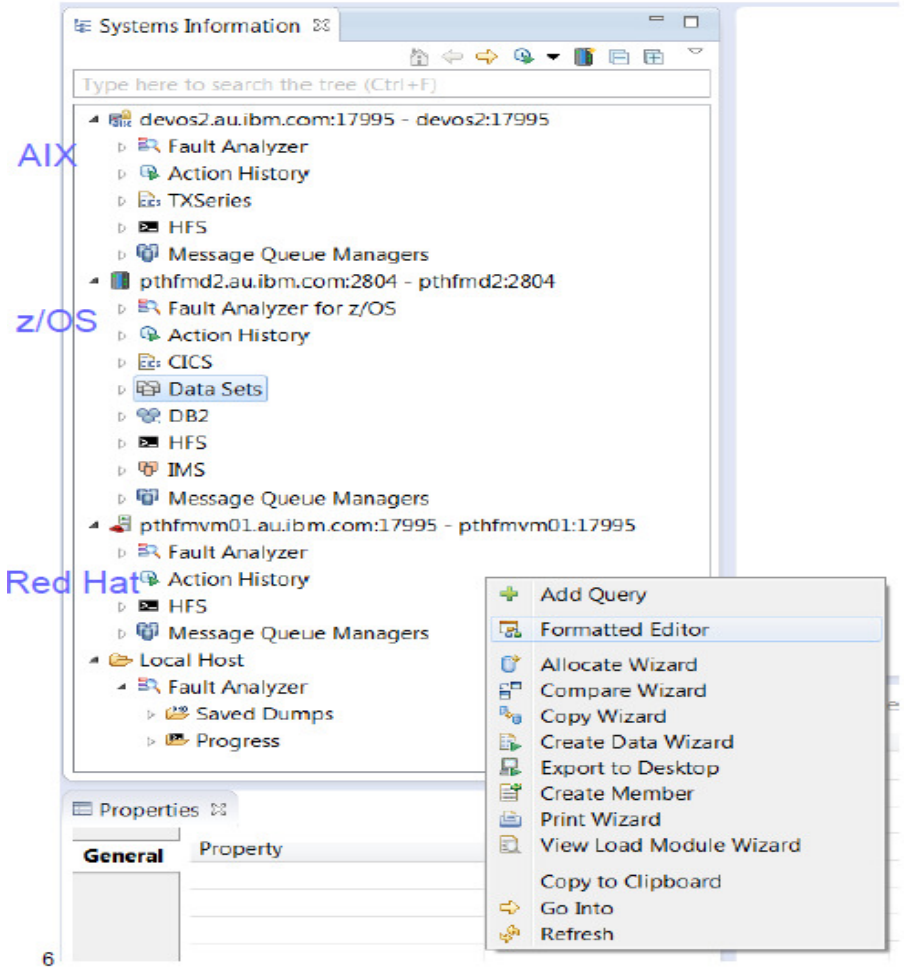
Dump Thread Analysis



- You can see all threads which were running at the time when a dump was taken.
- You can expand each thread and examine the current execution stack. You will see the source line number (if the information is available in the dump).
- You can select *Open Source* to look at the source information.

File Manager for Multi-platforms Overview

File Manager for Multiplatforms overview



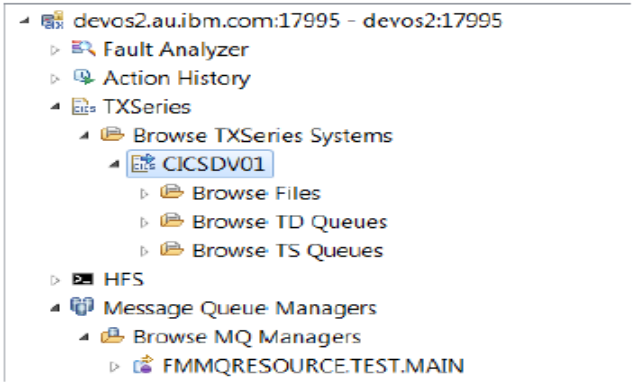
- Supports AIX and Red Hat operating systems on System p.
- Editing of hierarchical file system files, WebSphere MQ queues, and TXSeries resources (on AIX platform only).
- Formatted editing enabled using dynamic templates.
- Use the same navigation and look & feel of FM plug-in functions from z/OS platform.

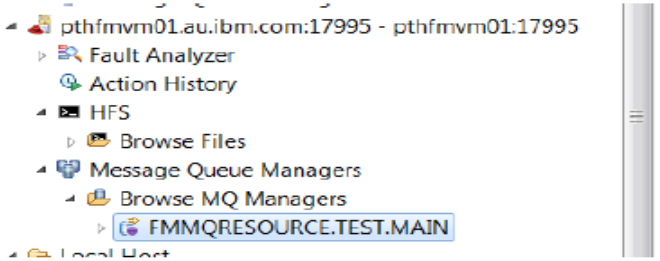
Familiar FM functionality provided for distributed platforms

File Manager for Multi-platforms Overview

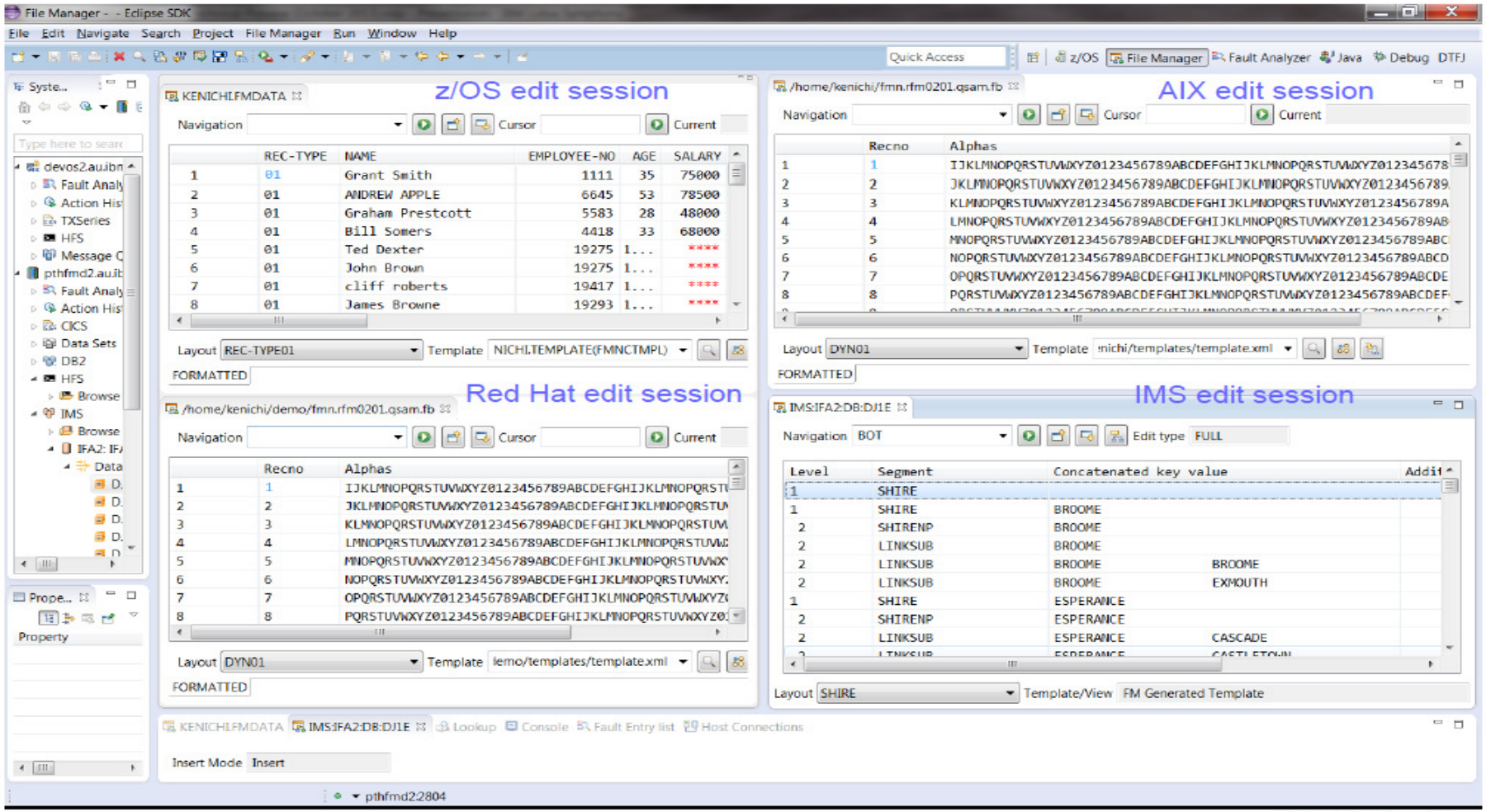
What features are available?

- Only dynamic templates are supported on distributed platforms. No criteria expression support (layouts are mapped to record based on record length + first come first serve basis).
- Editing in character mode.
- Editing in formatted mode using dynamic templates.
- Works with Hierarchical File System files, WebSphere MQ queues and TXSeries resources (CICS files, TD queues and TS queues).





File Manager for Multi-platforms Overview



New Debug Tool Code Coverage

- Not a replacement for integrated Code Coverage in RD/z
- Designed to provide quick Code Coverage statistics
- Designed to be use in conjunction with debugging sessions
- Allows flexibility in the reporting of Code Coverage results
- XML is use as mechanism to provide program selection, selection criteria, and final report
  - You can create your own facility to create XML for input into Code Coverage facility or to read results and present according to your needs.
- Help you isolate specifics statements of interest for the developer
- A complement for Unit Test verification but can also be use for integration testing and other types of testing scenarios
- Only support compilers that provides the SEPARATE compile option to create SYSDEBUG files.
- Not yet supported for C, C++, COBOL V5, and Assembler
- Can be run in batch
- It support CICS applications as well.

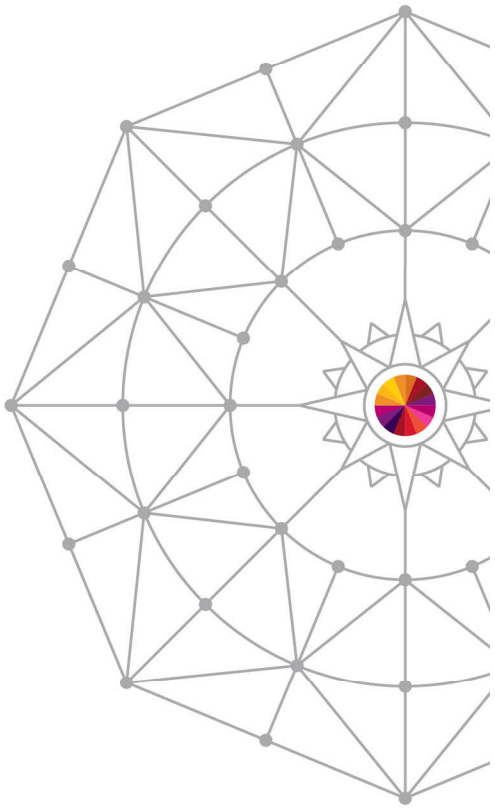
Code Coverage: Setup

- A TEST(ALL,*,PROMPT,MFI:*) runtime parm is used to tell Debug Tool to run. An LE Environment Variable ENVAR("EQA_STARTUP_KEY=CC") is used to tell Debug Tool to collect code coverage via the MFI debugger engine (vs RD/z for example, which writes its information out a different way).

```
//*
//EQAOPTS   DD *
          EQAXOPT  CCOUTPUTDSN,'&&USERID.DBGTOOL.CCOUTPUT'
          EQAXOPT  CCOUTPUTDSNALLOC,'MGMTCLAS(STANDARD)                  +
                STORCLAS(DEFAULT) LRECL(255) BLKSIZE(0) RECFM(V,B)       +
                DSORG(PS) SPACE(2,2) CYL'
          EQAXOPT  CCPROGSELECTDSN,'&&USERID.DBGTOOL.CCPRGSEL'
          EQAXOPT  END
//CEEOPTS   DD *
TEST(ALL,*,PROMPT,MFI:*),ENVAR("EQA_STARTUP_KEY=CC")
/*
```

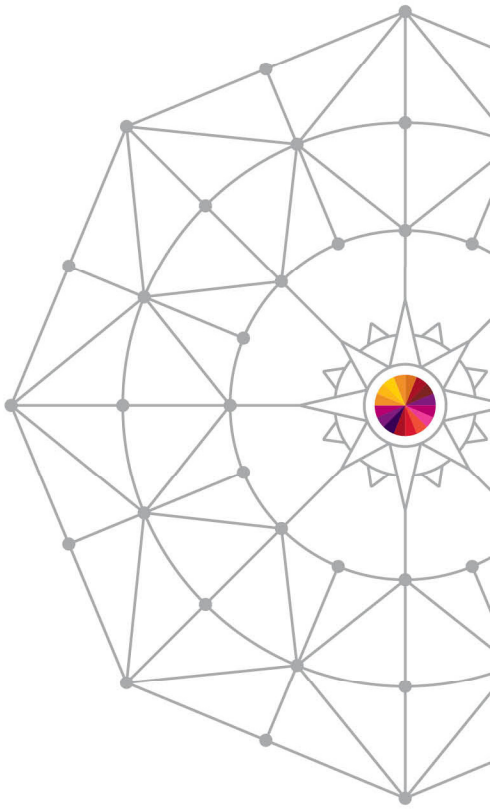
Code Coverage: Setup Debug Tool Utilities (ISPF)

```
------------------------- Debug Tool Code Coverage -------------------------
Option ===> 1


1  Observation viewer
   Browse code coverage observations.

2  Debug Tool options
   Create or modify the Debug Tool code coverage options.

3  Observation selection criteria
   Create or modify the observation selection criteria and source markers.

4  Observation extraction
   Extract code coverage observations using selection criteria.

5  Report generation
   Create report.
```



Code Coverage: Viewer

```
---------------- Debug Tool - Code Coverage Observation Viewer Row 1 to 4 of 7
Command ===>                                                  Scroll ===> PAGE

  Enter / to sort the table entries.

Enter (V)iew table entry command to view source listing.

b Run Date :  2013/10/02      Run Time:   17:08:44
  Group ID 1: COST            Group ID 2: BENEFIT         TSO ID:   GMYOUN2
  Load Name:  COBO1           Prog Name:  COBO1A
  Comp Date:  2013/10/01      Comp Time:  12:32:00   Debug override: N
  Tot Stmts:  17              Exec Stmts: 15              Percent:   88.23%
-------------------------------------------------------------------------
  Run Date :  2013/10/02      Run Time:   17:08:44
  Group ID 1: COST            Group ID 2: BENEFIT         TSO ID:   GMYOUN2
  Load Name:  COBO1           Prog Name:  COBO1B
  Comp Date:  2013/10/01      Comp Time:  12:32:00   Debug override: N
  Tot Stmts:  10              Exec Stmts: 9               Percent:   90.00%
-------------------------------------------------------------------------
  Run Date :  2013/10/02      Run Time:   17:08:44
  Group ID 1: COST            Group ID 2: BENEFIT         TSO ID:   GMYOUN2
  Load Name:  COBO1           Prog Name:  COBO1C
  Comp Date:  2013/10/01      Comp Time:  12:32:00   Debug override: N
  Tot Stmts:  14              Exec Stmts: 12              Percent:   85.71%
-------------------------------------------------------------------------
  Run Date :  2013/10/02      Run Time:   17:08:50
  Group ID 1: COST            Group ID 2: BENEFIT         TSO ID:   GMYOUN2
  Load Name:  PLIO1           Prog Name:  PLIO1A
  Comp Date:  2013/10/02      Comp Time:  17:08:14   Debug override: N
  Tot Stmts:  12              Exec Stmts: 9               Percent:   75.00%
```

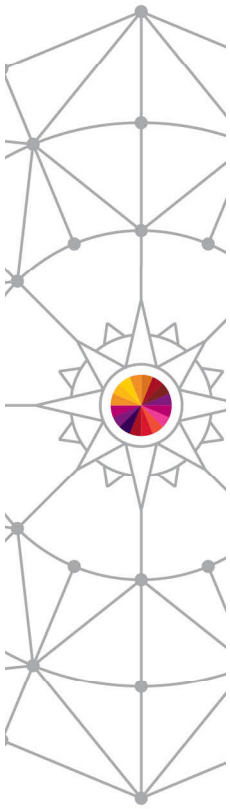


Shows all observations in repository

Code Coverage: Browsing observations using the viewer

You will then see something like this. The same statistical information as show above is listed, and then an annotated listing is shown (extracted from the SYSDEBUG file and then annotated with coverage information).

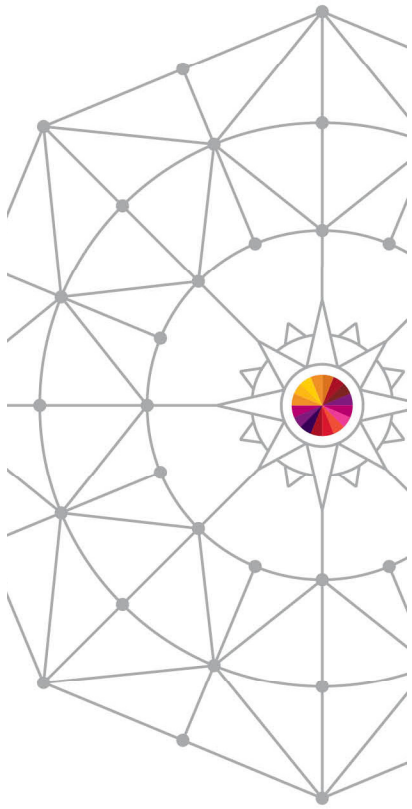
```
BROWSE    GMYOUN2.DTCCTEMP.D13276.T075437.EQARC621   Line 00000000 Col 001 080
Command ===>                                            Scroll ===> PAGE
******************************** Top of Data ********************************
Rpt Date :  2013/10/03      Rpt Time:   07:54:37

Run Date :  2013/10/02      Run Time:   17:08:44
Group ID 1: COST            Group ID 2: BENEFIT         TSO ID:   GMYOUN2
Load Name:  COB01           Prog Name:  COB01A
Comp Date:  2013/10/01      Comp Time:  12:32:00   Debug override: N
Tot Stmts:  17              Exec Stmts: 15             Percent:   88.23%

Rollup History:

Observation is not part of rollup

             ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----
     1               * COB01A - COBOL EXAMPLE FOR DTCU
     2
     3                IDENTIFICATION DIVISION.
     4                PROGRAM-ID. COB01A.
     5               *************************************************************
     6               * Licensed Materials - Property of IBM
     7               *
     8               * 5655-M18: Debug Tool for z/OS
     9               * 5655-M19: Debug Tool Utilities and Advanced Functions fo
    10               * (C) Copyright IBM Corp. 1997, 2004 All Rights Reserved
    11               *
    12               * US Government Users Restricted Rights - Use, duplication
```



Code Coverage: Browsing observations using the viewer….

- Scroll down to see the annotation. > means executed, < means not executed.

```
BROWSE    GMYOUN2.DTCCTEMP.D13276.T075437.EQARO821   Line 00000062 Col 001 080
Command ===>                                            Scroll ===> CSR
    49                * ACCESS BY LOW LEVEL QUALIFIERS
    50  >                 MOVE 'KY' TO STATE
    51  >                 MOVE 'LEX' TO CITY
    52  >                 MOVE 'VM ' TO OP-SYS
    53                    .
    54
    55                 PROCA.
    56  >                 PERFORM  LOOP1 UNTIL IAPARM1 = 0
    57
    58  >                 IF IAPARM2 = 0   THEN
    59                *   PROCA NOT EXECUTED
    60  <                     PERFORM PROCA.
    61
    62
    63  >                 PERFORM LOOP2 UNTIL IAPARM2 = 0
    64                    .
    65  >                 STOP RUN
    66                    .
    67
    68                 PROCA.
    69                *   PROCA NOT EXECUTED
    70  <                 MOVE 10 TO P1PARM1
    71                    .
    72                 LOOP1.
    73  >                   IF IAPARM1 > 0 THEN
    74  >                     SUBTRACT 1 FROM IAPARM1.
```



Code Coverage: Observation Criteria

- We are going to use this to generate a report from the coverage collected above. Our report will only show information that has the GROUP ID1 and 2 COST BENEFIT, was run by your TSO ID, and has one of the COB01 COBOL testcases.

- In addition, various source markers can be specified, which are then marked (if found) in the the report generated in a subsequent step and (and which also have statistics calculated for the lines that are marked). You may have to scroll down to see the marker section.

```
------------ Debug Tool - Edit Code Coverage Selection Criteria ------------
Command ===>
                                                                More:     +
Specify code coverage observation selection criteria

Enter attribute value and comparison operator. Comparison operators
are (E)qual, (G)reater, (L)ess, (GE) greater than or equal,
(LE) less than or equal, and (NE) not equal.

Attribute name              Value              Operator              Rollup
Run date (YYYY/MM/DD)                             (E,G,L,GE,LE,NE)
Run time (HH:MM:SS)                               (E,G,L,GE,LE,NE)
Group ID 1                  COST               E  (E,NE)             N (Y/N)
Group ID 2                  BENEFIT            E  (E,NE)             N (Y/N)
TSO ID                      GMYOUN2            E  (E,NE)             N (Y/N)
Load module name                                  (E,NE)
Program name                COB01*             E  (E,NE)
Compile date (YYYY/MM/DD)                         (E,G,L,GE,LE,NE)
Compile time (HH:MM:SS)                           (E,G,L,GE,LE,NE)
Debug override                                    (E,NE)                (Y/N)
Total statements                                  (E,G,L,GE,LE,NE)
Executed statements                               (E,G,L,GE,LE,NE)

Specify source markers for code coverage percentage analysis

Marker type: SINGLE/SECTIONBEGIN/SECTIONEND
Selection:   INCLUDE/EXCLUDE

Marker type   Selection  Column Column String
Selection:   INCLUDE/EXCLUDE

Marker type   Selection  Column Column String
                         Start  End
SINGLE        INCLUDE    73     75     PMR
SINGLE        EXCLUDE    73     80     PMR11114
SECTIONBEGIN  INCLUDE    7      80     DEFECT123BEGIN
SECTIONEND    INCLUDE    7      80     DEFECT123END
```

Code Coverage: Observation Extraction

- Now specify the names of the output data set from the debugger, the selection criteria data set, and a new data set to hold the extracted output, and press enter.

```
------------- Debug Tool - Code Coverage Observation Extraction ------------
Command ===>


The observation selection function extracts observations that meet the
selection criteria from the observation data set. It writes the result
to the observation output data set.

Specify the name of a code coverage observation data set.

  Data Set Name . . . 'GMYOUN2.DBGTOOL.CCOUTPUT'

Specify the name of a code coverage selection criteria data set.

  Data Set Name . . . 'GMYOUN2.DBGTOOL.CCOBSSEL.COB01'

Specify the name of a code coverage observation output data set.

  Data Set Name . . . 'gmyoun2.dbgtool.ccoutput.extract'

Press Enter to continue.
Press Exit or Cancel to exit.
```

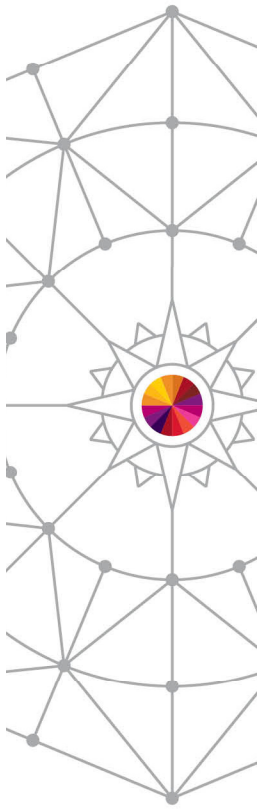
You will see a message at the top of the screen indicating that the extraction was done.

```
-------------- Debug Tool - Code Coverage Observation
Extract observations OK
```

Code Coverage: Report Generation

- Specify option 3, and the extracted observation data set from above, the selection criteria data set from above and a new report data set.

