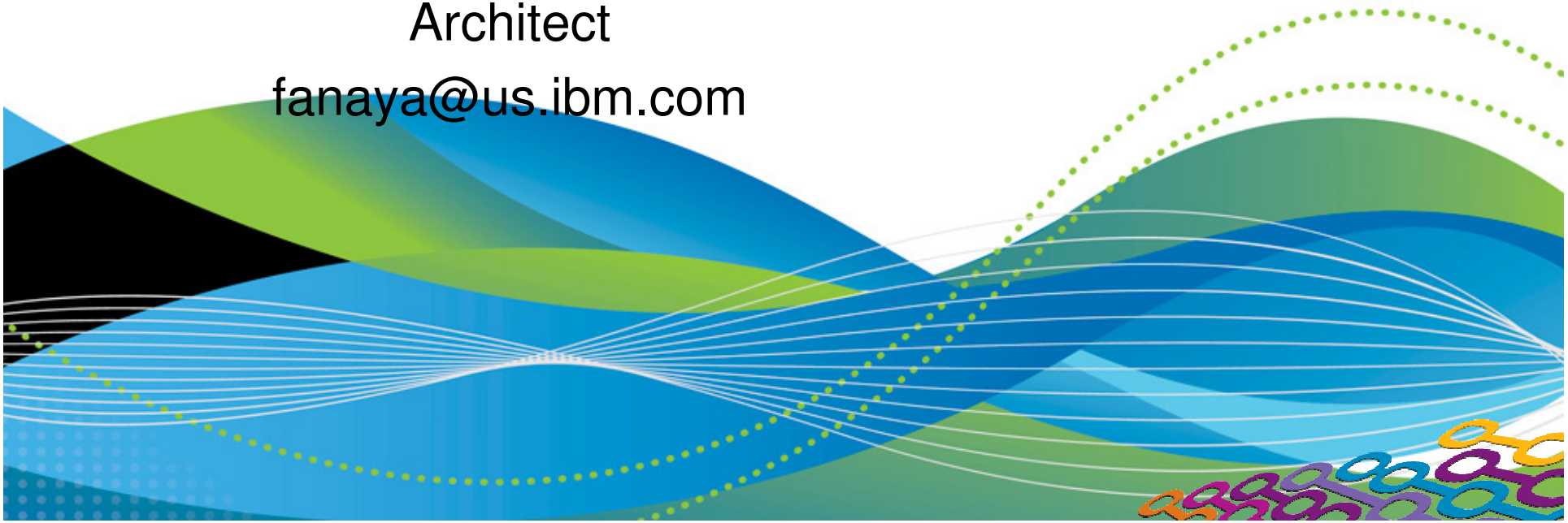
#SHAREorg

10758: IBM Problem Determination Tools for a Smarter Development Environment

Tools for Building Traditional and Composite Applications

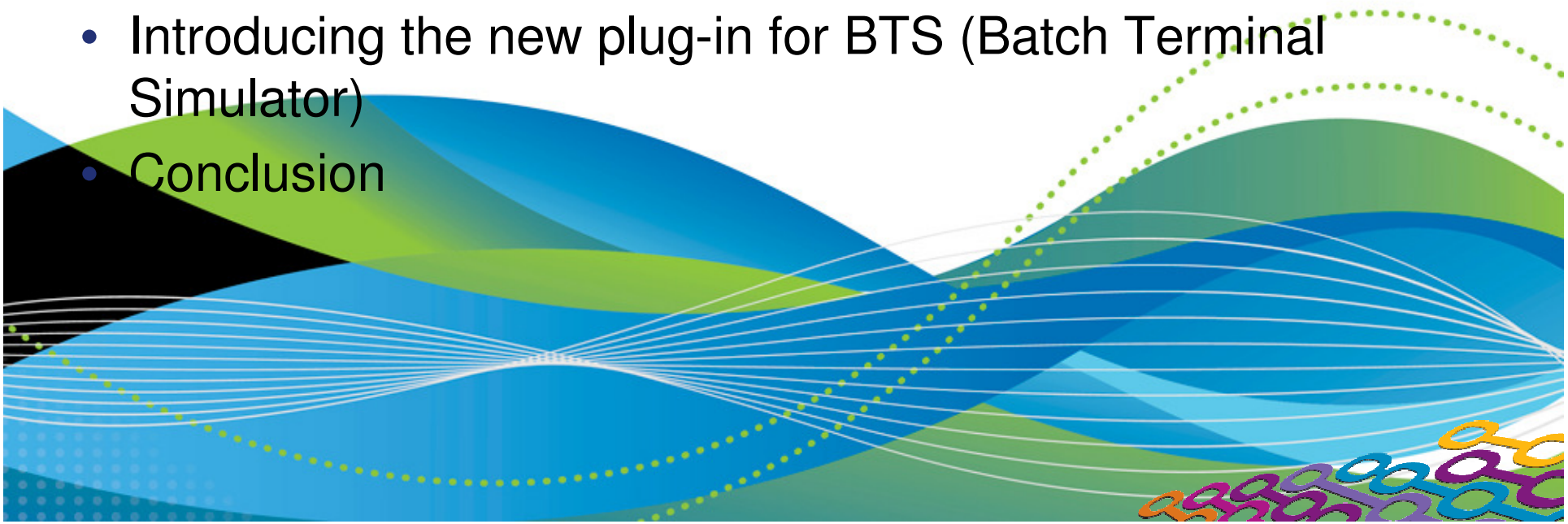
Francisco Anaya, zSeries PD Tools Architect