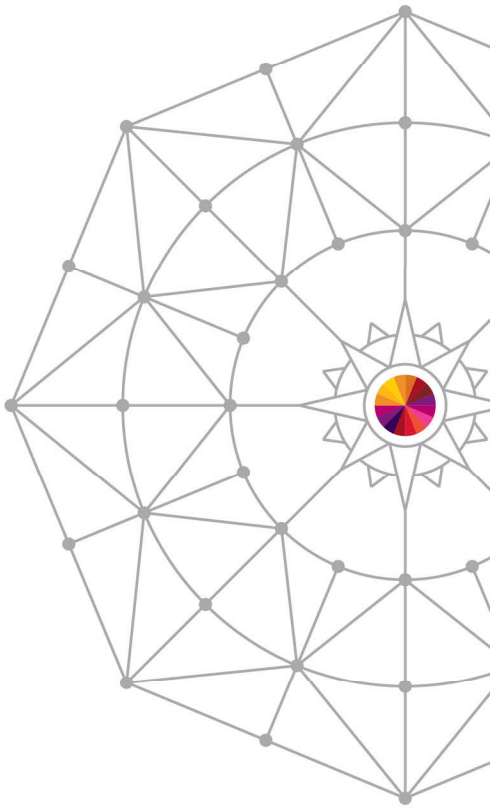
```
---------------- Debug Tool - Code Coverage Report Generation ----------------
Command ===>


The report generator adds marked source statements and code coverage
statistics to the extracted observations. It writes the result
to the report output data set along with the selection criteria.

Select a report action.

3  1. Create report in XML format
   2. Create report in Presentation format
   3. Create and browse report in Presentation format

Specify the name of a code coverage extracted observation data set.

  Data Set Name . . . 'GMYOUN2.DBGTOOL.CCOUTPUT.EXTRACT'

Specify the name of a code coverage selection criteria data set.

  Data Set Name . . . 'GMYOUN2.DBGTOOL.CCOBSSEL.COB01'

Specify the name of a code coverage report data set.

  Data Set Name . . . 'gmyoun2.dbgtool.ccoutput.report'

Press Enter to continue.
Press Exit or Cancel to exit.
```



Press enter to see the report.

Code Coverage Report

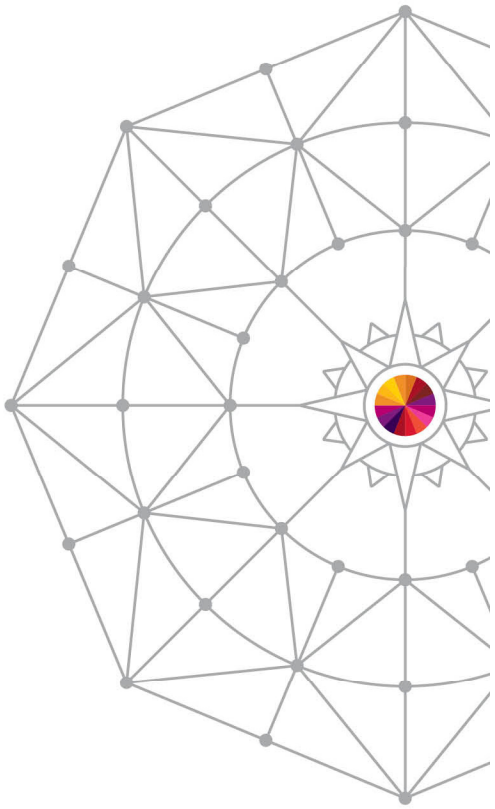
```
 BROWSE    GMYOUN2.DBGTOOL.CCOUTPUT.REPORT            Line 00000000 Col 001 080
 Command ===>                                                 Scroll ===> CSR
******************************** Top of Data ***********************************
Rpt Date :  2013/10/03      Rpt Time:   12:20:33

Run Date :  2013/10/02      Run Time:   17:08:44
Group ID 1: COST            Group ID 2: BENEFIT         User ID:   GMYOUN2
Load Name:  COB01           Prog Name:  COB01A
Comp Date:  2013/10/01      Comp Time:  12:32:00   Debug override: N
Tot Stmts:  17              Exec Stmts: 15              Percent:   88.23%

Rollup History:

Observation is not part of rollup

              ----+--*A-1-B--+----2----+----3----+----4----+----5----+----6----+
     1              * COB01A - COBOL EXAMPLE FOR DTCU
     2
     3               IDENTIFICATION DIVISION.
     4               PROGRAM-ID. COB01A.
     5              ***************************************************************
     6              * Licensed Materials - Property of IBM
     7              *
     8              * 5655-M18: Debug Tool for z/OS
     9              * 5655-M19: Debug Tool Utilities and Advanced Functions for
    10              * (C) Copyright IBM Corp. 1997, 2004 All Rights Reserved
    11              *
    12              * US Government Users Restricted Rights - Use, duplication
```



Code Coverage Report .....

Scroll down to see the annotation and markers (I Include, E Exclude, B Both):

```
BROWSE    GMYOUN2.DEGTOOL.CCOUTPUT.REPORT          Line 00000052 Col 001 080
Command ===>                                              Scroll ===> CSR
   39               * Defect456Begin
   40
   41                PROG.
   42               * ACCESS BY TOP LEVEL QUALIFIER
   43 I>                 MOVE 'ILCHIMVS' TO TASTRUCT
   44
   45               * ACCESS BY MID LEVEL QUALIFIERS
   46 I>                 MOVE 'ILSPR' TO LOC-ID
   47 I>                 MOVE 'AIX' TO OP-SYS
   48
   49               * ACCESS BY LOW LEVEL QUALIFIERS
   50 B>                 MOVE 'KY' TO STATE
   51 B>                 MOVE 'LEX' TO CITY
   52 B>                 MOVE 'VM ' TO OP-SYS
   53                    .
   54
   55                PROGA.
   56  >                 PERFORM  LOOP1 UNTIL TAPARM1 = 0
   57
   58  >                 IF TAPARM2 = 0    THEN
   59               *   PROCA NOT EXECUTED
   60 I<                     PERFORM PROCA.
   61
   62
   63 I>                 PERFORM LOOP2 UNTIL TAPARM2 = 0
   64                    .
```



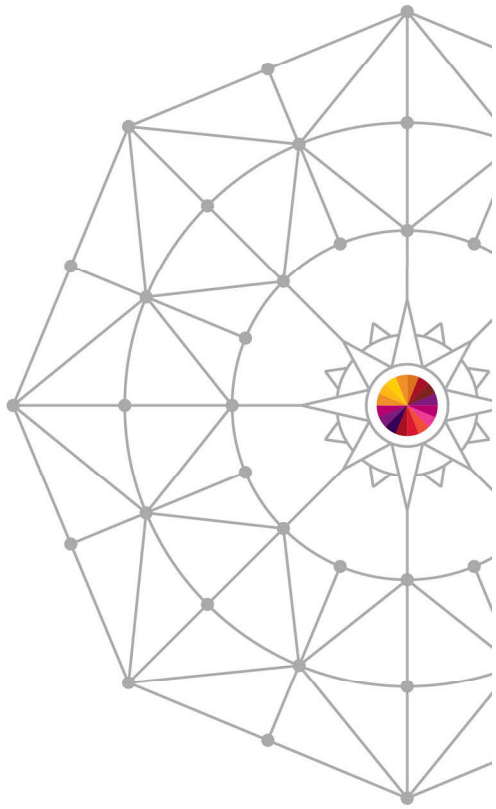
Code Coverage Report …..

- Scroll to the right (RIGHT 30) to see the source markers

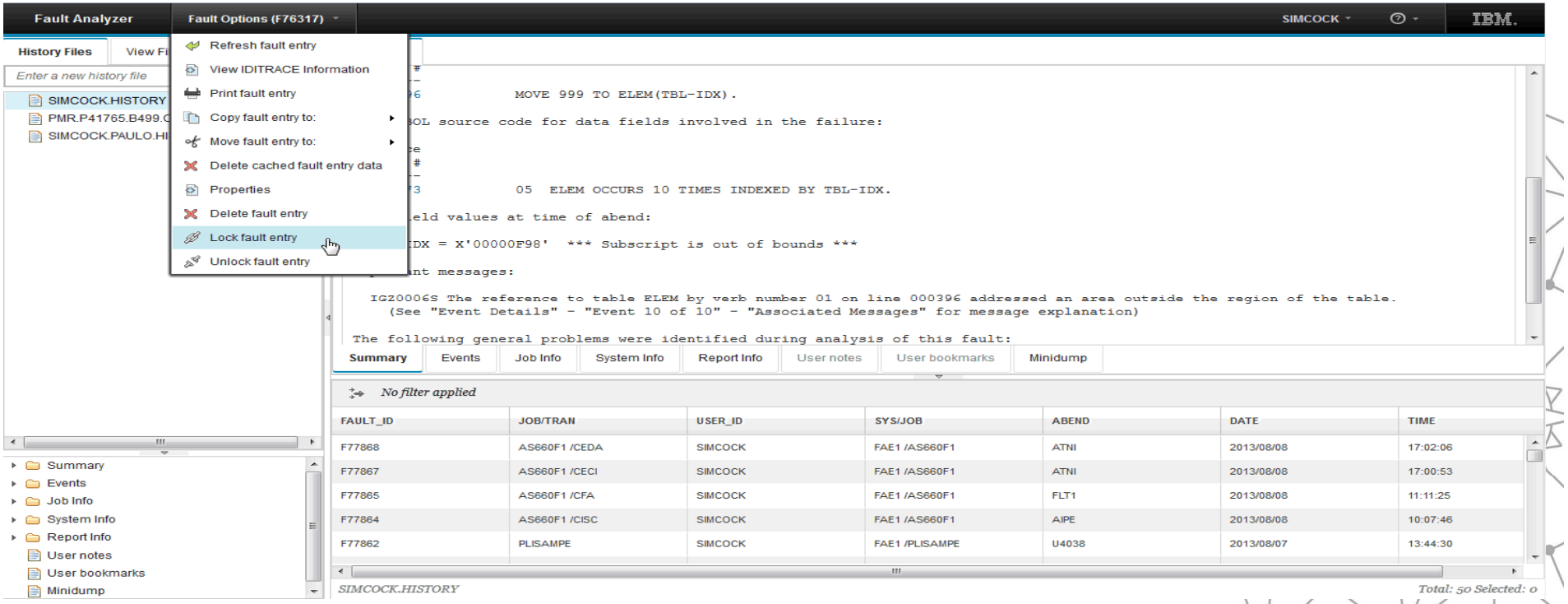
```
 BROWSE    GMYOUN2.DBGTOOL.CCOUTPUT.REPORT          Line 00000052 Col 031 110
 Command ===>                                             Scroll ===> CSR
56Begin


                                                        PMR11112
BY TOP LEVEL QUALIFIER                                  PMR11112
 'ILCHIMVS' TO TASTRUCT                                 PMR11112

BY MID LEVEL QUALIFIERS                                 PMR11113
 'ILSPR' TO LOC-ID                                      PMR11113
 'AIX' TO OP-SYS                                        PMR11113

BY LOW LEVEL QUALIFIERS                                 PMR11114
 'KY' TO STATE                                          PMR11114
 'LEX' TO CITY                                          PMR11114
 'VM ' TO OP-SYS                                        PMR11114
                                                        PMR11114
```

scroll left and down some more to see the source marker info and some statistics based on them.

```
BROWSE    GMYOUN2.DBGTOOL.CCOUTPUT.REPORT          Line 00000096 Col 001 060
Command ===>                                             Scroll ===> CSR
                          Start  End
Marker type   Selection   Column Column String
------------ ---------   ------ ------ ------------------------------
SINGLE       INCLUDE         73     75 PMR
SINGLE       EXCLUDE         73     80 PMR11114
SECTIONBEGIN INCLUDE          7     60 DEFECT123BEGIN
SECTIONEND   INCLUDE          7     80 DEFECT123END

          Statements  Executed  Percentage
          ----------  --------  ----------
Total             17        15       88.23
Included           8         6       75.00
Excluded           0         0        0.00
Incl/Excl          3         3      100.00
```



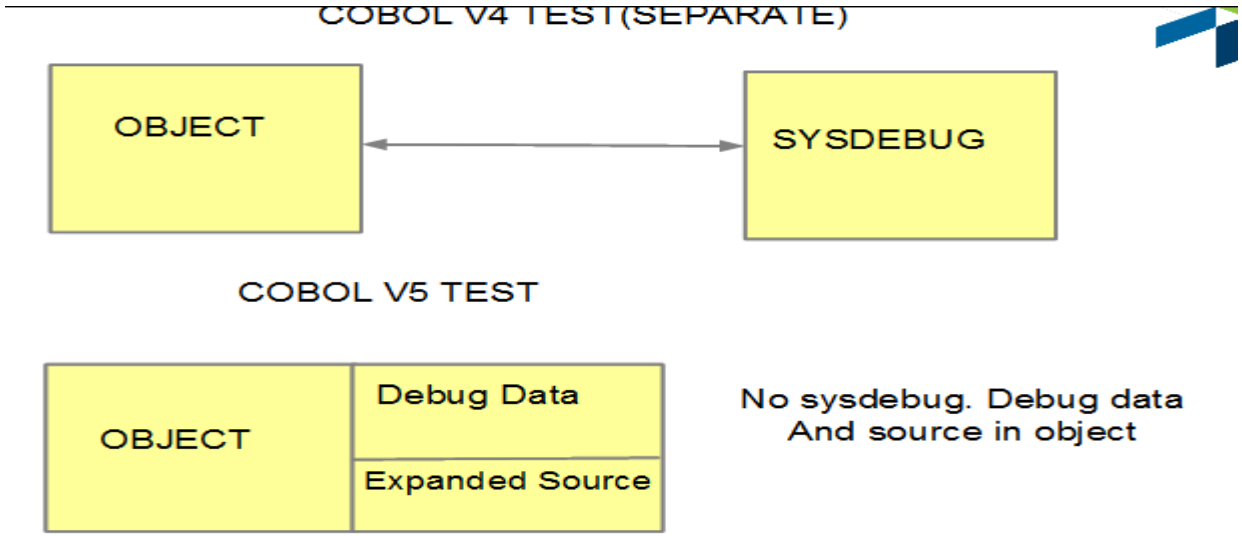
Fault Analyzer new WEB interface



- Modern Dojo-based Web 2.0 interface using AJAX for seamless data loading and presentation
- IBM One UI theme (IDX)
- Enhanced functionality, at parity with Eclipse-based plug-in interface, including
  - Fault entry refresh capability
  - Fault Entry management (MOVE/COPY/DELETE)
  - Sidefile (source) display
  - Ability to bookmark/annotate minidump
  - Print capability
- Simplified installation and management requirements
- Very fast and responsive

Support for COBOL V5

- All PD Tools support the new COBOL V5 compiler
- Major change in COBOL compiler architecture in the last 3 decades.
- Debug information embedded in the load module. Not need for a SYSDEBUG file



Debug Tool Enhancements for COBOL V5

DESCRIBE ATTRIBUTES, MOVE and IF commands

The PIC string shown in Debug Tool appears as it is specified in the source and not normalized as it was prior to Enterprise COBOL Version 5.

Level members are shown as written in the source code and not normalized as they were prior to Enterprise COBOL Version 5.

There are clearer data descriptions. For example, you could now see:

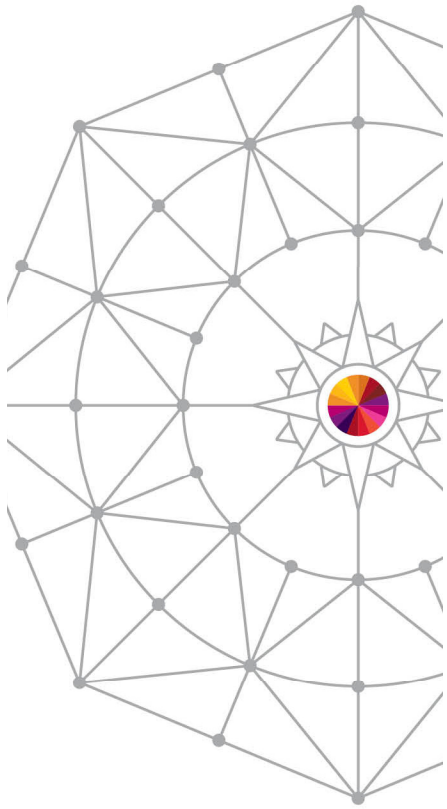
S9(5) SIGN LEAD SEP DISP

instead of

S9(5) DSLS

The MOVE and COMPUTE commands in Debug Tool have expanded to allow the same data types as the compiler for receivers and senders. This enhancement removes previous restrictions on the use of those commands.

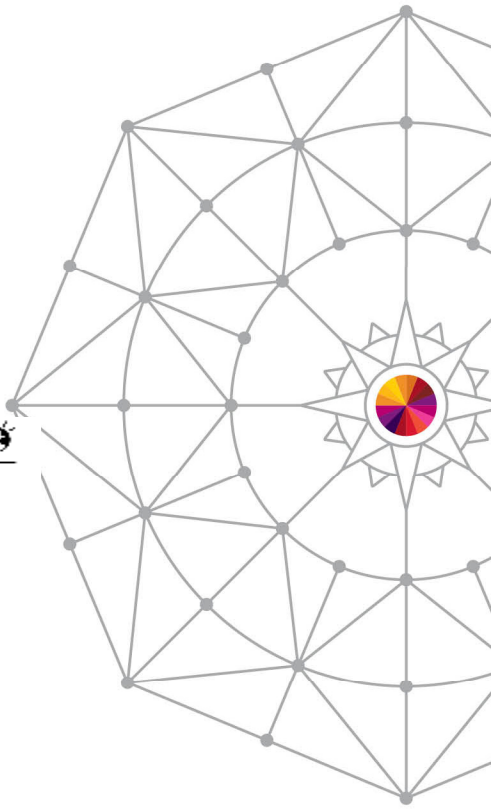
The IF command has been expanded. Allowable comparisons for relational conditions are expanded in Debug Tool with Enterprise COBOL V5. The allowable comparison for relational conditions (involving data items, literals, and figurative constants) are implemented according to the Enterprise COBOL Language Reference.



Debug Tool Enhancements for COBOL V5

STEP command

You can STEP and set breakpoints for the WHEN phrase of EVALUATE.
STEP OVER with PERFORM is now supported.
Support for COBOL types
Debug Tool now supports the correct maximum value in all binary data types.
For example, an 8-byte, unsigned COMP-5 data item can contain a maximum value of 18,446,744,073,709,551,615, which is 20 digits.
Better debugging support at all OPT levels



Load Module Analyzer : AMODE/RMODE

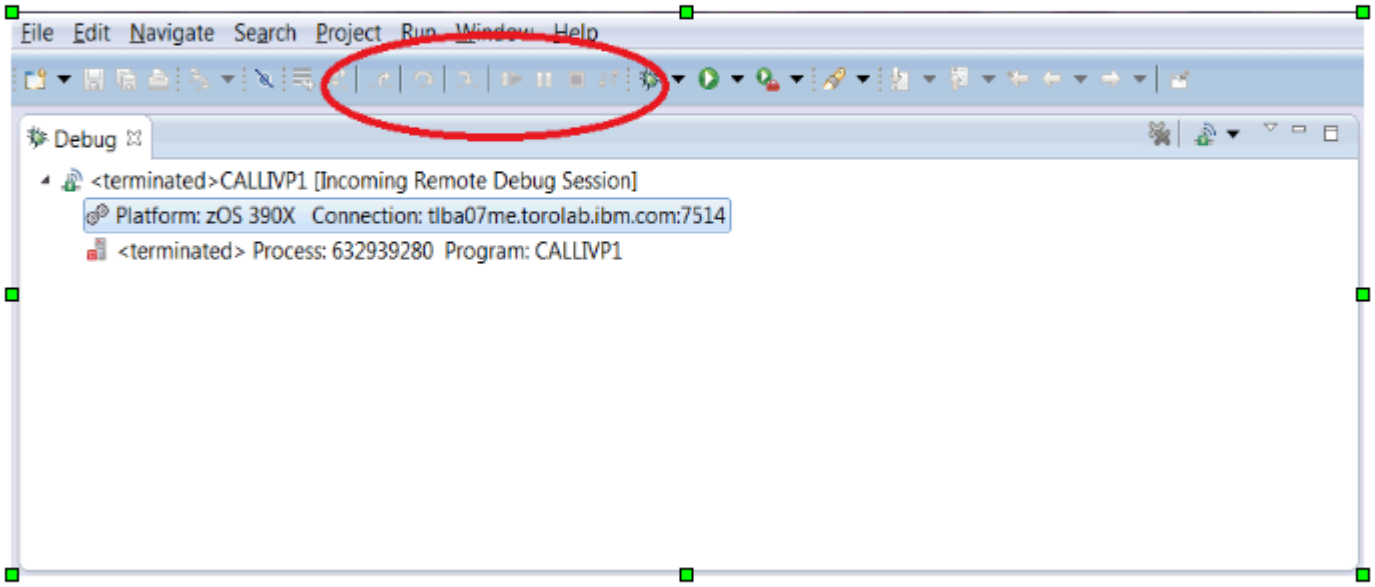
```
1 5655-Q10   Debug Tool Version 13 Release 1.2  Load Module Analyzer   2
          Load Module  TSFLNT2.TEST.LOAD(BKMEB11)   AMODE(31),RMODE(ANY)
```

Load Module Analyzer : COBOL V5

```
 Trn-Date    Program-Description
 2013/10/21  Enterprise COBOL for z/OS Version 5
```

Playback for Remote debugging

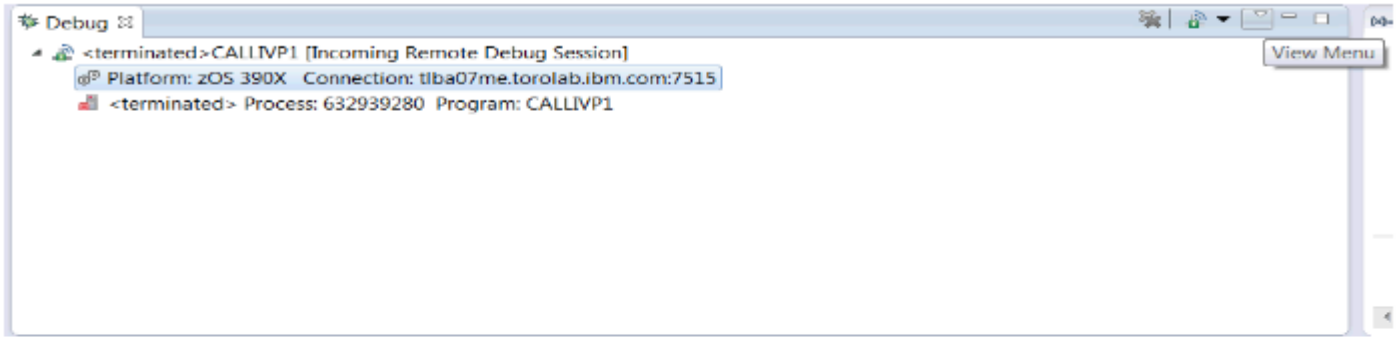
- Works with RD/z 9.0, PD Tools Studio 12.1.1.2 , CICS Explorer 5.1.1.1
- Only NODATA is supported (variables are not restored)



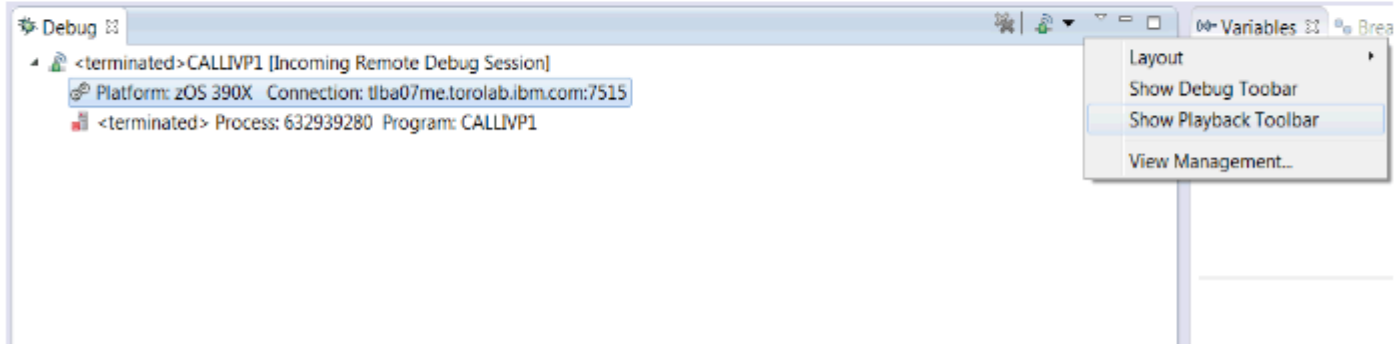
You may notice the Debug buttons (resume, step, etc...) are no longer visible in the Debug pane. They have been moved into the main task bar. This is eclipse 4.2 behavior

Playback for Remote debugging

First, we need to show the playback toolbar. Open the menu in the Debug pane.

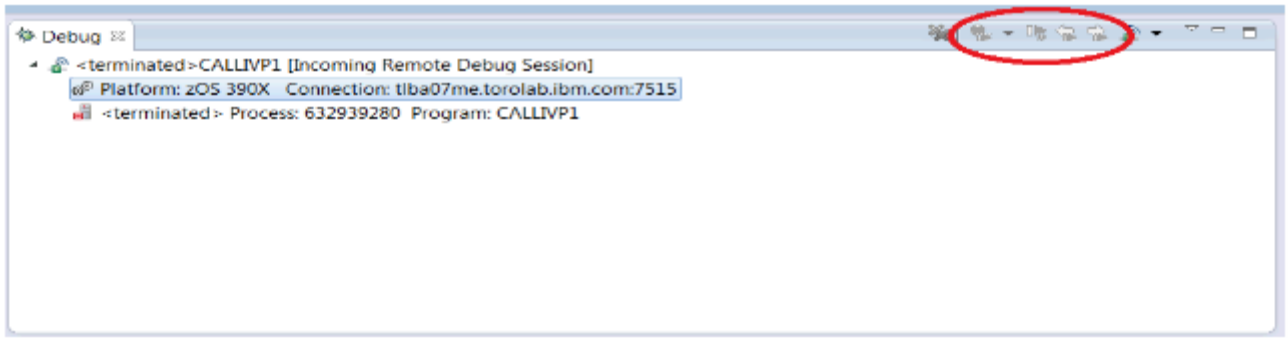


And select 'Show Playback Toolbar'. (You can also select 'Show Debug Toolbar' if you want to add resume/step/terminate/etc.)

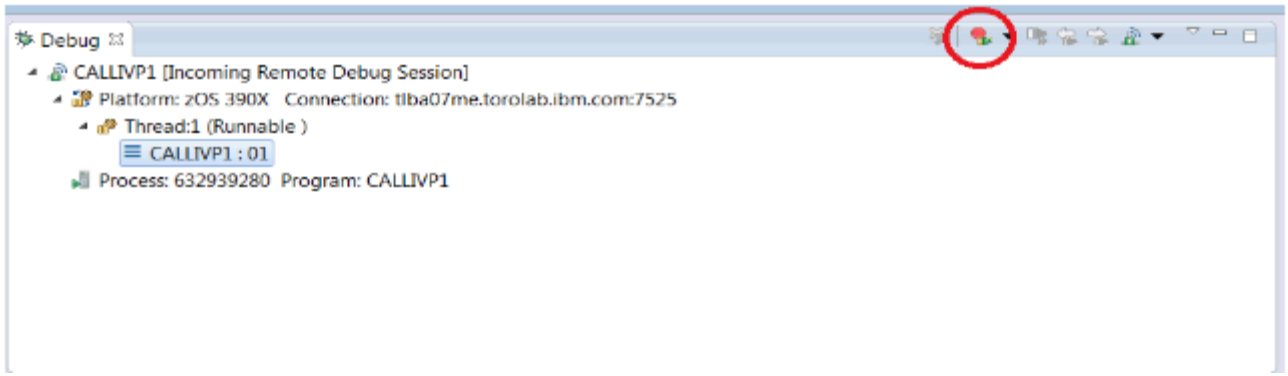


Playback for Remote debugging

There should now be four new buttons visible. 'Start Playback Recording', 'Resume Debugging', 'Move Back', and 'Move Forward'.

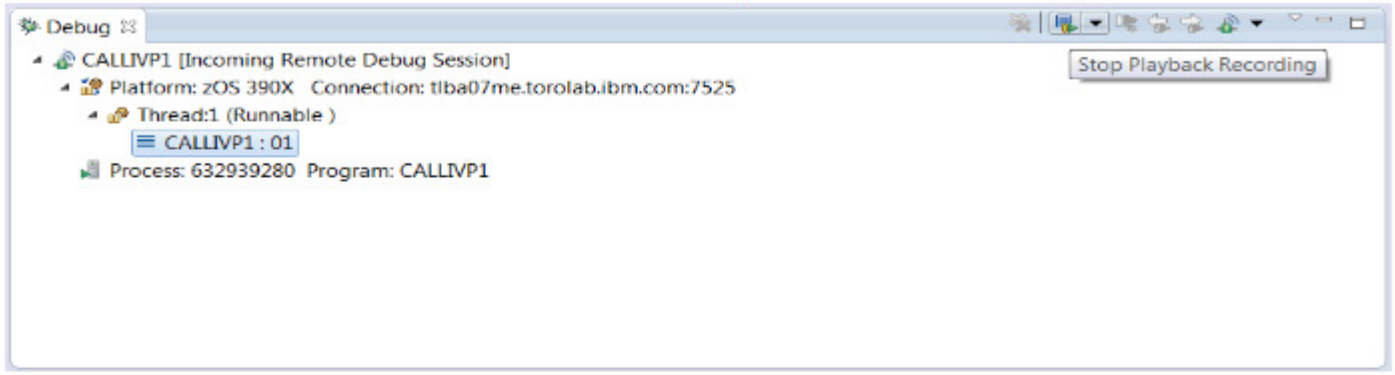


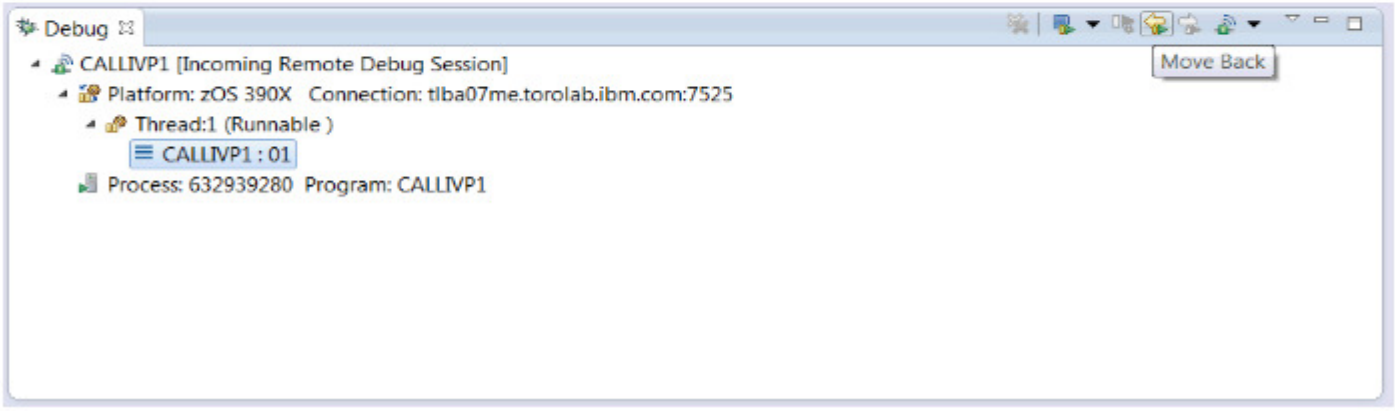
Once you start a debug session, the 'Start Playback Recording' will become available.



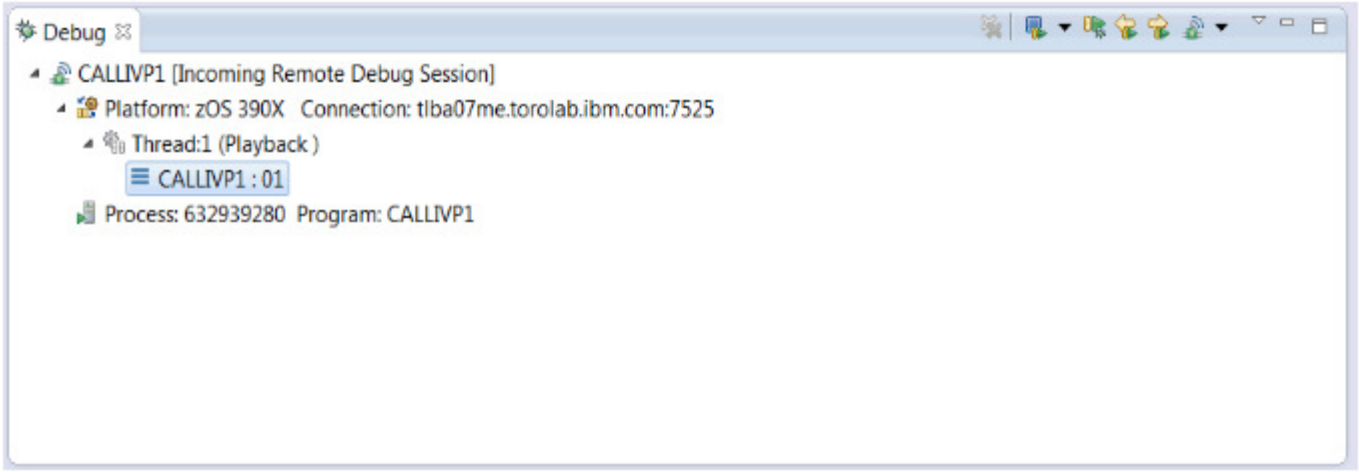
Playback for Remote debugging

Click 'Start Playback Recording' - this is equivalent to executing a 'PLAYBACK ENABLE' command in MFI. The icon will change to indicate recording has started, and the button will change to 'Stop Playback Recording' (equivalent to 'PLAYBACK DISABLE').





Playback for Remote debugging



We are now in playback mode. The thread state has changed from 'Runnable' to 'Playback'. Step and resume buttons are disabled. Any monitored expressions are 'frozen' from immediately before we entered playback and cannot be modified. The registers view will update as we move backwards and forwards in playback mode. The 'Resume Debugging' and 'Move Forward' buttons are also now available.

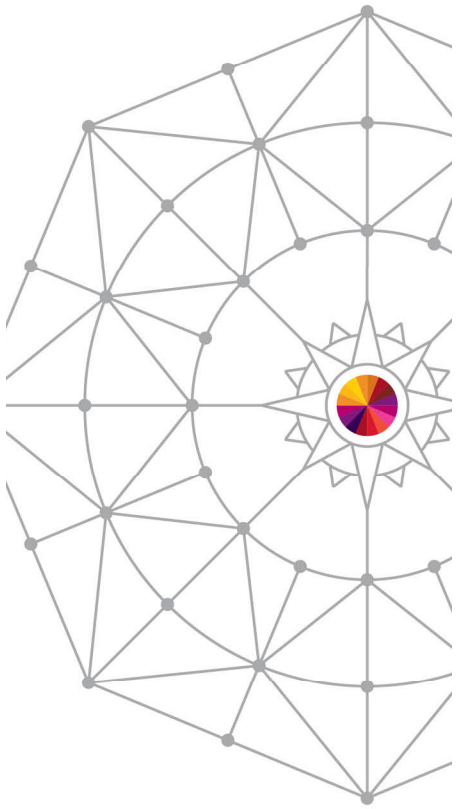
'Resume Debugging' is equivalent to the 'PLAYBACK STOP" command in MFI. We will return to the current point of execution and leave playback mode.

There are no 'PLAYBACK BACKWARD' or 'PLAYBACK FORWARD' command equivalents because there is no playback direction in remote. Forward and backward stepping have discrete buttons.

There is no 'PLAYBACK START" command equivalent. Playback mode is entered automatically when the 'Move Back' button is used.

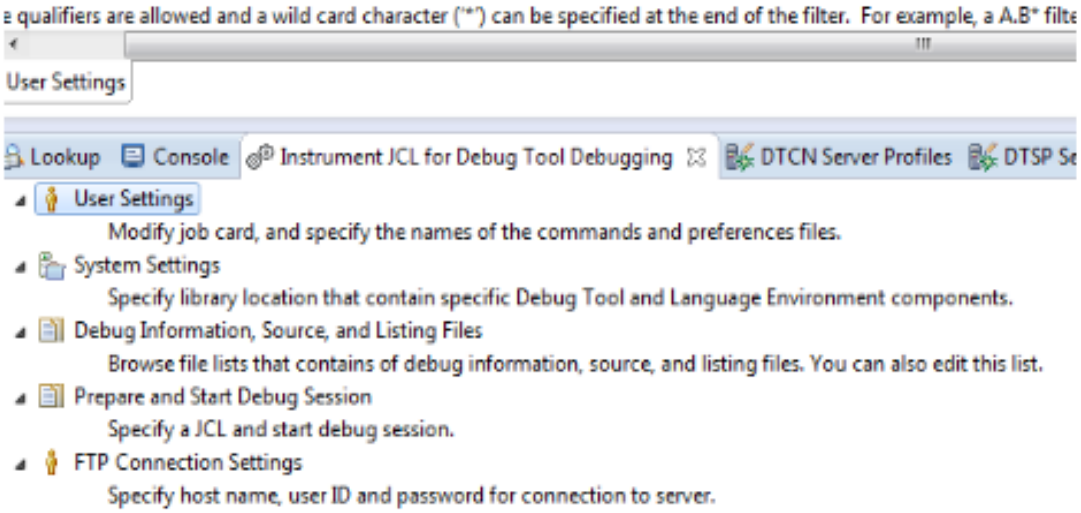
JCL instrumentation plug-in: Overview

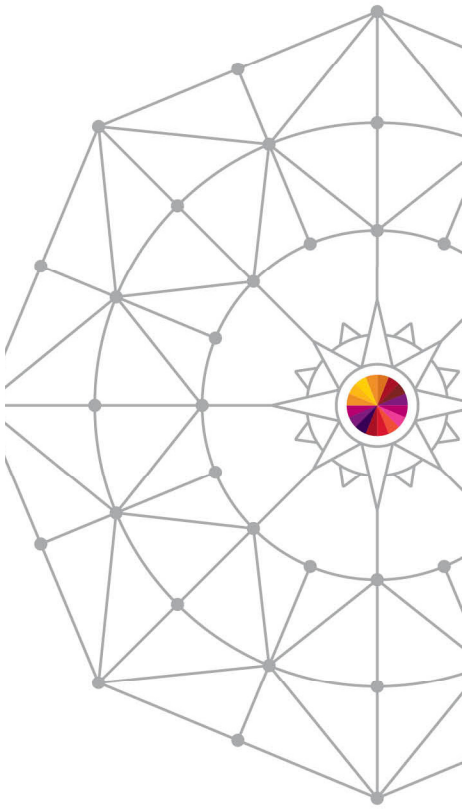
- Provides a UI that includes a wizard that help you instrument your JCL to start Debug Tool for batch jobs.
- It is based on DTU Option 8. JCL for Batch Debugging
- Includes a wizard that takes you thru the steps necessary to instrument the JCL
- Uses ftp to connect to z/OS. (Not common component for this version)
- Uses DT User Exits
- Plugin can be installed in RD/z, z/OS Explorer, and prepackage with PD Tools Studio
- Updates to JCL are temporary but option to save is provided
- Allows you to browse JCL after updates
- Updates clearly delimited in the JCL



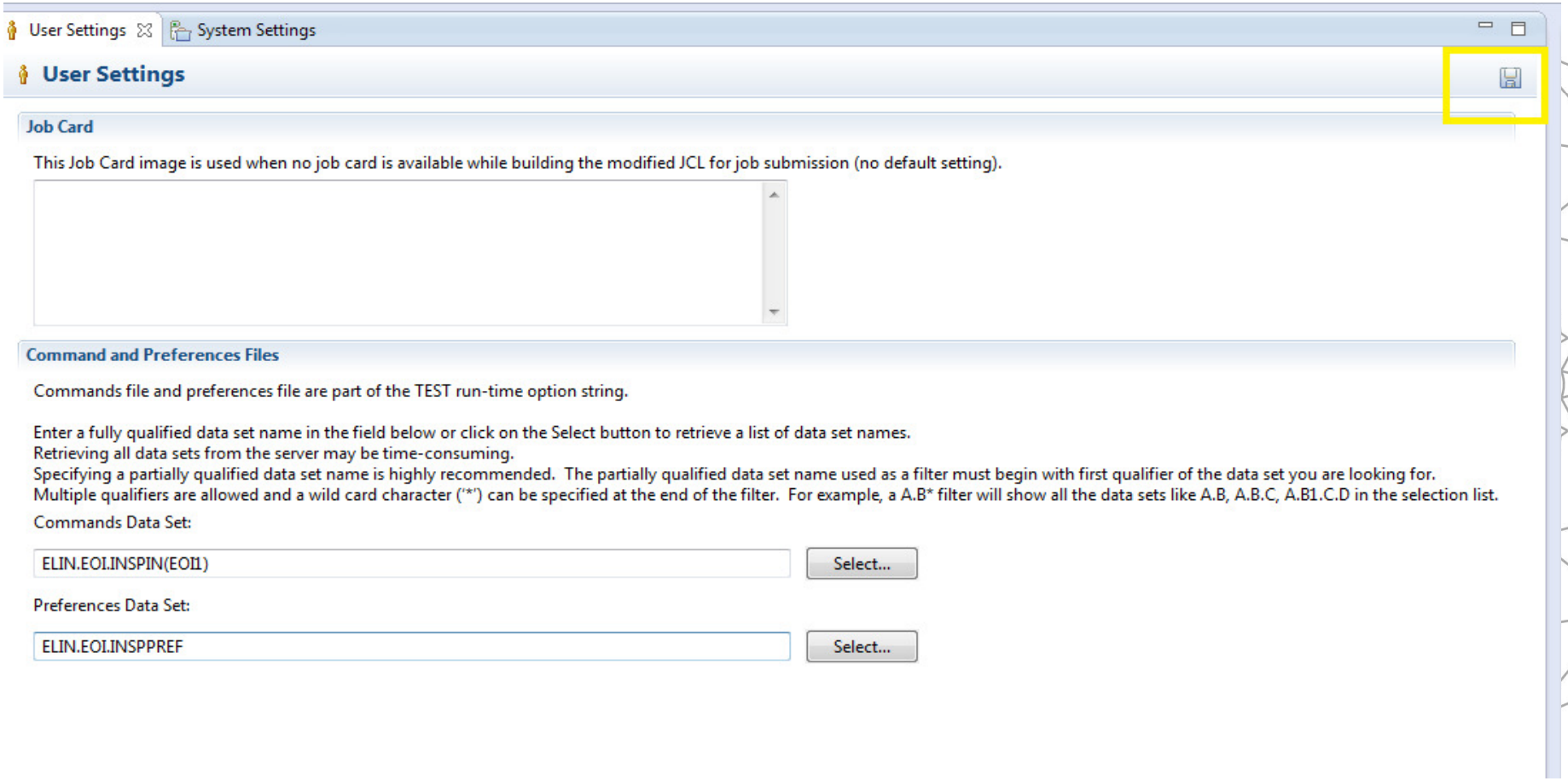
JCL instrumentation plug-in: Overview

Click Window → Show view → Other
Find "Instrument JCL for Debug Tool Debugging"