fanaya@us.ibm.com



Agenda

- Development challenges
- Introduction to the IBM Problem Determination Tools
- The new GUI face of the IBM PD Tools
  - Eclipse plug-ins with RDz or CICS Explorer
- Introduction to Rational Developer for System z
- Introducing the new plug-in for BTS (Batch Terminal Simulator)
- Conclusion



Business drivers for development teams

- **Do more with less**
  - Pressure to reduce or contain costs, and that means:
    - leverage existing investments and infrastructure
    - reduce staff or cut back on other expense categories
  - But NO letting up on incoming requests

- **Velocity**
  - Market dynamics are changing - and that means:
    - delivering finished results more quickly, *while*
    - providing "more modern" devices and systems

- **Service Quality**
  - Keep key systems up and running, *and*
  - Meet new demands, requirements, and regulations

Challenges facing development teams

Complexity

- Systems –
- Software –
- Process -

Time

TTM - TTR

- **Increased complexity**
  - More devices, richer user interaction, more integration
  - Requirements for more streamlined processes, while
  - Meeting increased governance requirements
- **Reduced time**
  - Time can impact the business in terms of loss of revenue for:
    - Meeting Time to Market demands for new functionality
    - Meeting Time to Repair situations with mission critical systems

Challenges facing development teams

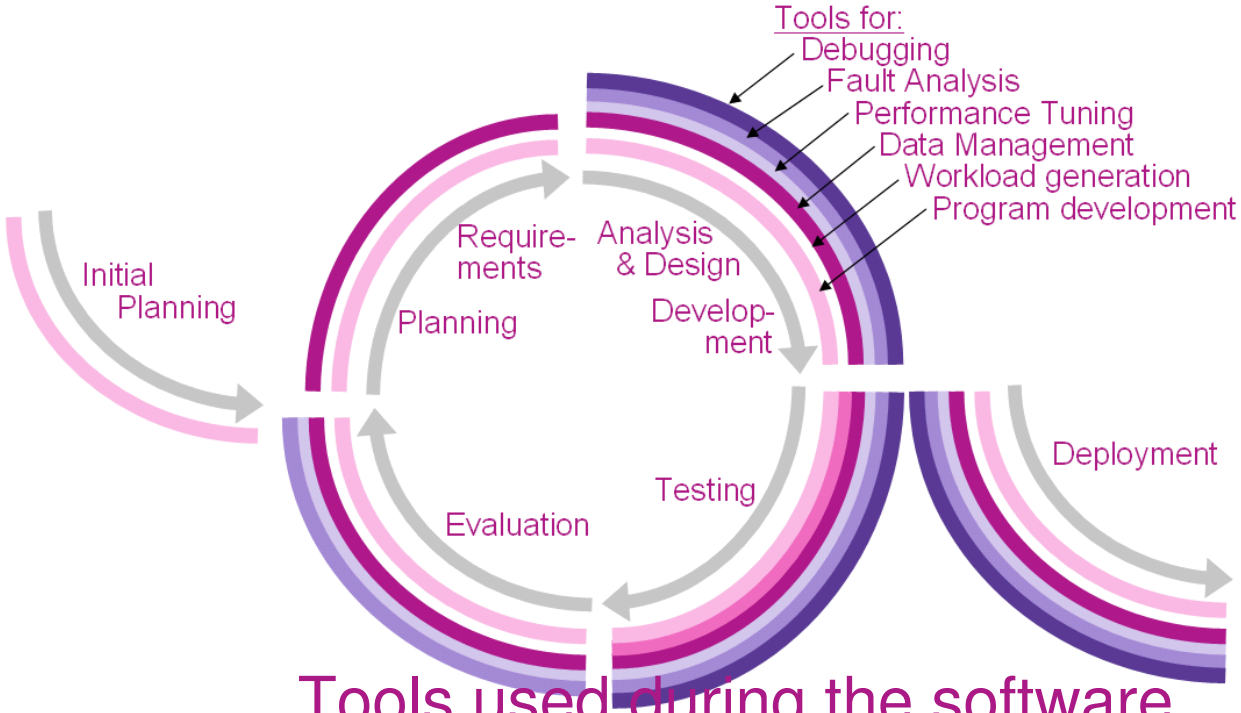
Resources

Skills - Tools

- **Resources**
  - Do your developers have the right skills not only develop new systems but also keep the existing systems running?
  - Do they have the tools they need?
    - To work with the broad range of technologies deployed
    - To quickly work with multiple systems in a Sysplex / CICSplex?
  - Are different teams using different tools?
    - Do different tools create integration issues or even gaps in the development process?

Development processes can only be as good as development tools

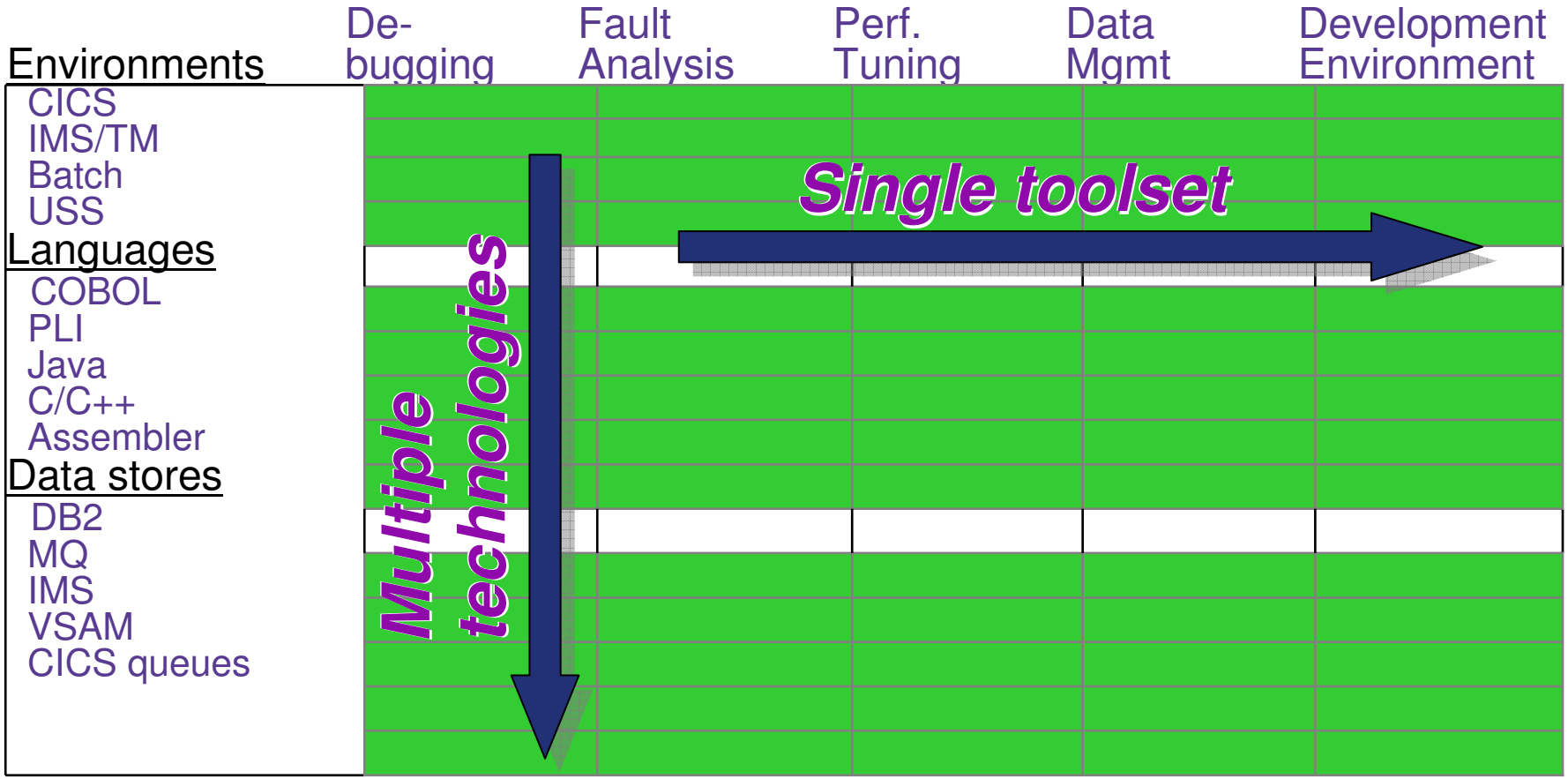
- The right tools optimize productivity, and minimize time and cost
- Processes are impeded by inadequate tools that are not integrated, or do not support your technology spectrum