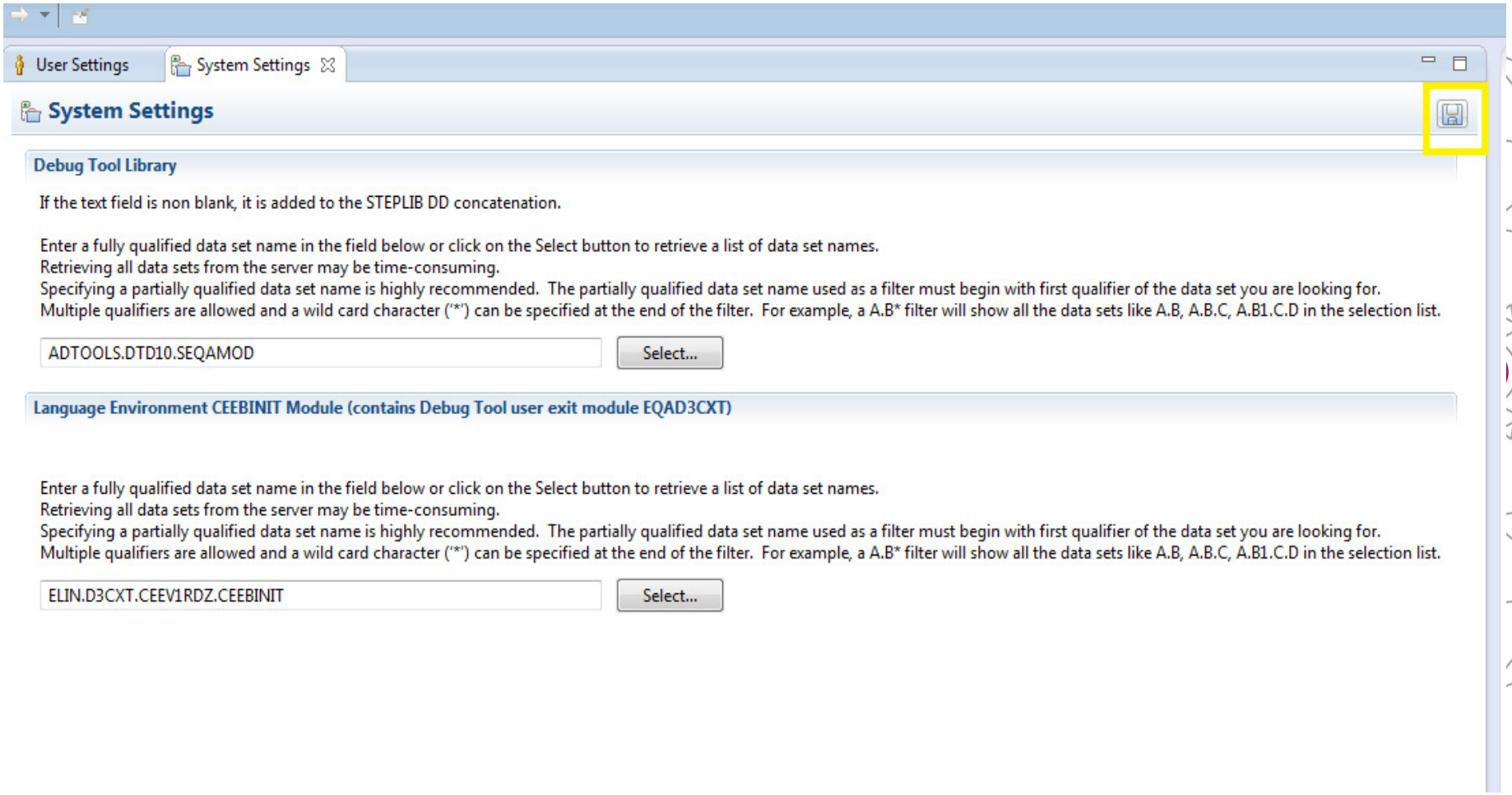




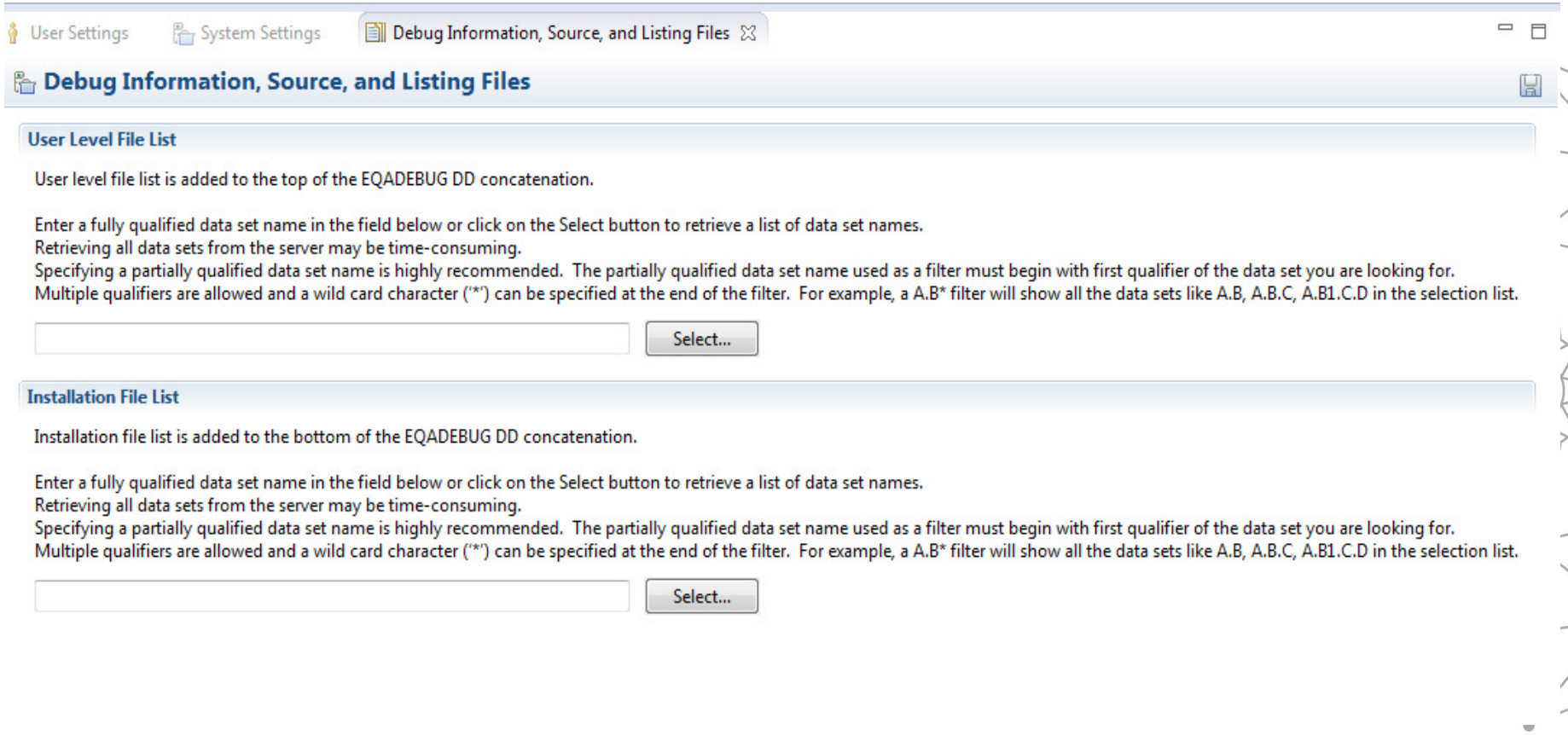
JCL instrumentation: User Settings



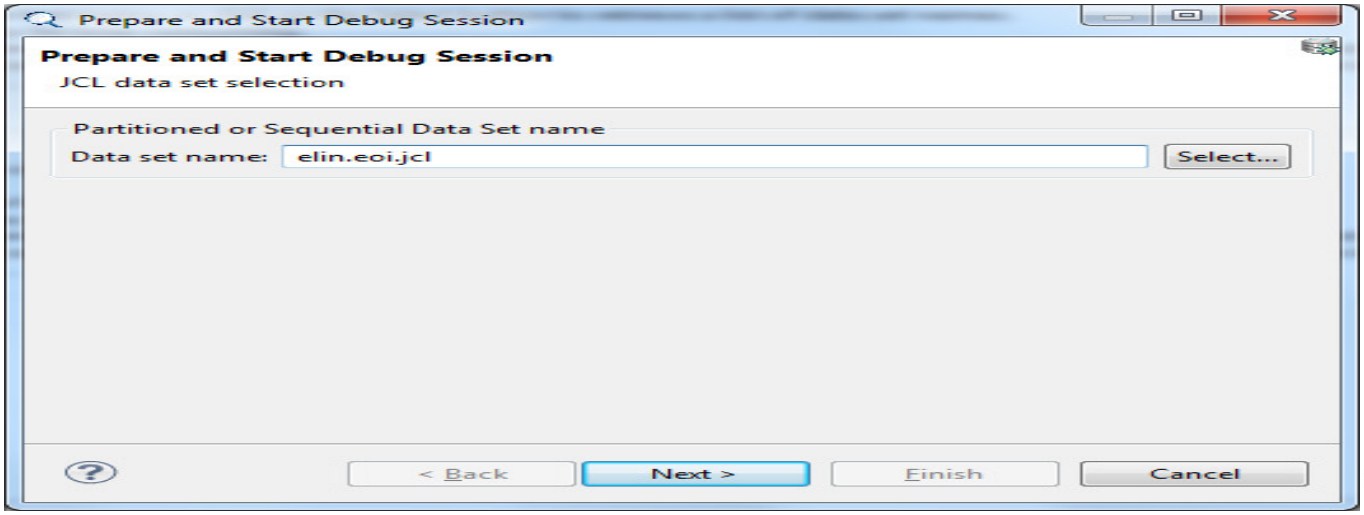
JCL instrumentation: System Settings

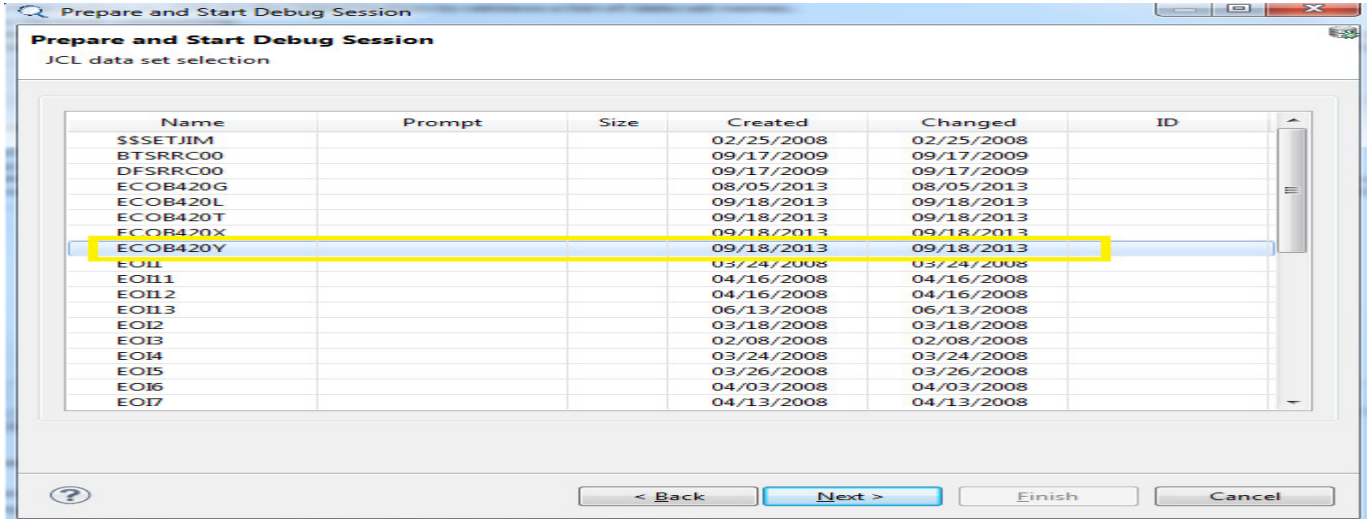


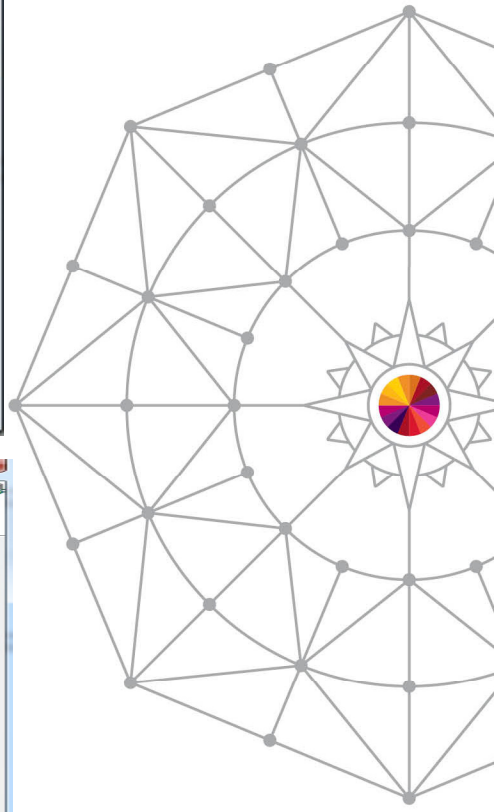
JCL instrumentation: Debug Information, Source and Listing Files



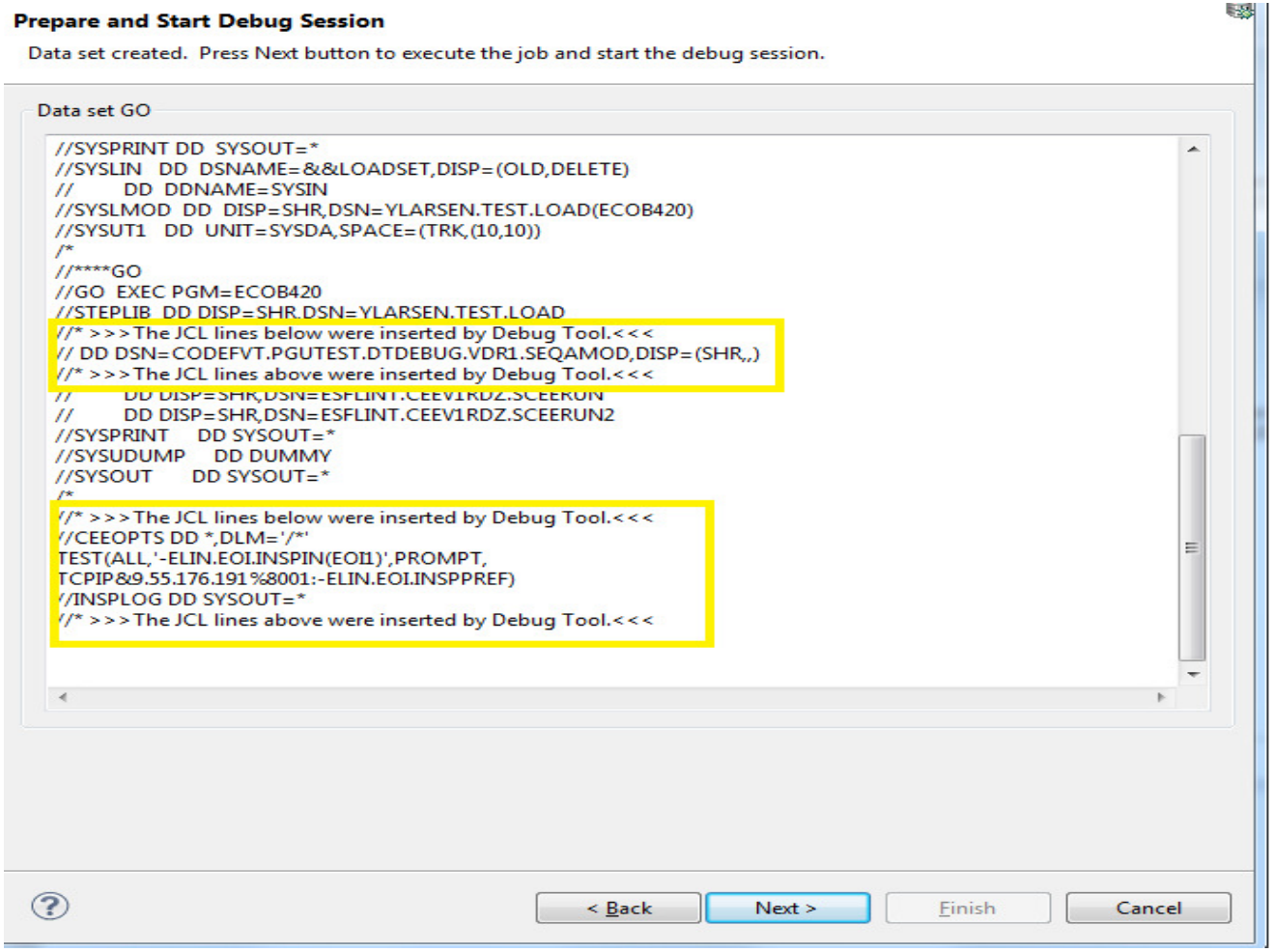
JCL instrumentation: System Settings

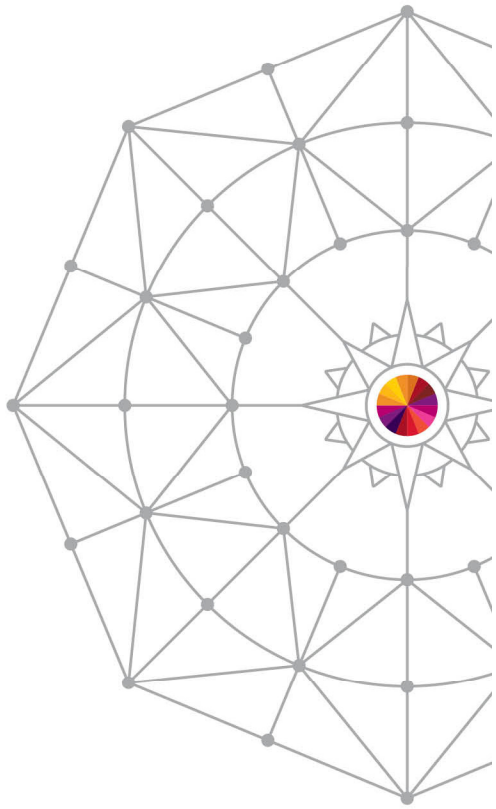




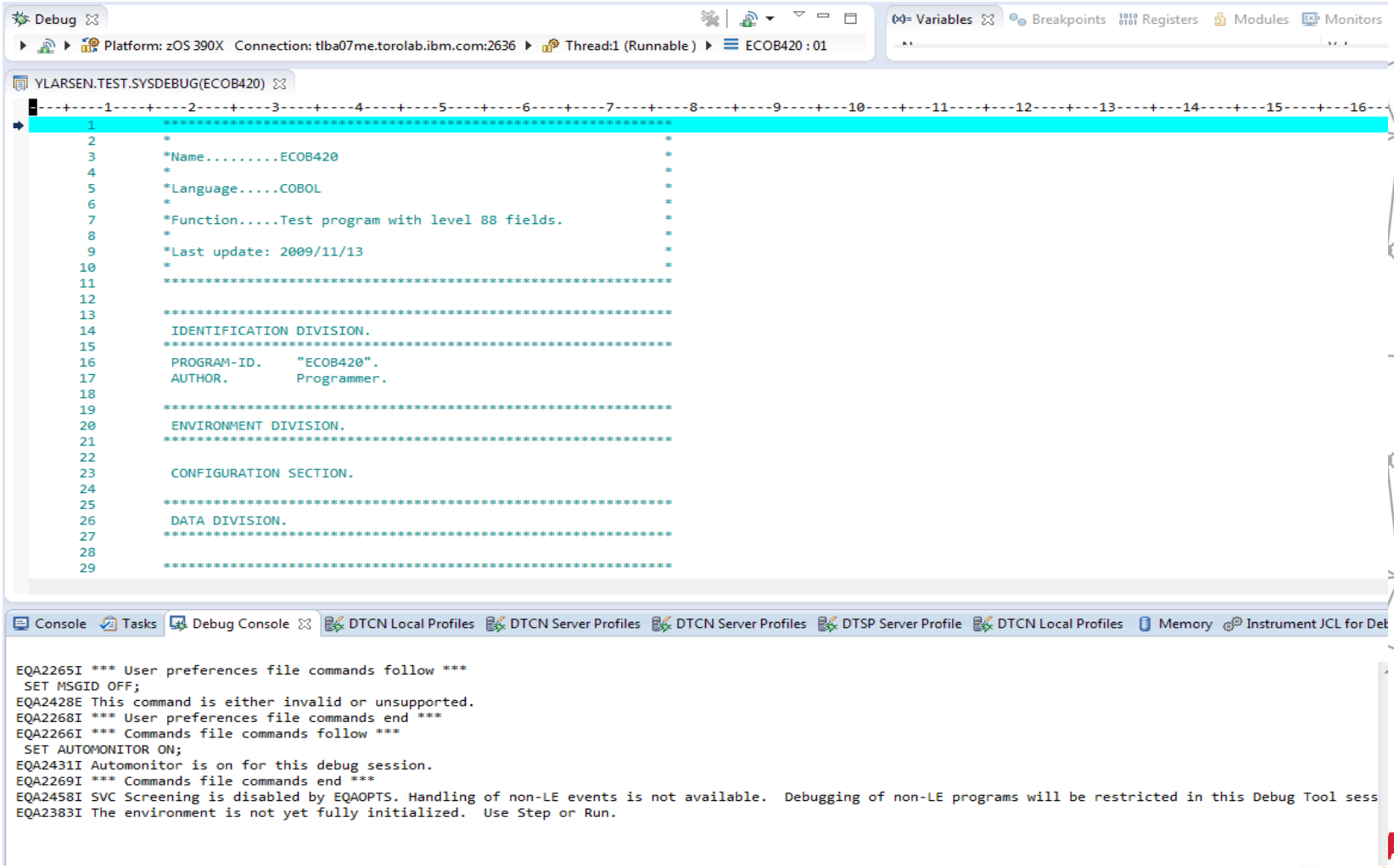


JCL instrumentation: JCL with updates





JCL instrumentation: Debug session up!



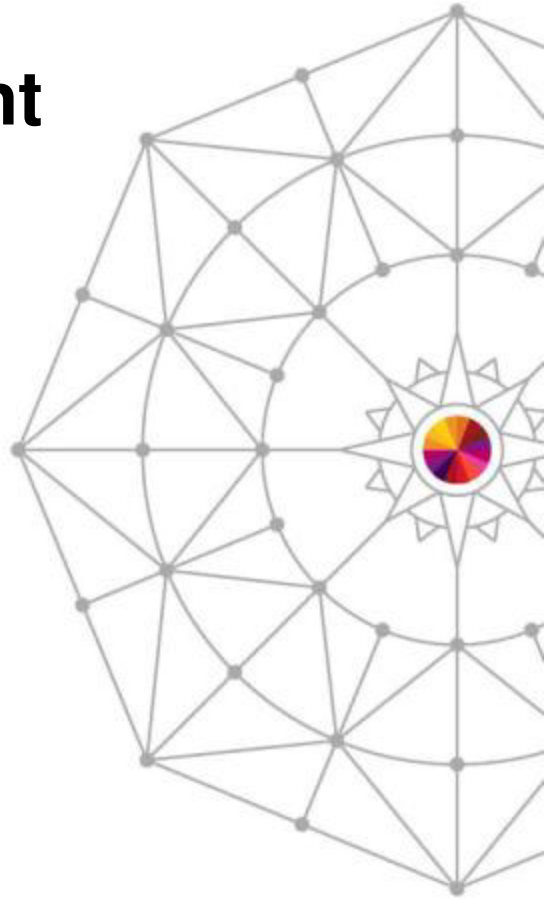
#SHAREorg

SHARE
Technology · Connections · Results

Debug Tool IMS Private Message Region Enhancement

Randy Campbell
IBM

March 10, 2014
Session 14621

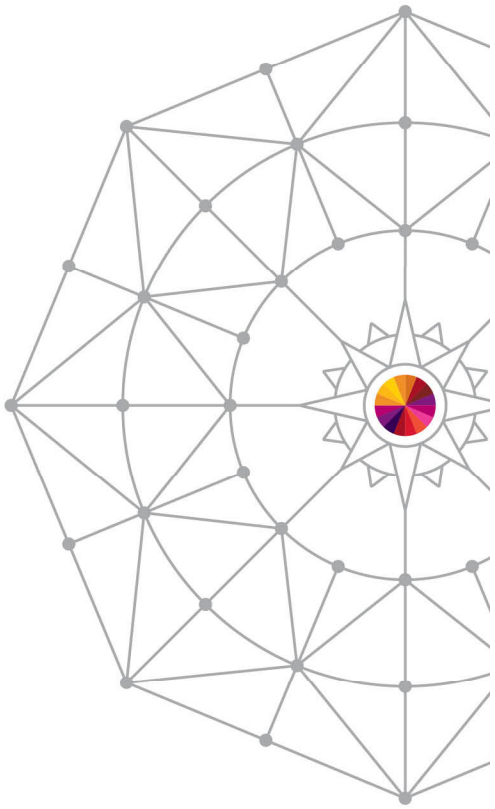


Insert Custom Session QR if Desired.

SHARE in Anaheim

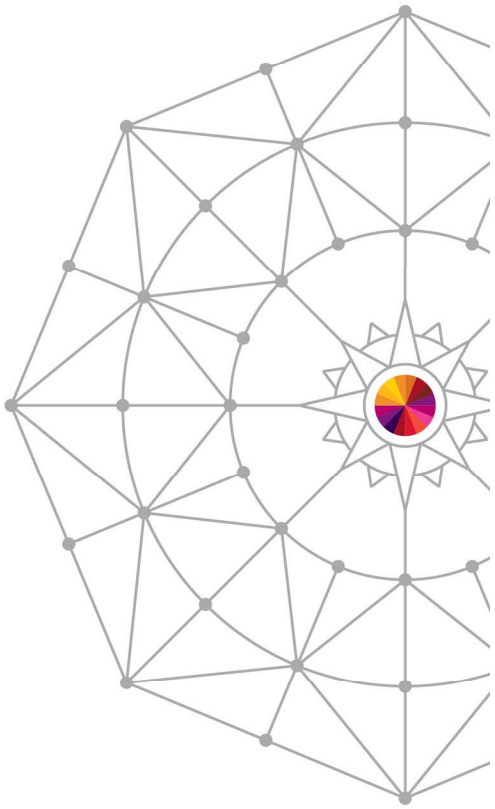
Agenda

- IMS private message region enhancement
  - Customer requirement
  - Proposed solution
  - Demo



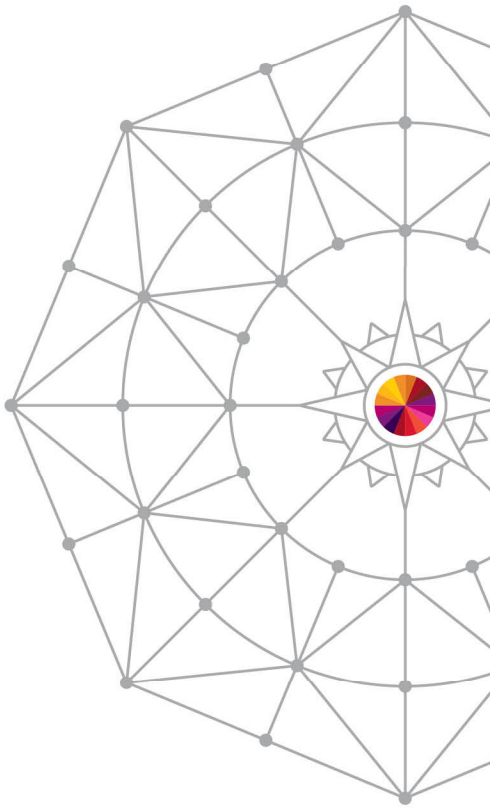
Customer Requirement

- In IMS online environments, one or more users debugging transactions in Debug Tool can monopolize the scheduling environment for a given message class.
- Customers would like a way to isolate debug users in private message regions, serving a class dedicated to Debug Tool. This will allow the normal class to continue scheduling non-debug transactions.



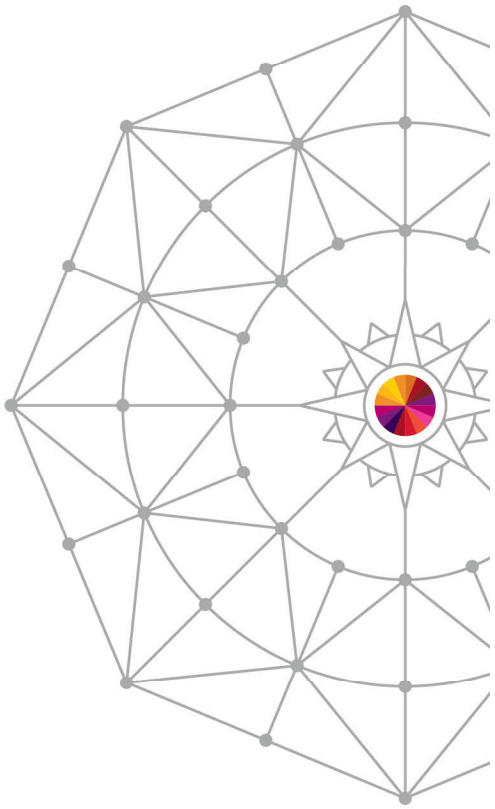
Solution

- Debug Tool will add panels to Debug Tool Utilities to facilitate the following:
  - Allow administrators to create Message Region Templates, with dedicated message classes.
  - Allow debug users to select a Message Region Template to launch as a private message region.
  - Also, allow debug users to specify a transaction to schedule in their private message region and debug.



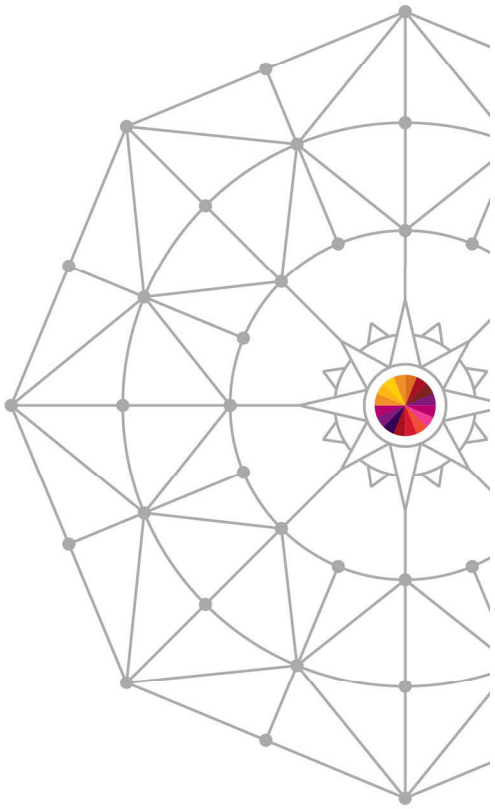
Installer Actions

- The Debug Tool installer will need to perform some customization of DTU and RACF.
  - Set defaults in EQAZDFLT for the default message region template data set, and the default job names for private message regions.
  - Give authority to the EQANBSWT BMP AOI to issue the /START, /STOP, /ASSIGN and /DISPLAY IMS commands.
  - Authorize administrators to the EQADTOOL.IMSTEMPCREATE FACILITY.



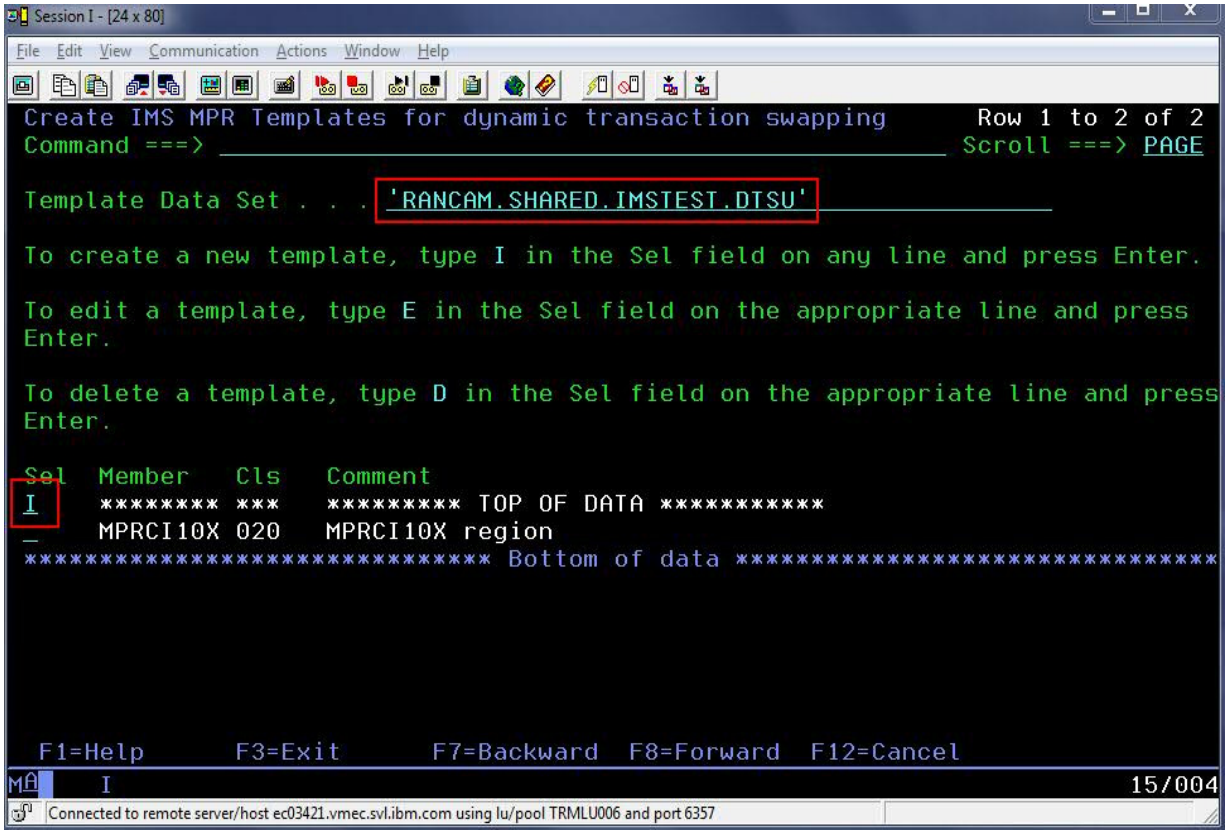
Administrator Use Case

- Administrator needs to create a new message region template.
- Use DTU option 4, sub-option 4.
- Copy from the JCL of an existing IMS message region.



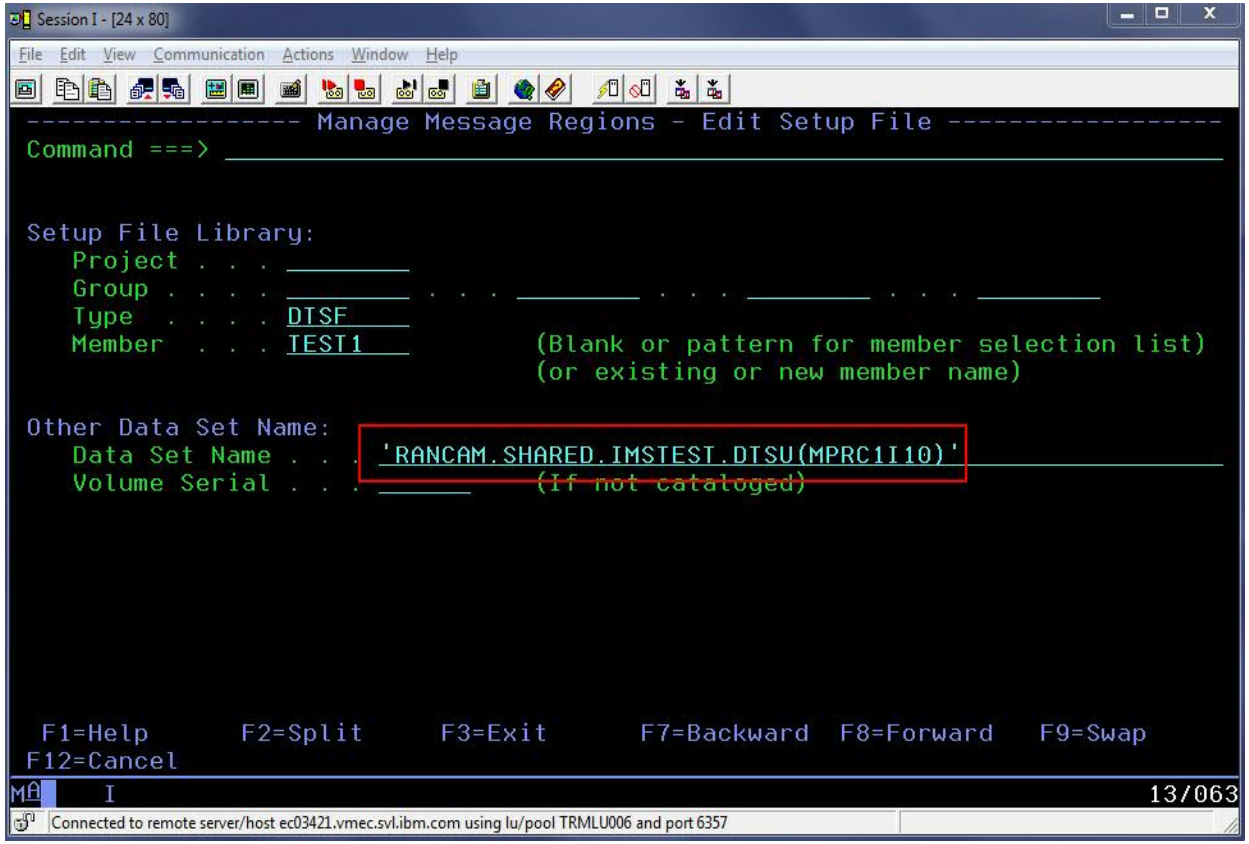
Administrator panel

- This is the panel the administrator sees after selecting DTU option 4.4.
- The administrator may overtype the name of the template data set.
- To create a new template, the administrator enters “I” in the table.



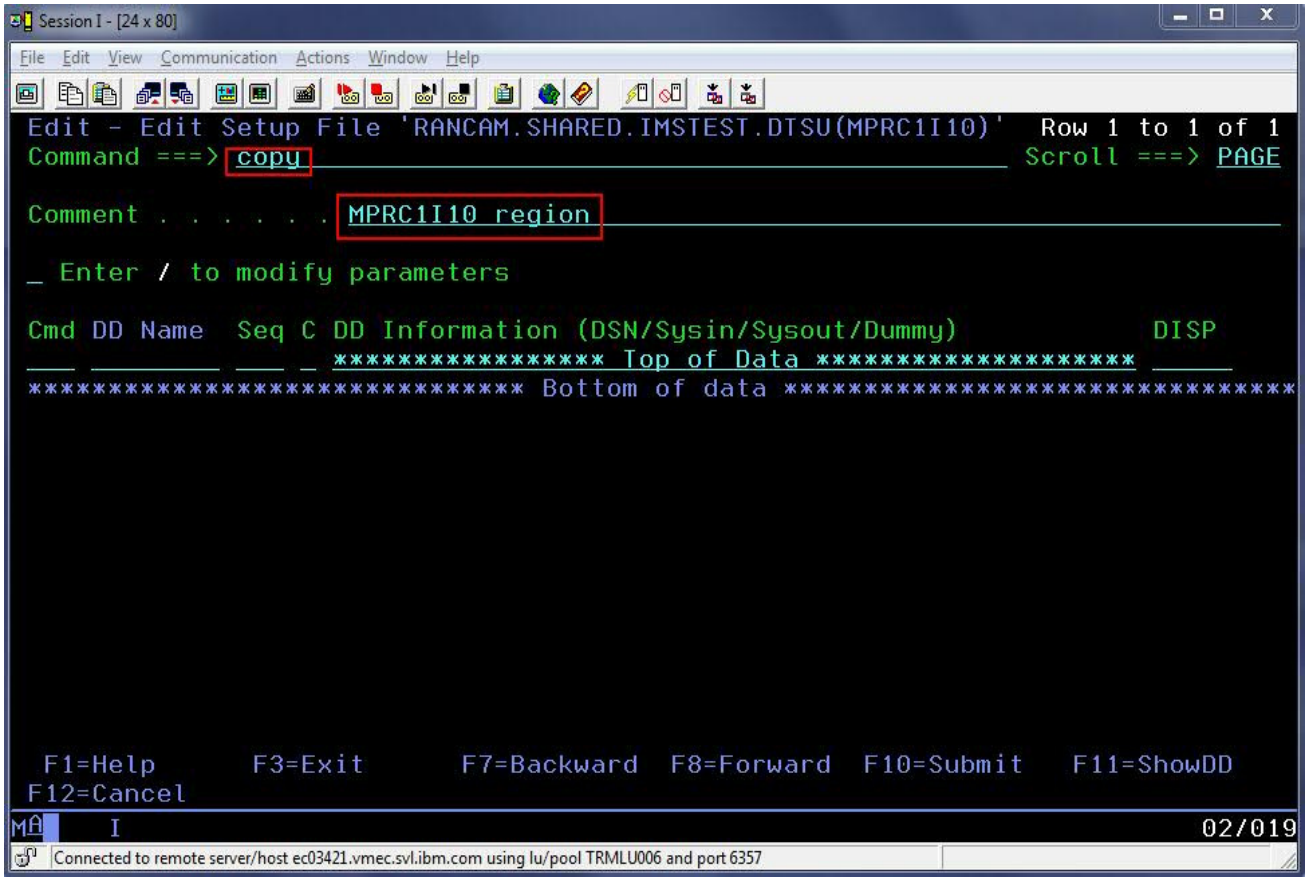
Supply data set name for template

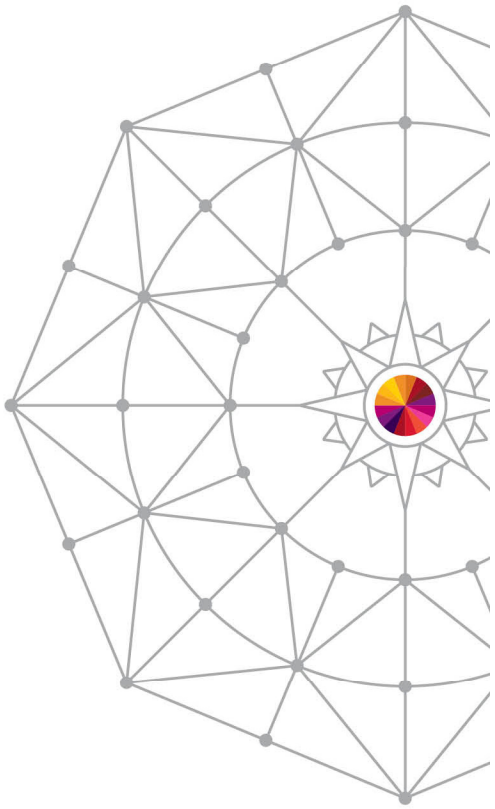
- The data set name will be pre-filled with the template name. The administrator adds a member name and presses Enter.



Enter the COPY command

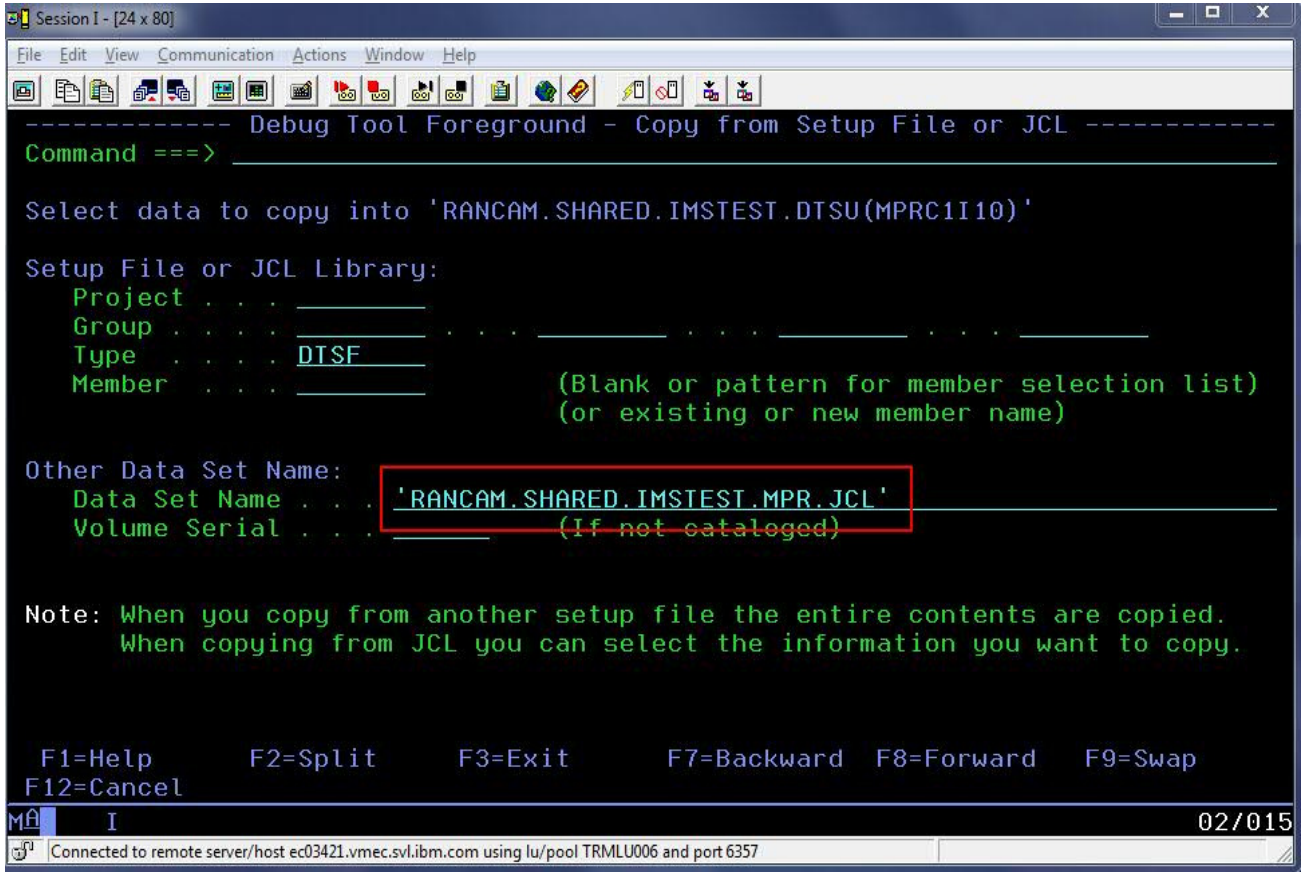
- The administrator types in a region comment, and then enters the “copy” command and presses Enter.

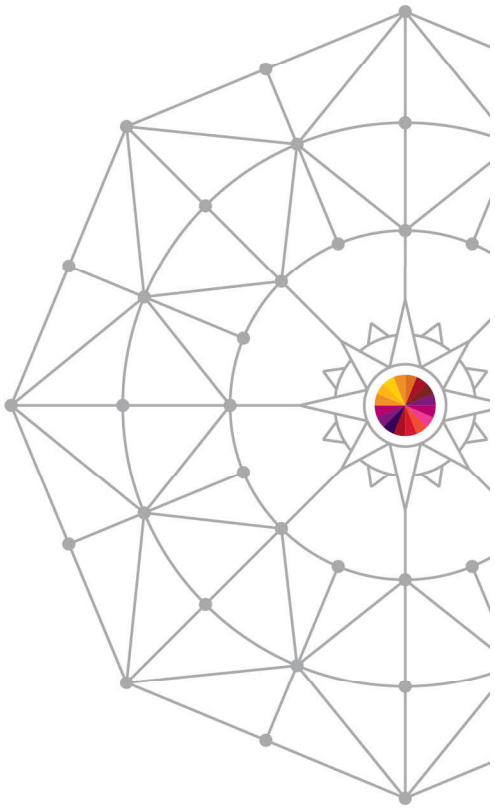




Supply JCL data set name

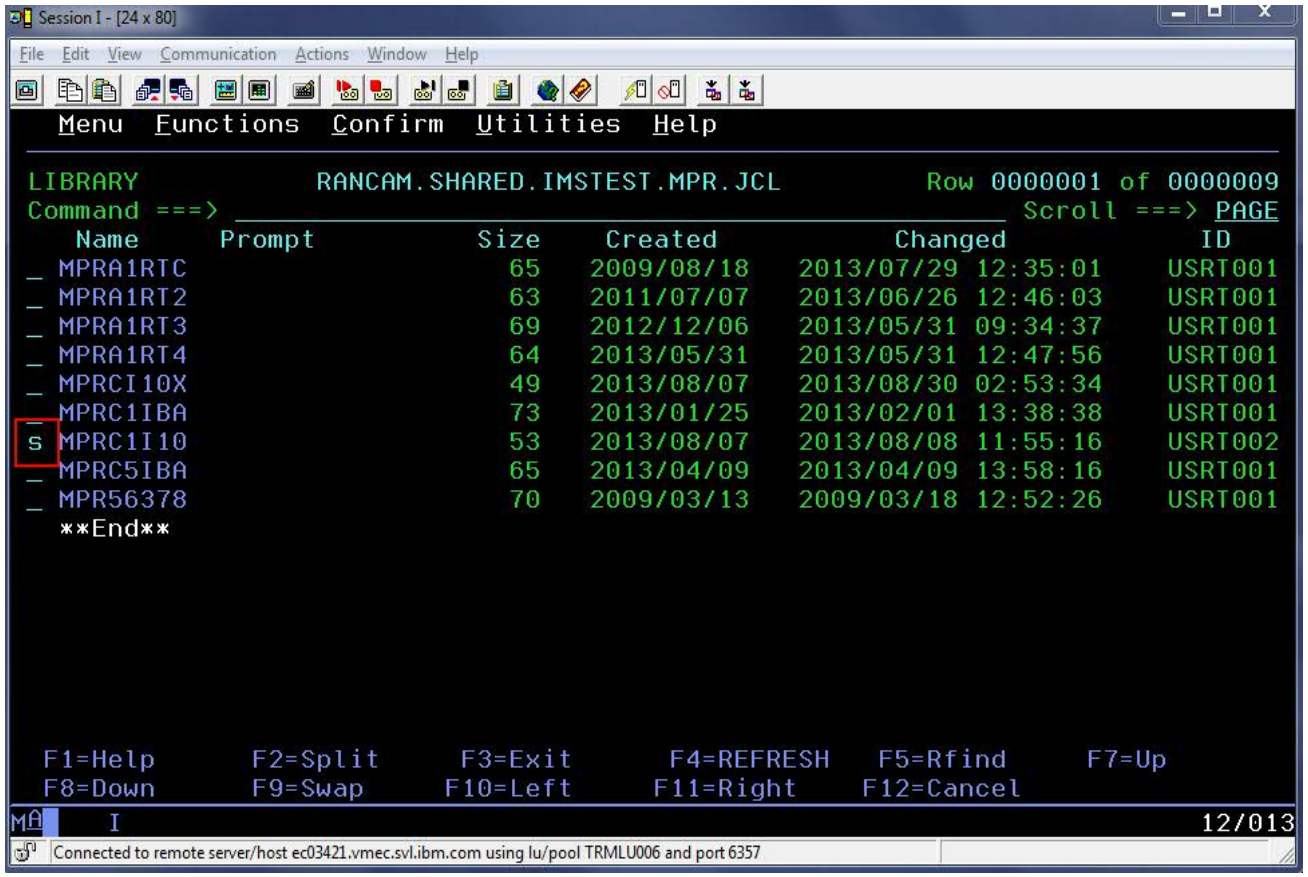
- The administrator will the type the name of a data set which contains IMS MPR JCLs and press Enter.