Tools used during the software development lifecycle

The IBM zOS development toolset approach

- A single integrated toolset with support across a broad zOS technology spectrum



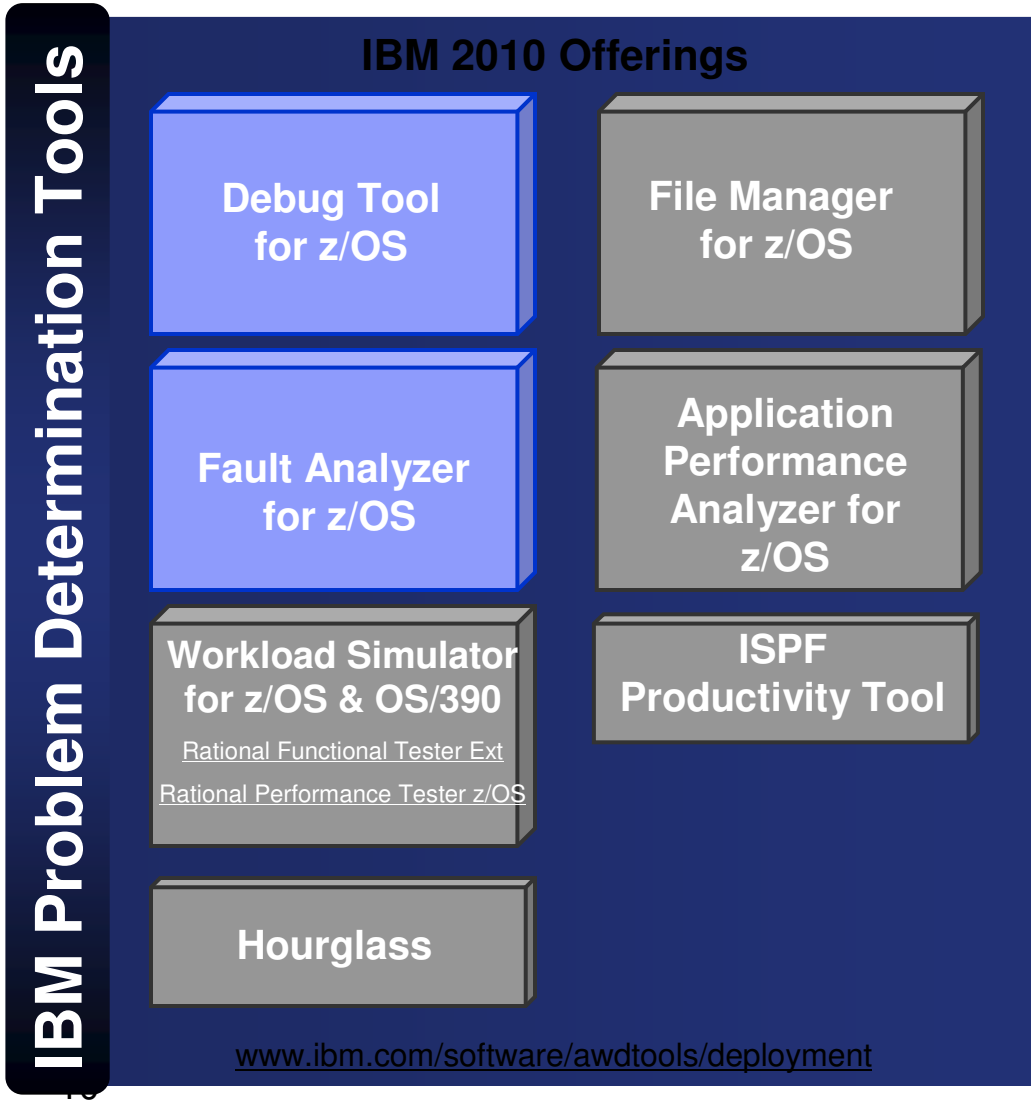
Agenda

- Development challenges
- Introduction to the IBM Problem Determination Tools
- The new GUI face of the IBM PD Tools
  - Eclipse plug-ins with RDz or CICS Explorer
- Introduction to Rational Developer for System z
- Demonstrations of the Problem Determination tools
- Conclusion

What are the IBM zSeries problem determination tools?