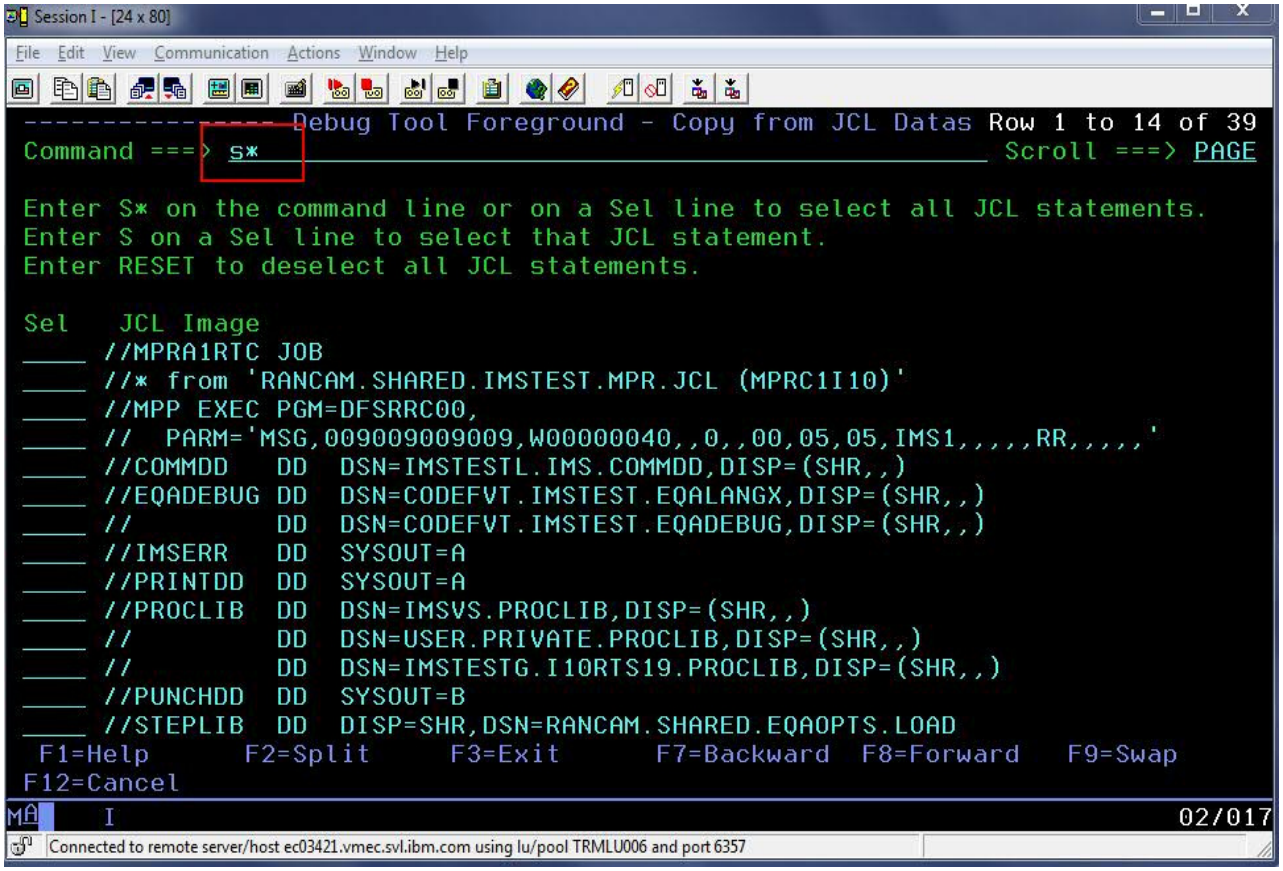
Select member from JCL data set

- The administrator selects the member to copy and presses Enter.



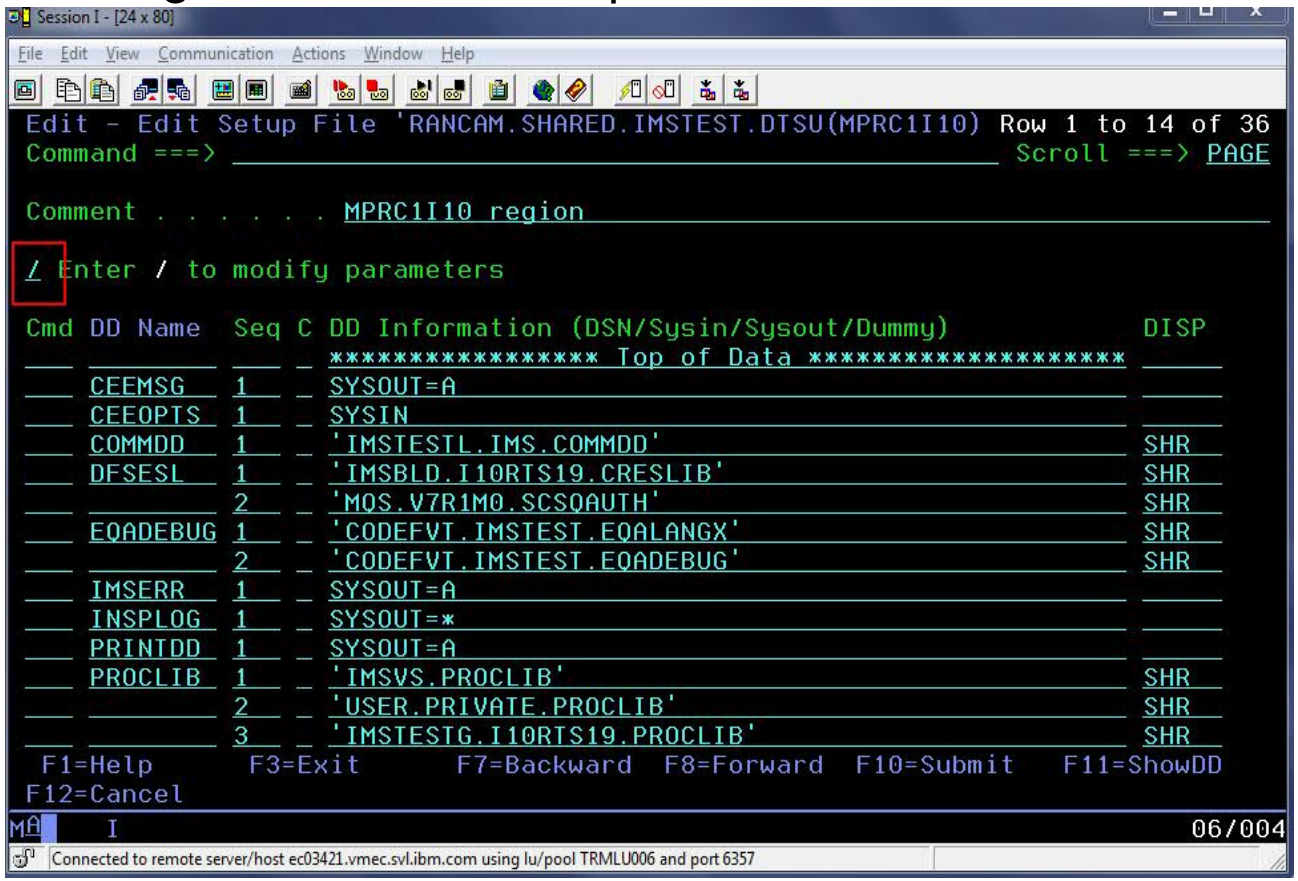
Select JCL cards to import to template

- This will present a panel with the JCL cards for the MPR.
- To select the entire job, the administrator types the command S* and hits Enter, and then presses PF3 to populate the template from the JCL.



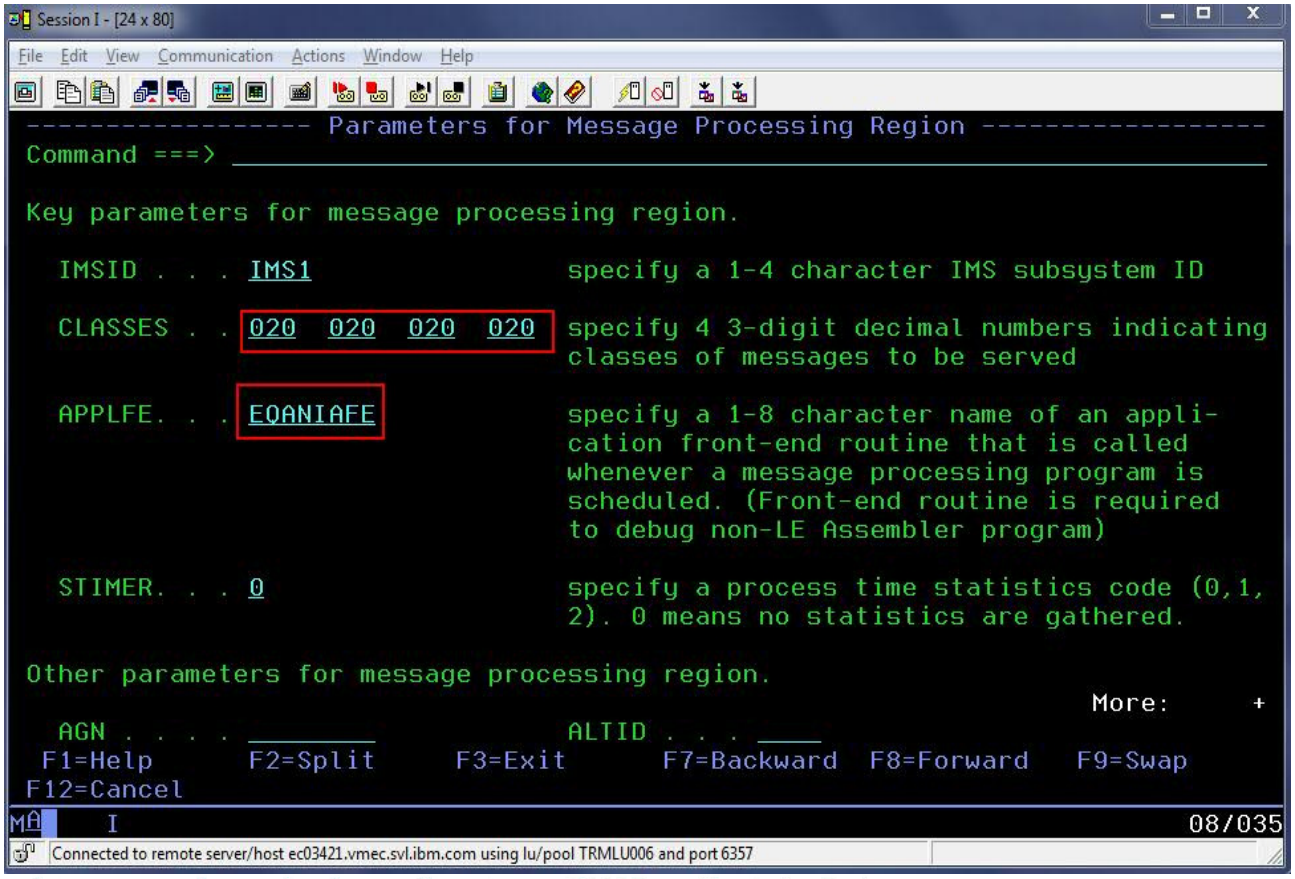
Modify parameters to set message class

- The administrator types a forward slash (/) to modify the parameters. This allows the administrator to assign a message class to the template.



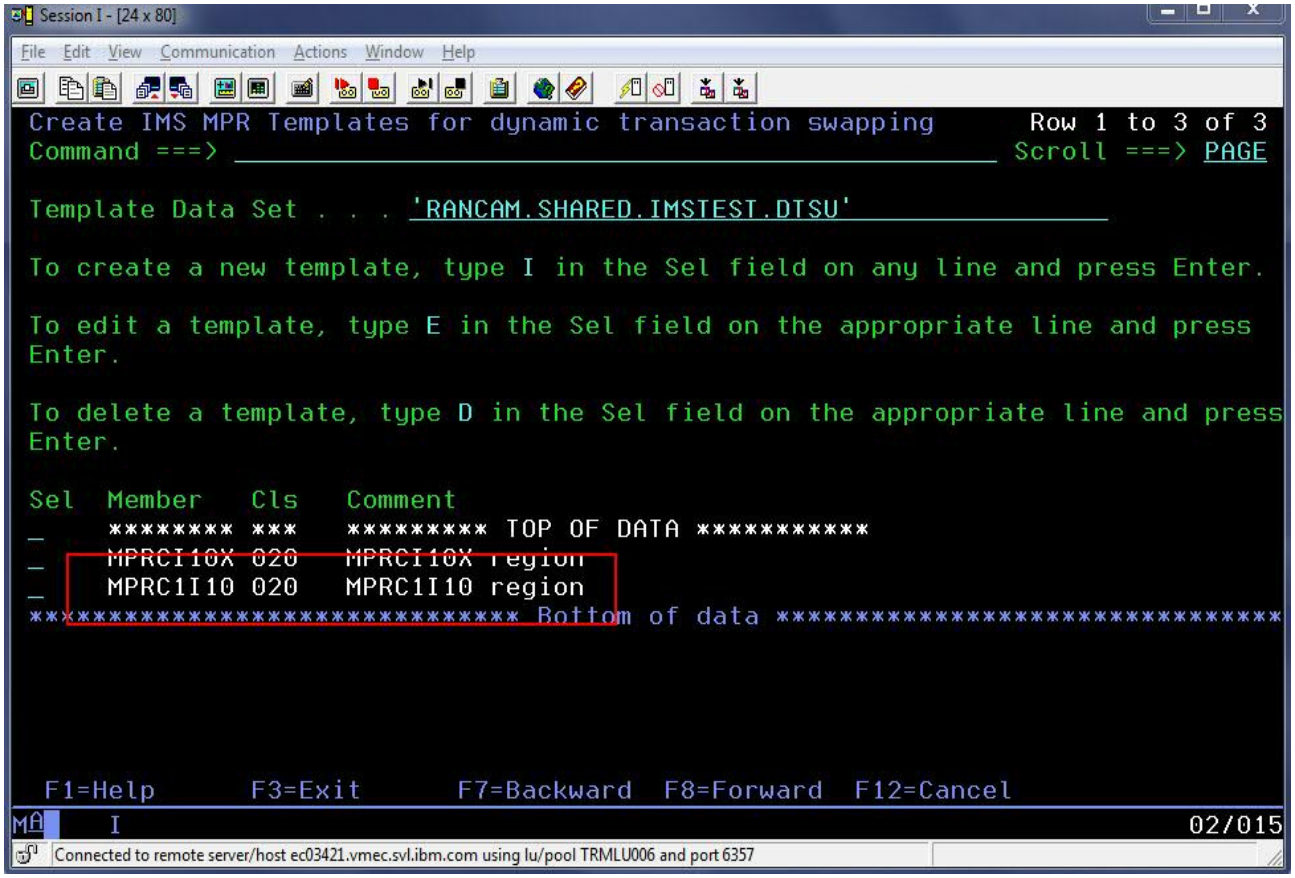
“Classes” field will contain new message class.

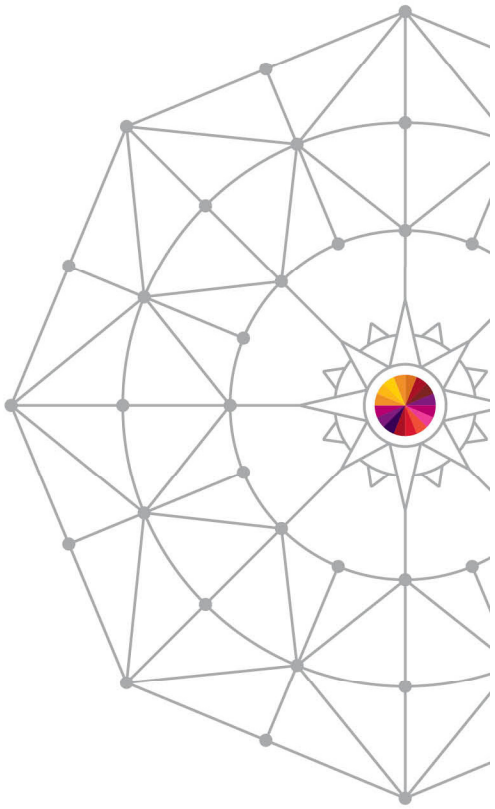
- The administrator changes the message classes.
- Also, note that APPLFE of EQANIAFE is required and will be forced on if not specified.



New template is now in the list

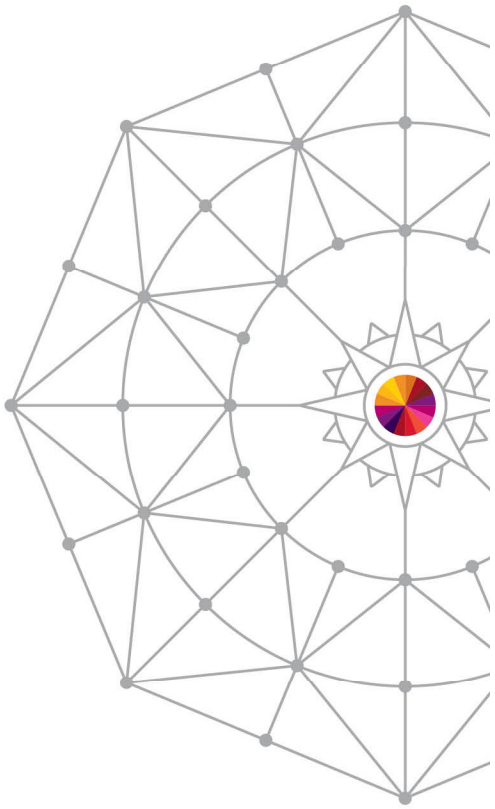
- Once the administrator presses PF3 to back all the way out, the new template will appear in the list.





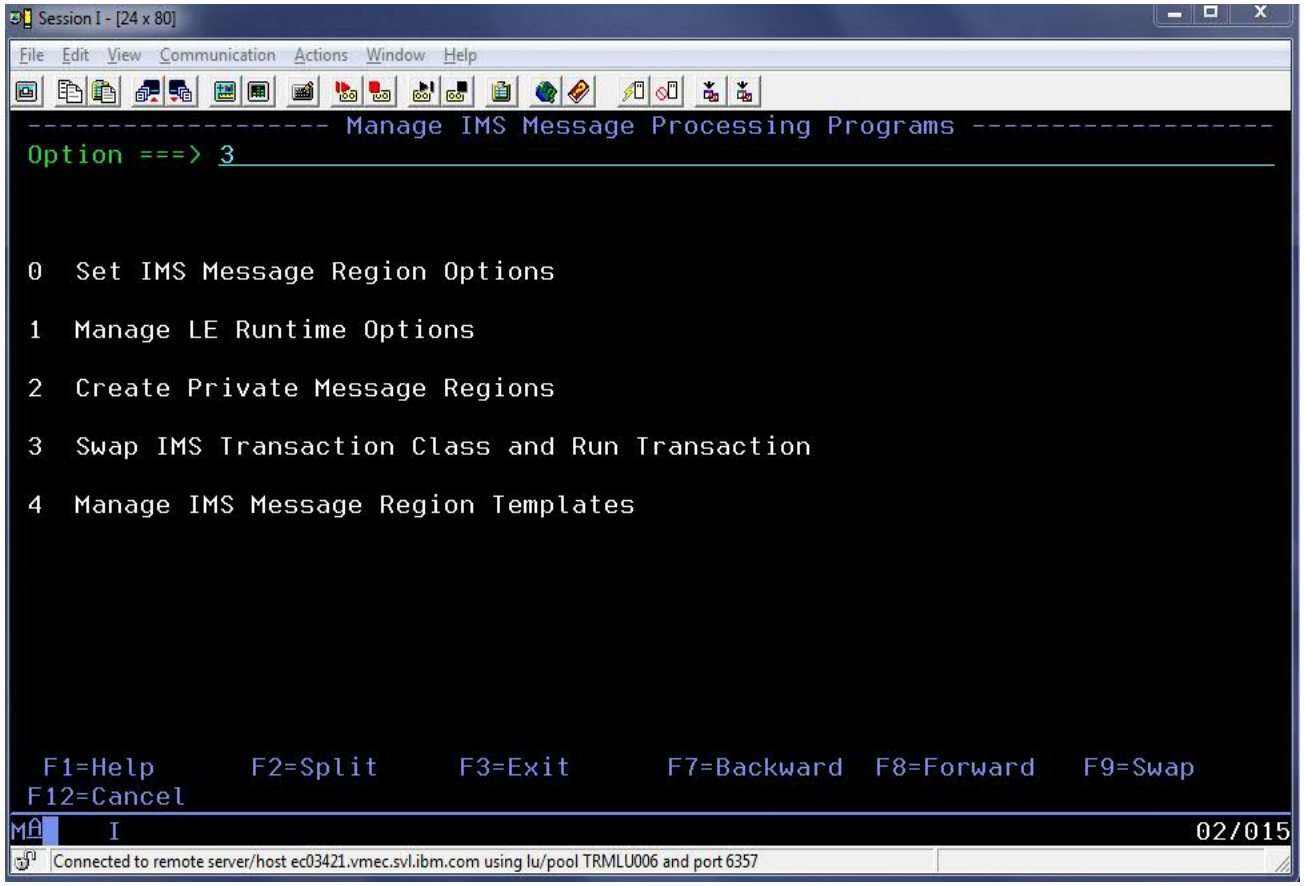
Developer Use Case

- A developer knows that a certain transaction has a problem.
- The developer would like to debug to determine the location & cause of the failure, in a private message region.
- After debugging, the developer codes a fix and would like to test it without affecting other users.
- The developer would use a private message region template to accomplish these tasks.



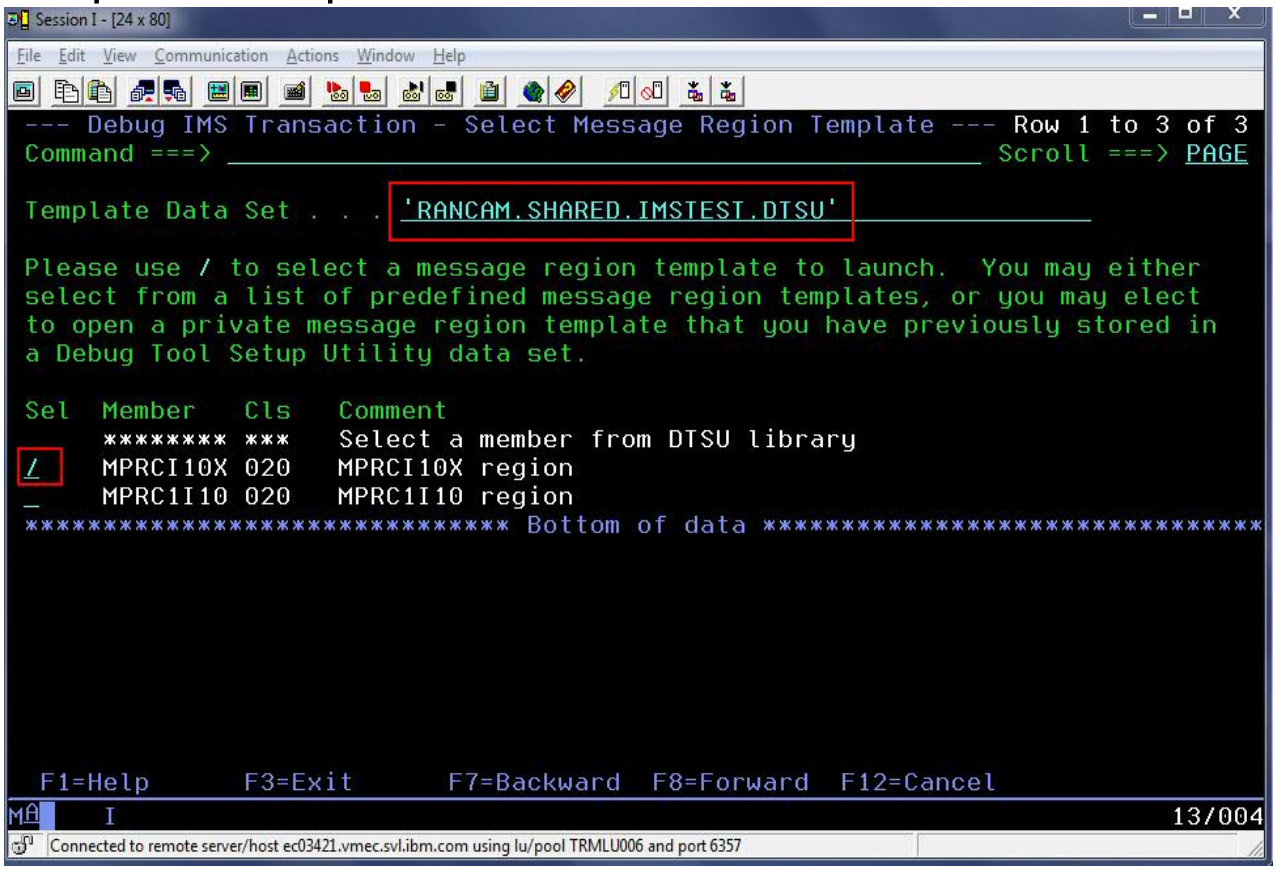
Select DTU option 4, sub-option 3

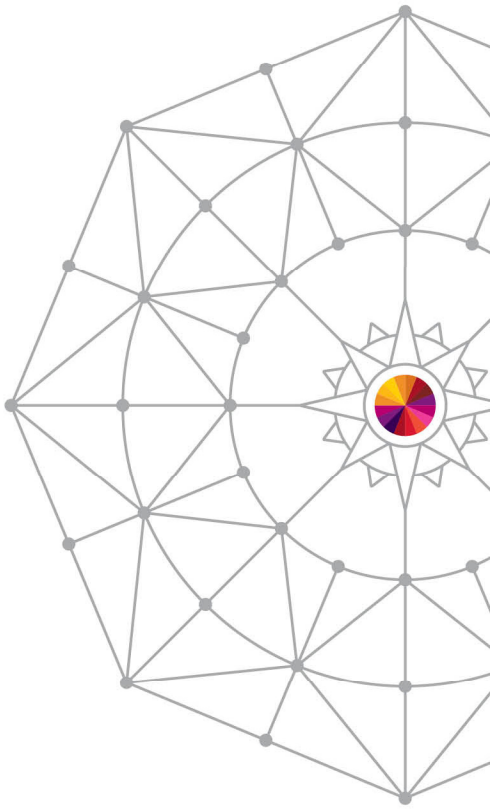
- User selects sub-option 3 to test with a private message region template.



Select a pre-defined IMS region template

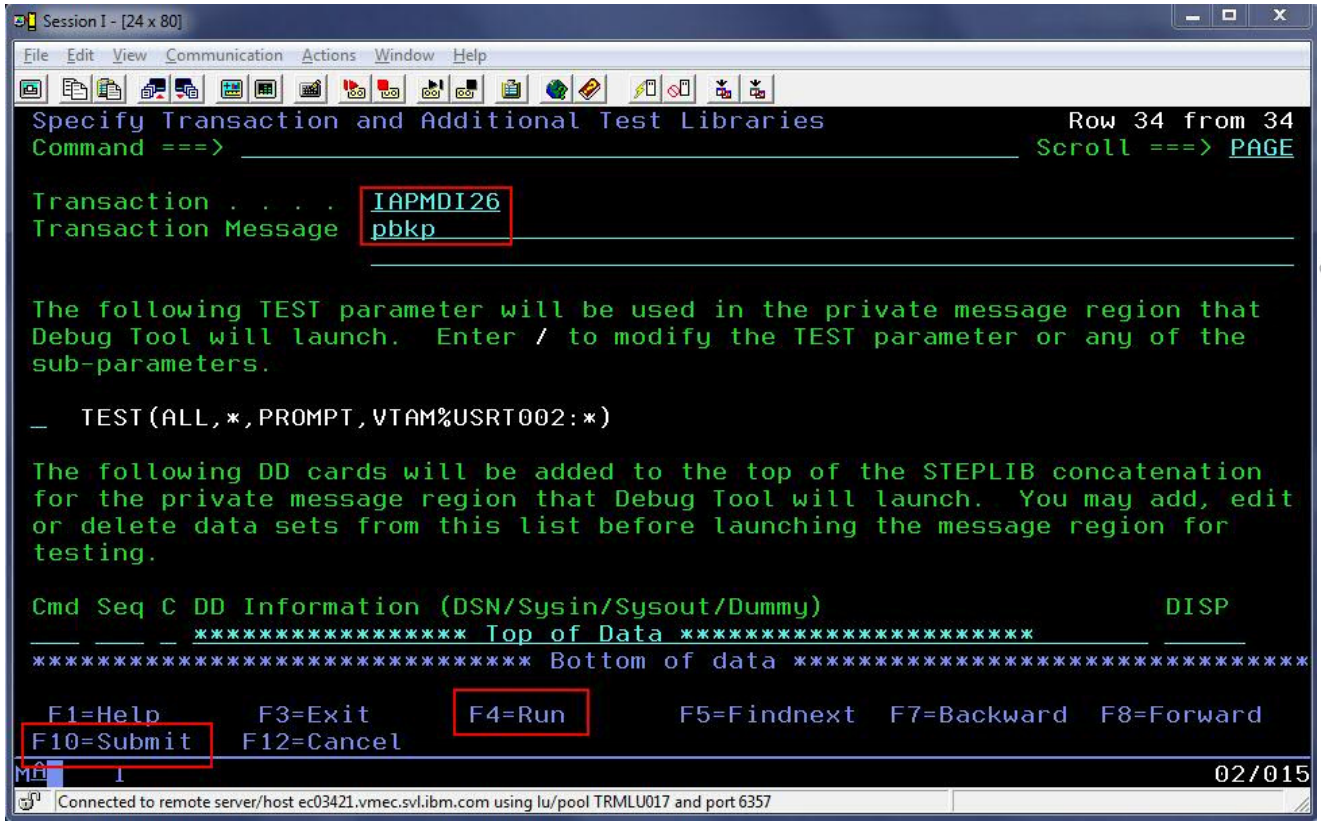
- User can overtype the template data set name, if desired. Then, user enters forward slash (/) next to the desired template and presses Enter.





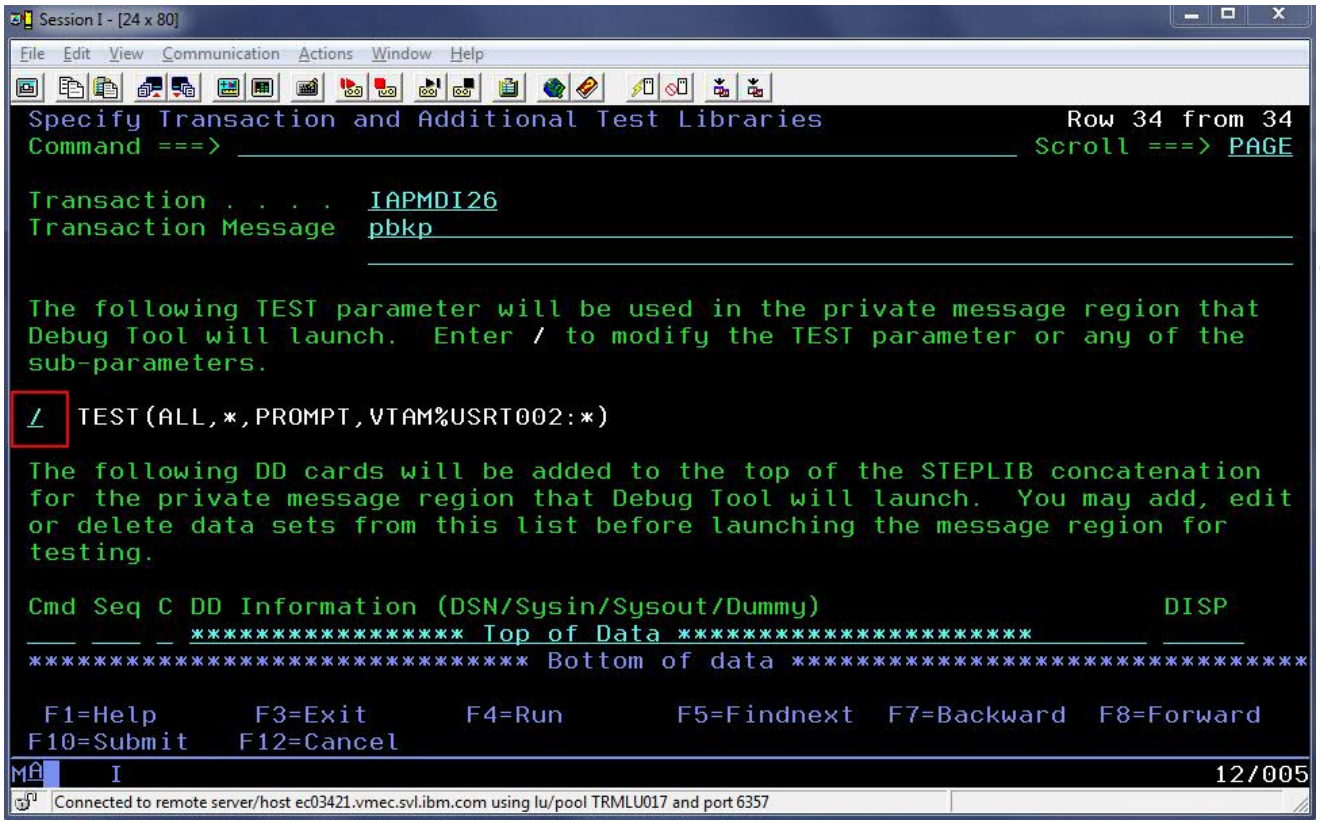
Supply a transaction name and message

- To debug with the template, the user will fill in the transaction and any message for the transaction, then hit PF4 to “Run” or PF10 to “Submit”.



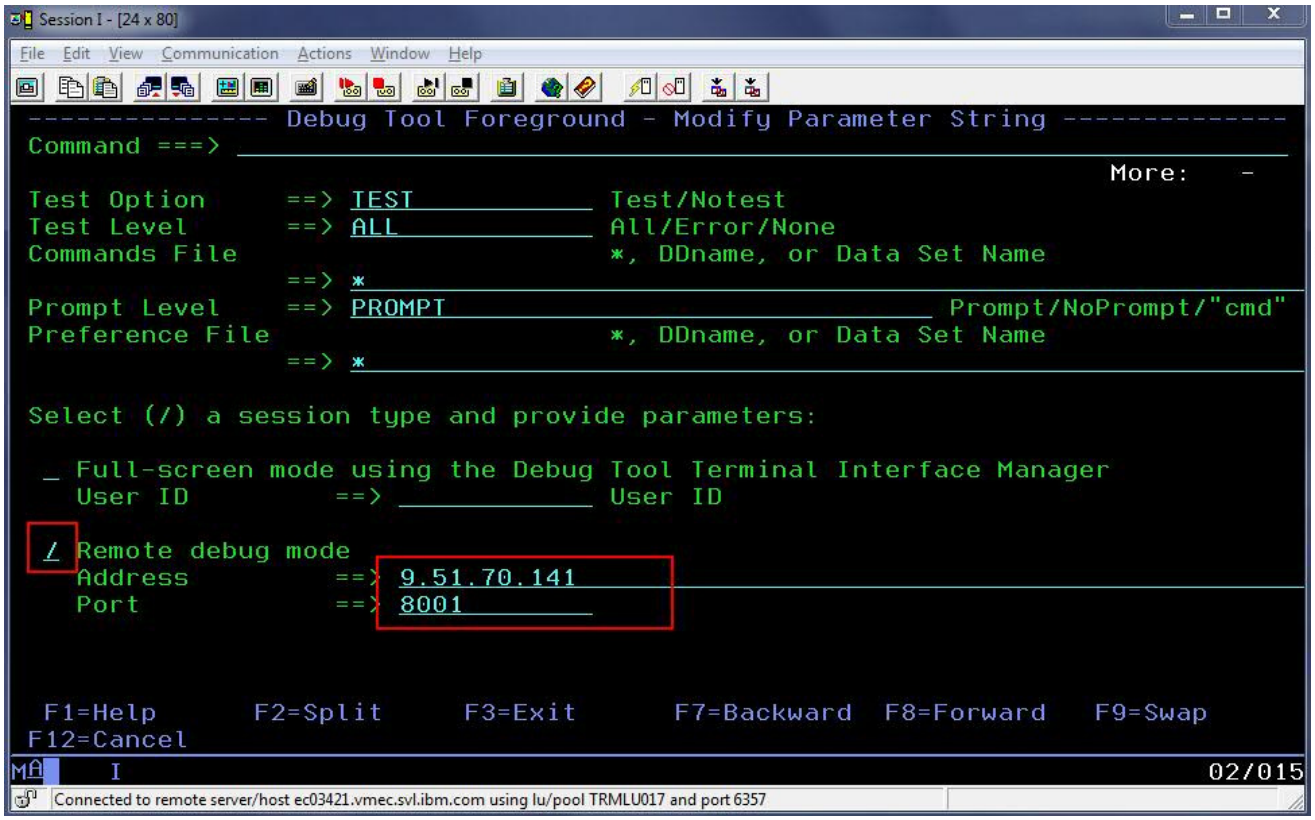
Edit the TEST parameter string

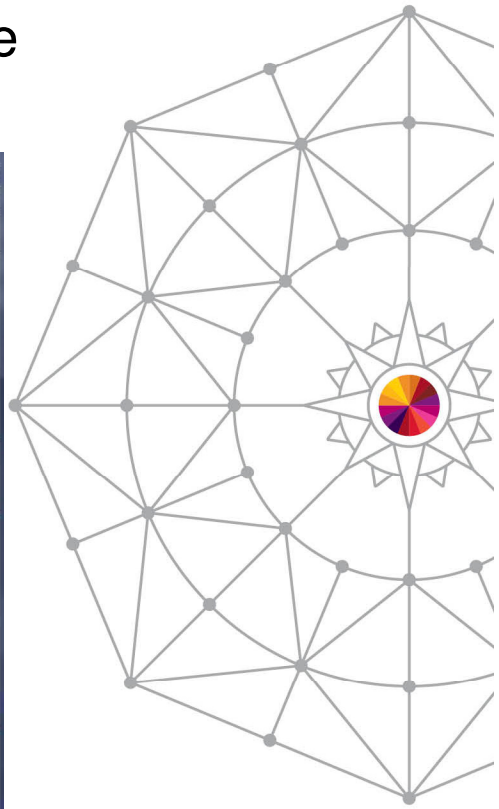
- The user can modify the TEST parameter by placing a forward slash (/) next to the existing parameter and pressing Enter.



Panel to modify the parameter string

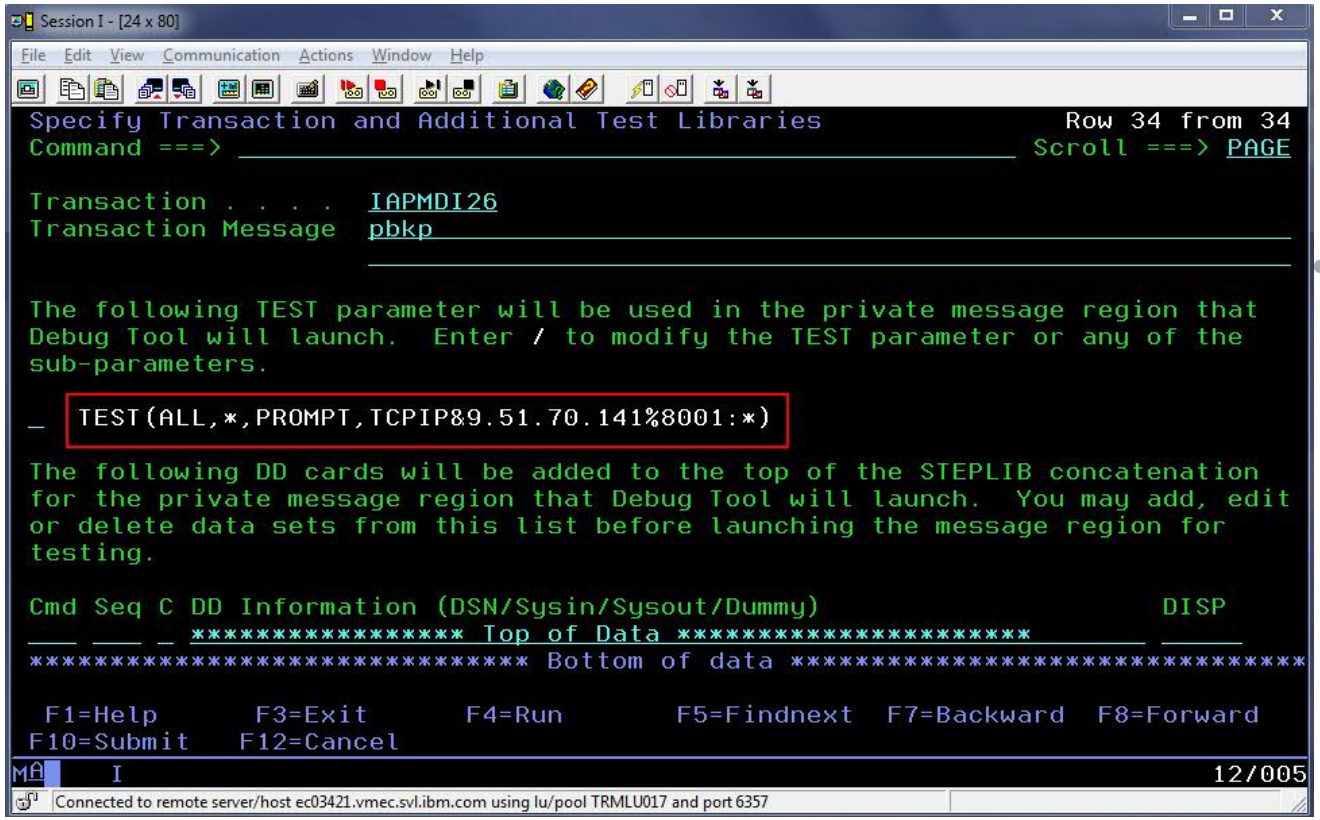
- On this panel, the user can change the TEST string to specify a commands or preference file, and also change the “session type” (VTAM or via the remote plugin).





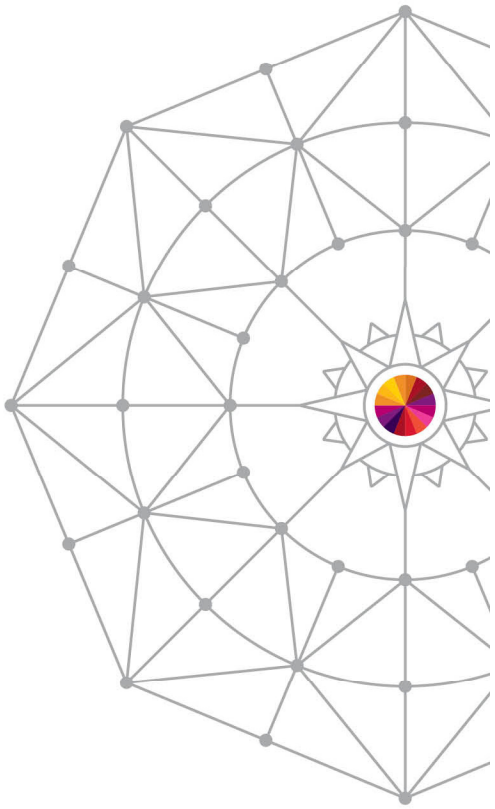
TEST parameter has been modified

- Note that the TEST parameter will now direct debug sessions to the remote plugin at the specified IP address and port.



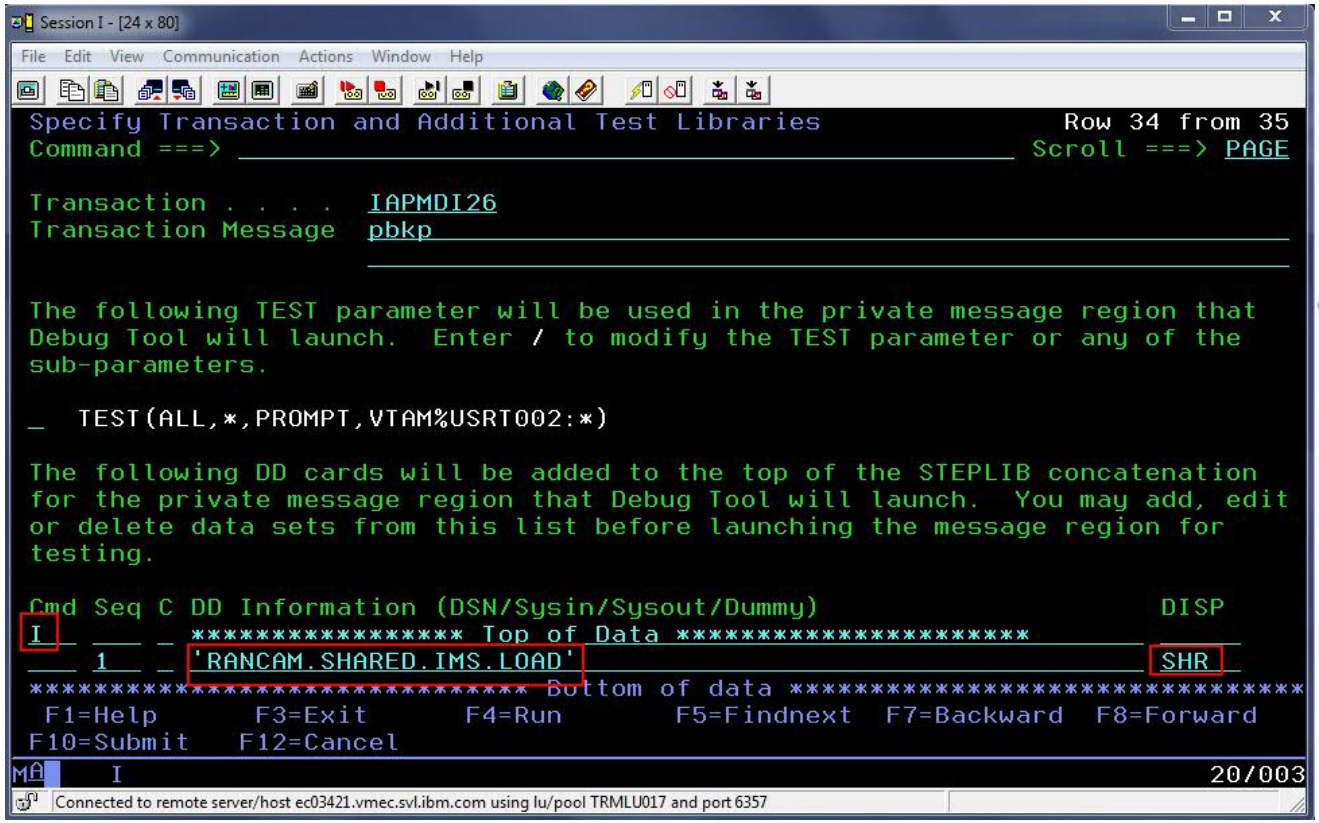
Notes on “Run” and “Submit”

- Both options run an IMS BMP called EQANBSWT.
- EQANBSWT will launch two message regions based on the selected template, will assign the selected transaction to the new message class, and will send a message to the transaction.
- The first region will hard code the TEST parameter selected by the user.
- The second region will hard code NOTEST.
- “Run” executes EQANBSWT in the foreground.
- “Submit” shows JCL to run EQANBSWT, which the user can SUBMIT to JES.