- **A suite of products that provide essential capabilities needed by zOS development, testing, and operations support teams**
  - Interactive program debugging
  - Program abnormal termination failure analysis
  - File and database management for test and production
  - Application performance monitoring
  - Testing support

The IBM Problem Determination Tools Suite for z/OS



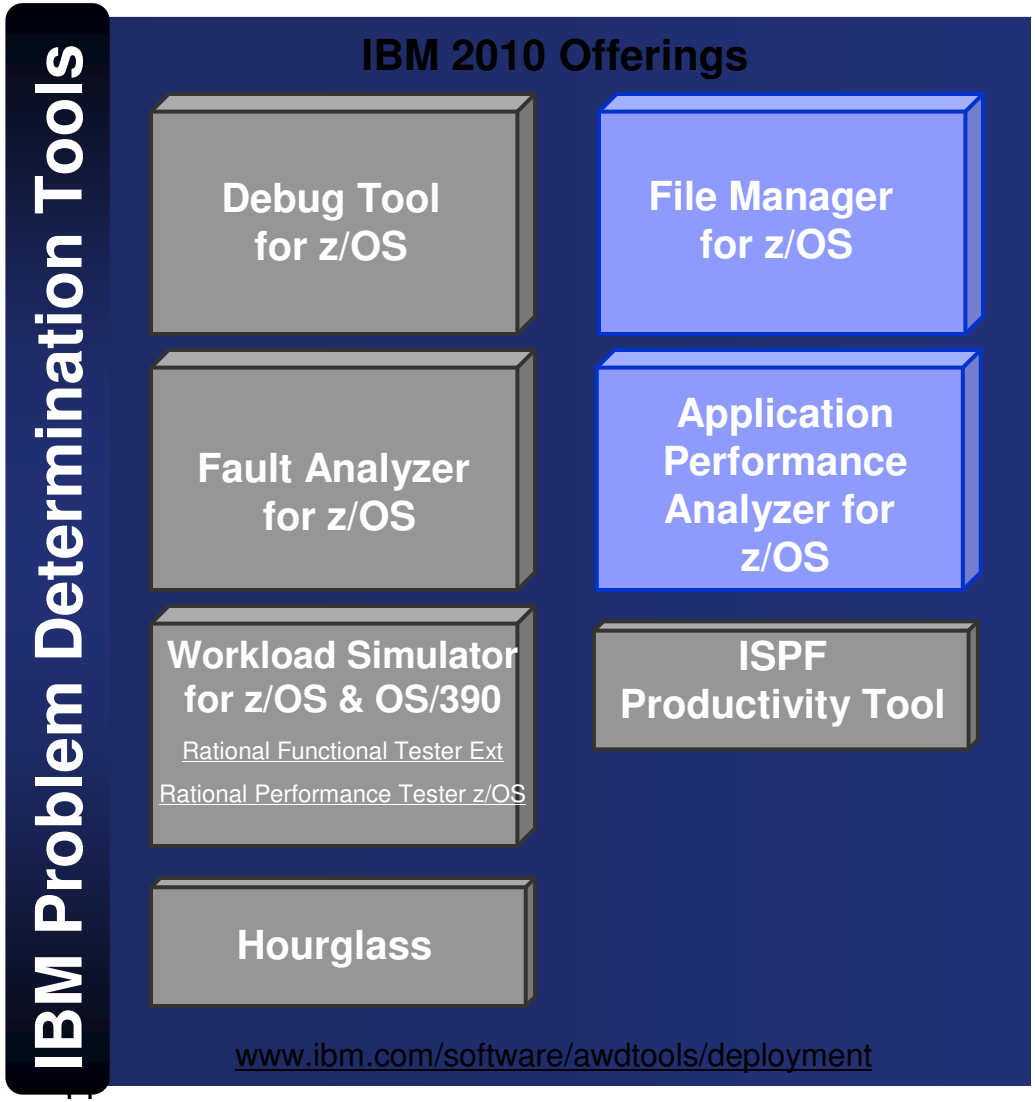
Debug Tool for zOS

- Interactive program debugging
- Multiple languages and zOS environments
- Code coverage reporting
- COBOL modernization