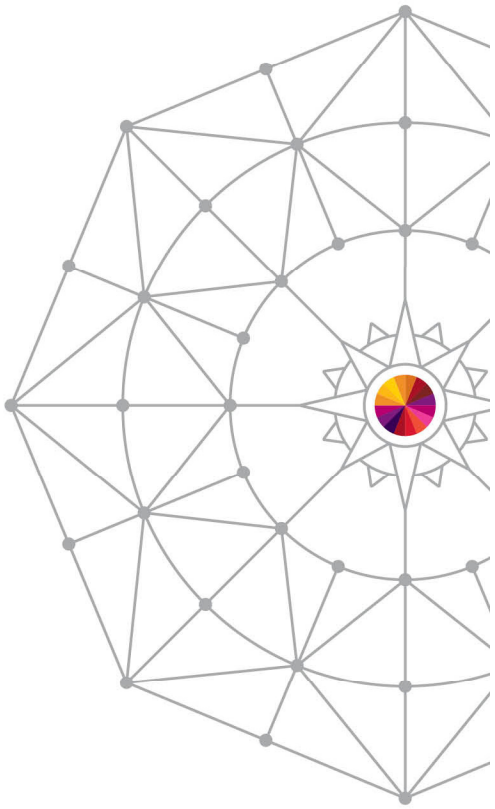
Adding a personal library to template JCL

- The user determines the cause of the failure and codes a fix. To test the fix, the user may use a private message region template with an additional test library. The user accesses the same panel as before, but now uses “I” to insert a library at the top of the STEPLIB concatenation, and then presses PF4 or PF10.



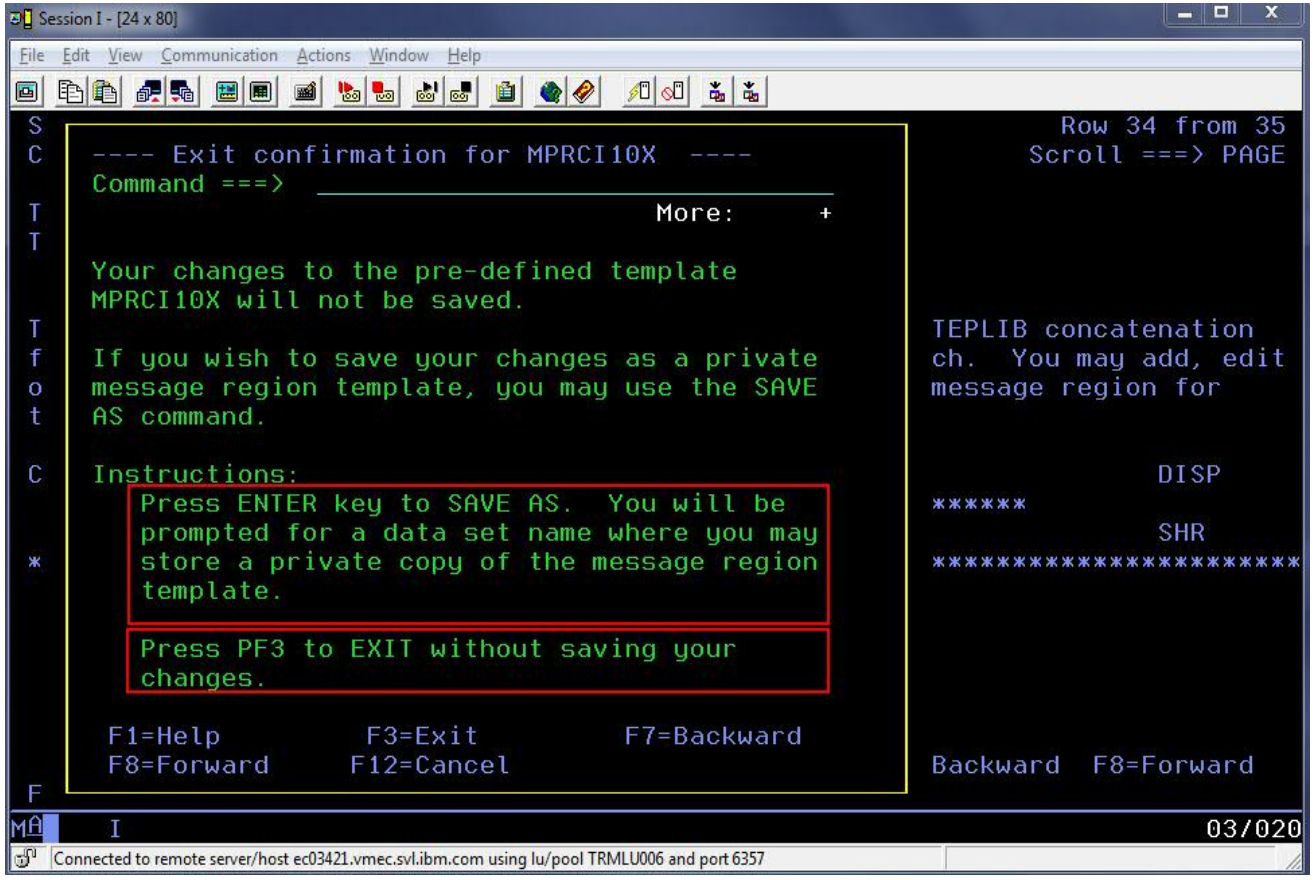
Saving a test case

- After successful completion of testing, the user may wish to save a test case for later.
- To do this, the user may use the “SAVE AS” command.
- If the user attempts to exit without saving, he/she will be prompted.
- To use the test case later, the user may choose “Select a member from DTSU library” on the “Select Message Region Template” panel.



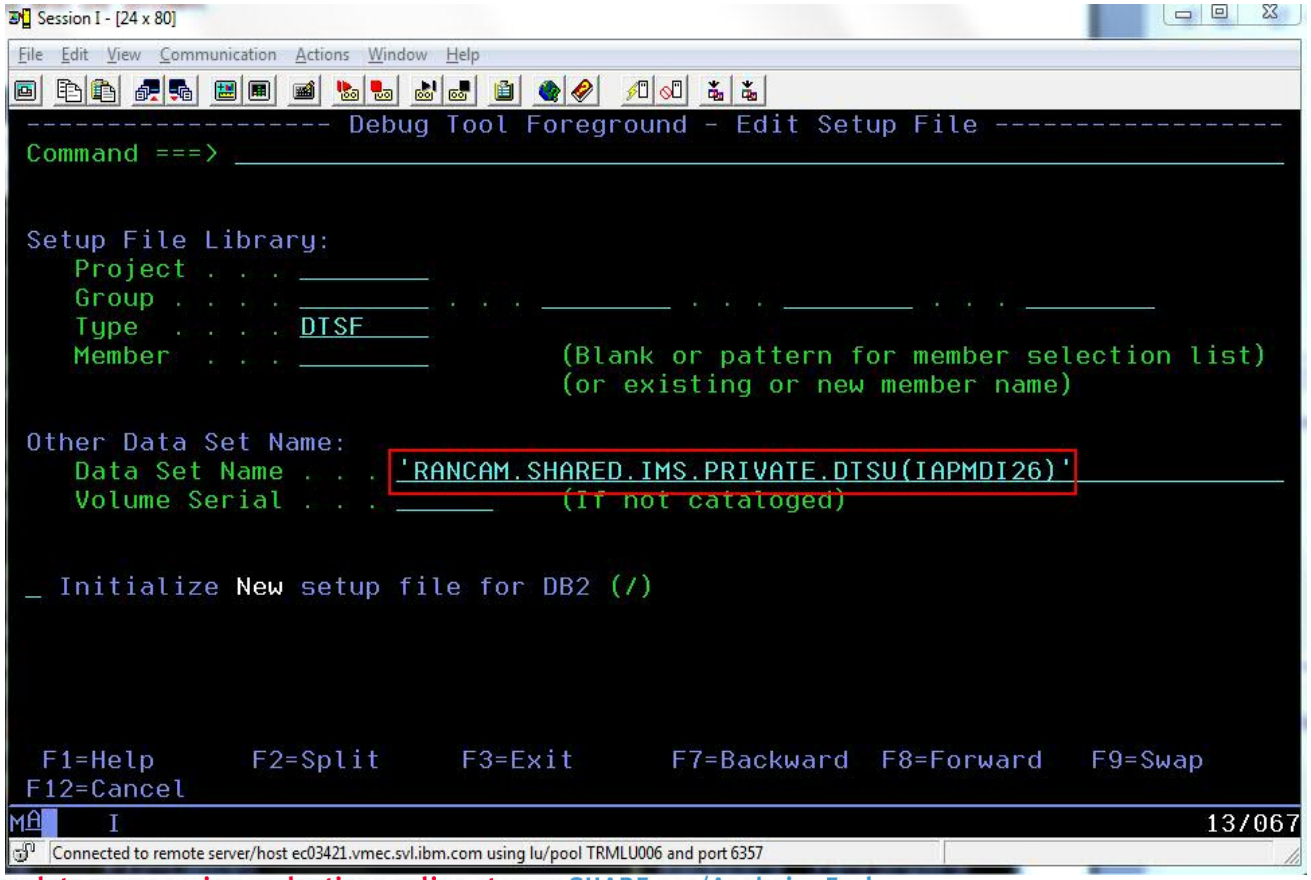
SAVE AS

- After testing, the user can enter SAVE AS or simply press F3 to exit. If the changes are not saved, a prompt will be shown.



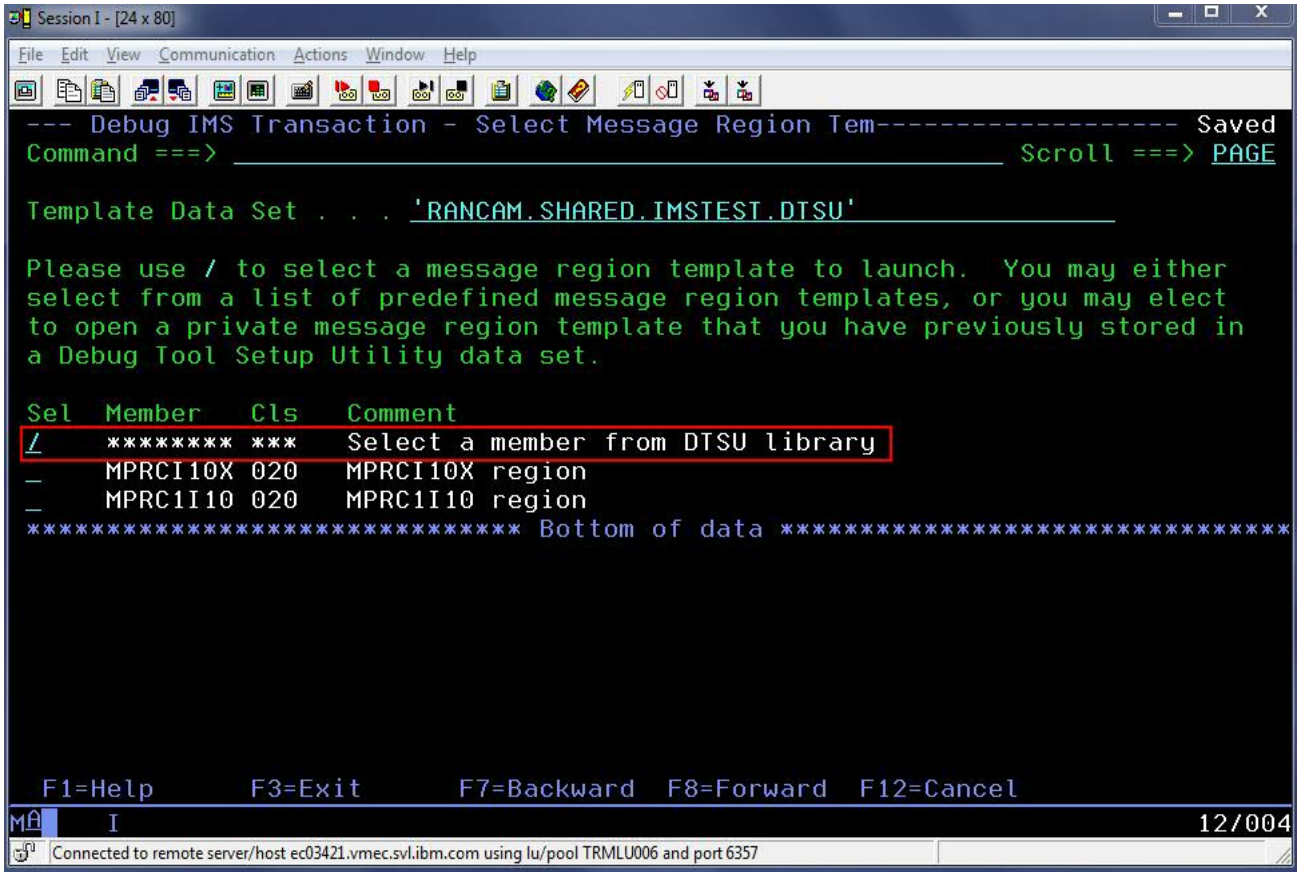
SAVE AS Step 2

- The user can type the name of a DTSU data set and a member name to store the private template. If a member is not specified, a member list will be shown.



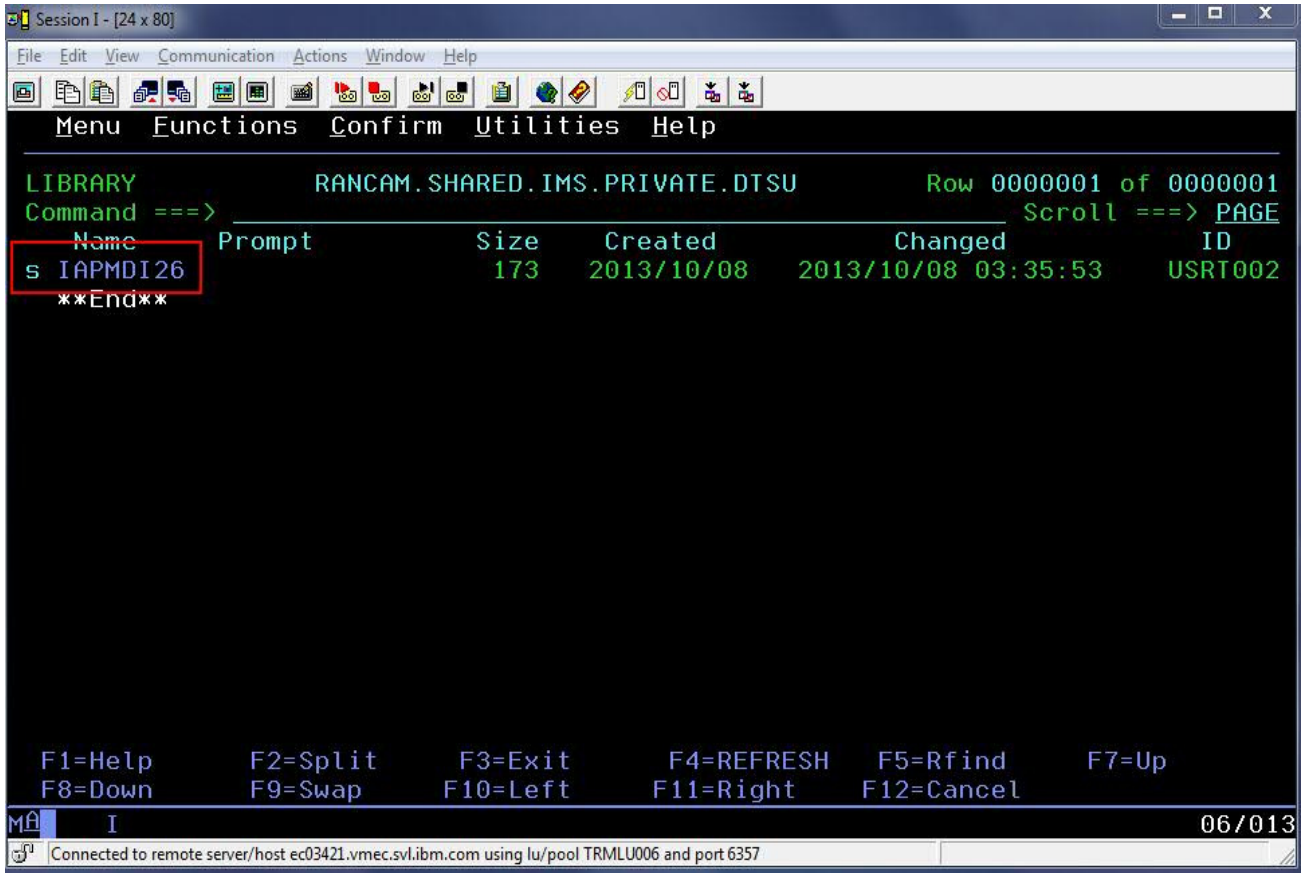
Using a saved test case

- To use the template later, the user will choose “Select a member from DTSU library” and press Enter. This will allow the user to browse a DTSU data set and choose a specific member.



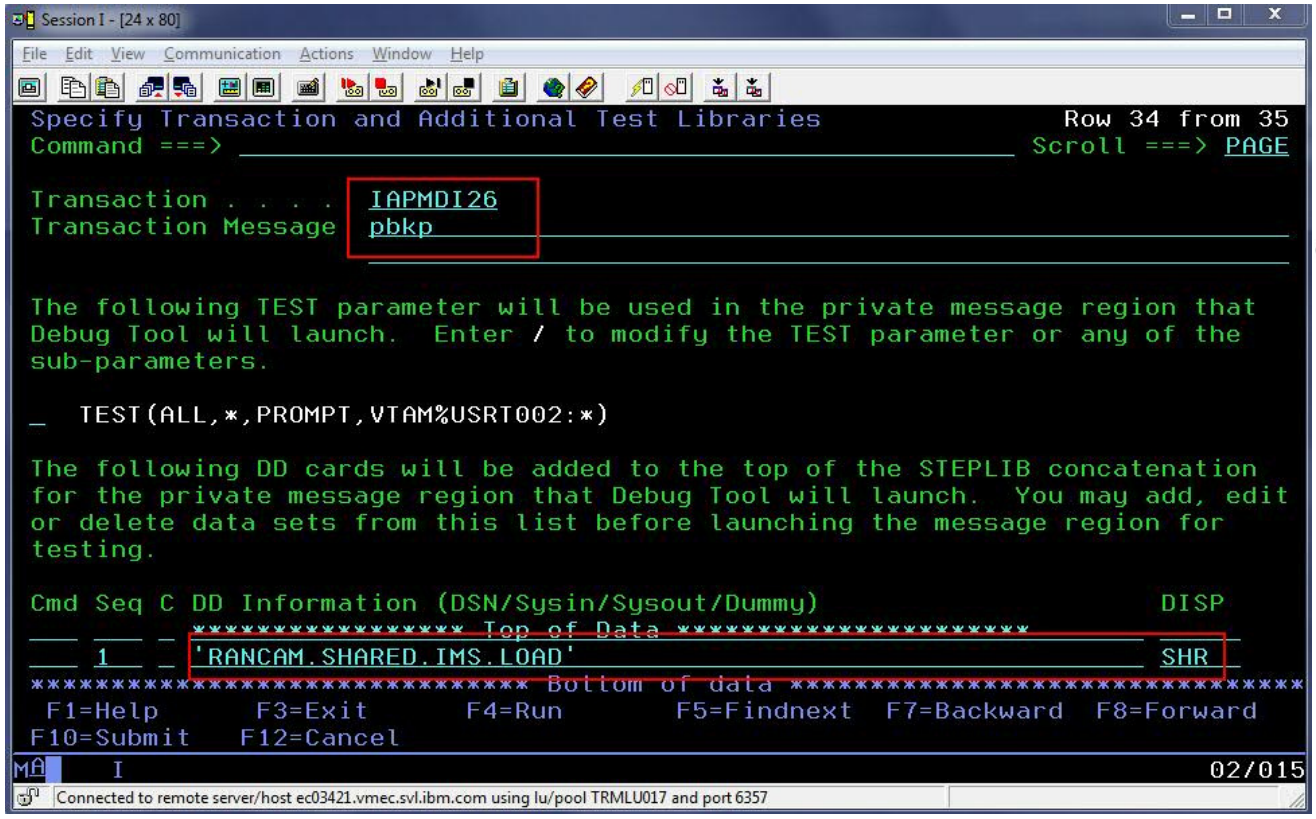
Select test case member

- The user selects the member to use for testing.



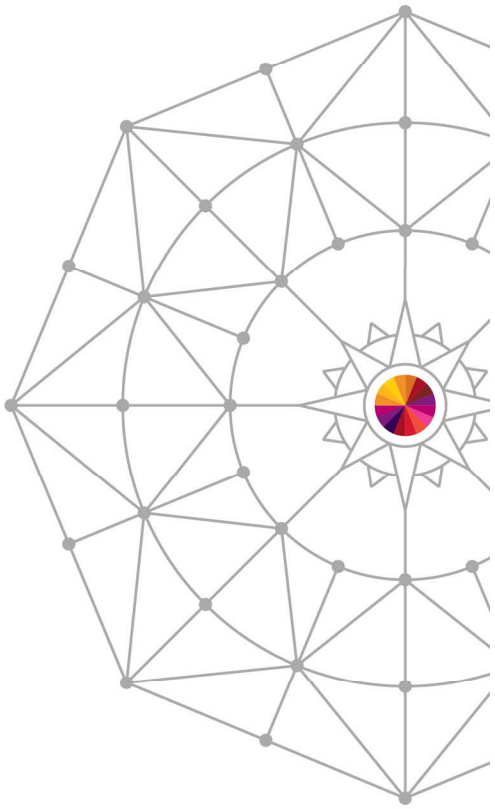
Running the saved test case

- The saved template will be shown. Note that the Transaction, Transaction Message and additional libraries have been saved.



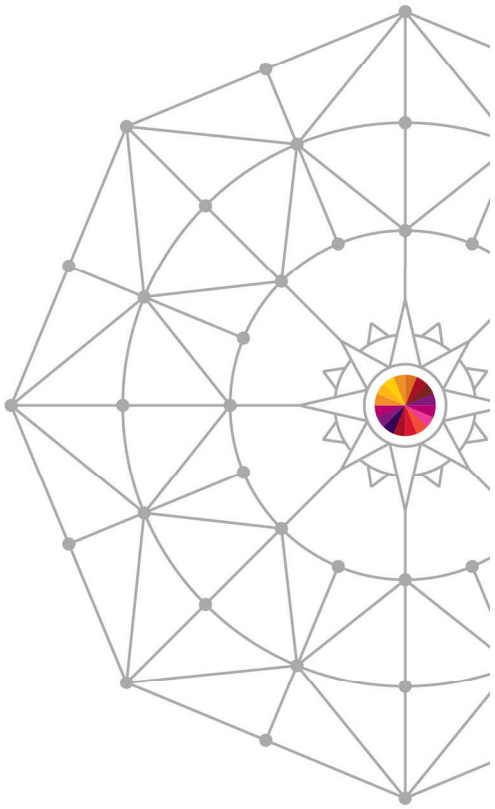
Future Enhancements

- Add support for non-LE transactions
- Add support for conversational transactions
- Allow user to customize behavior of EQANBSWT (e.g., only launch regions & assign transaction, but don't send message).
- Create an Eclipse plugin with same functionality as ISPF panels.



Two-Column Slide (Type Size=28)

- Topic A (Type Size=24)
  - Subtopic 1 (Type Size=22)
  - Subtopic 2 (Type Size=22)
  - Subtopic 3 (Type Size=22)
  - Subtopic 4 (Type Size=22)
- Topic B (Type Size=24)



Slide with Table