Fault Analyzer for zOS

- Automatic program abend capture and reporting
- Program source-level reporting
- Multiple languages and zOS environments

The IBM Problem Determination Tools Suite for z/OS



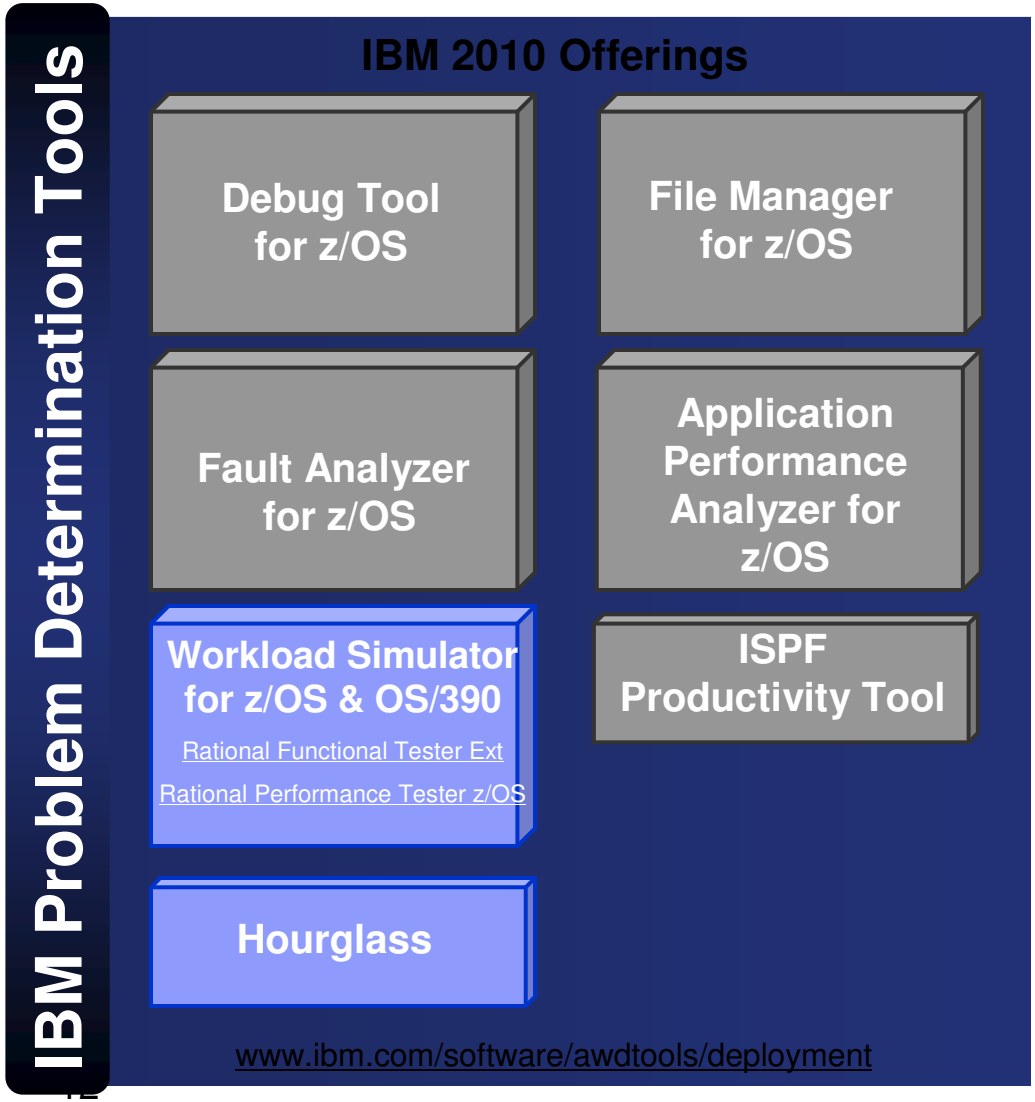
File Manager for zOS

- Edit and view files and databases of any size
- VSAM, sequential, and PDS(e) files, DB2 and IMS databases
- Extensive file and data utilities

Application Performance Analyzer for zOS

- Monitor and report application performance
- Source level reporting
- Multiple languages and zOS environments

The IBM Problem Determination Tools Suite for z/OS



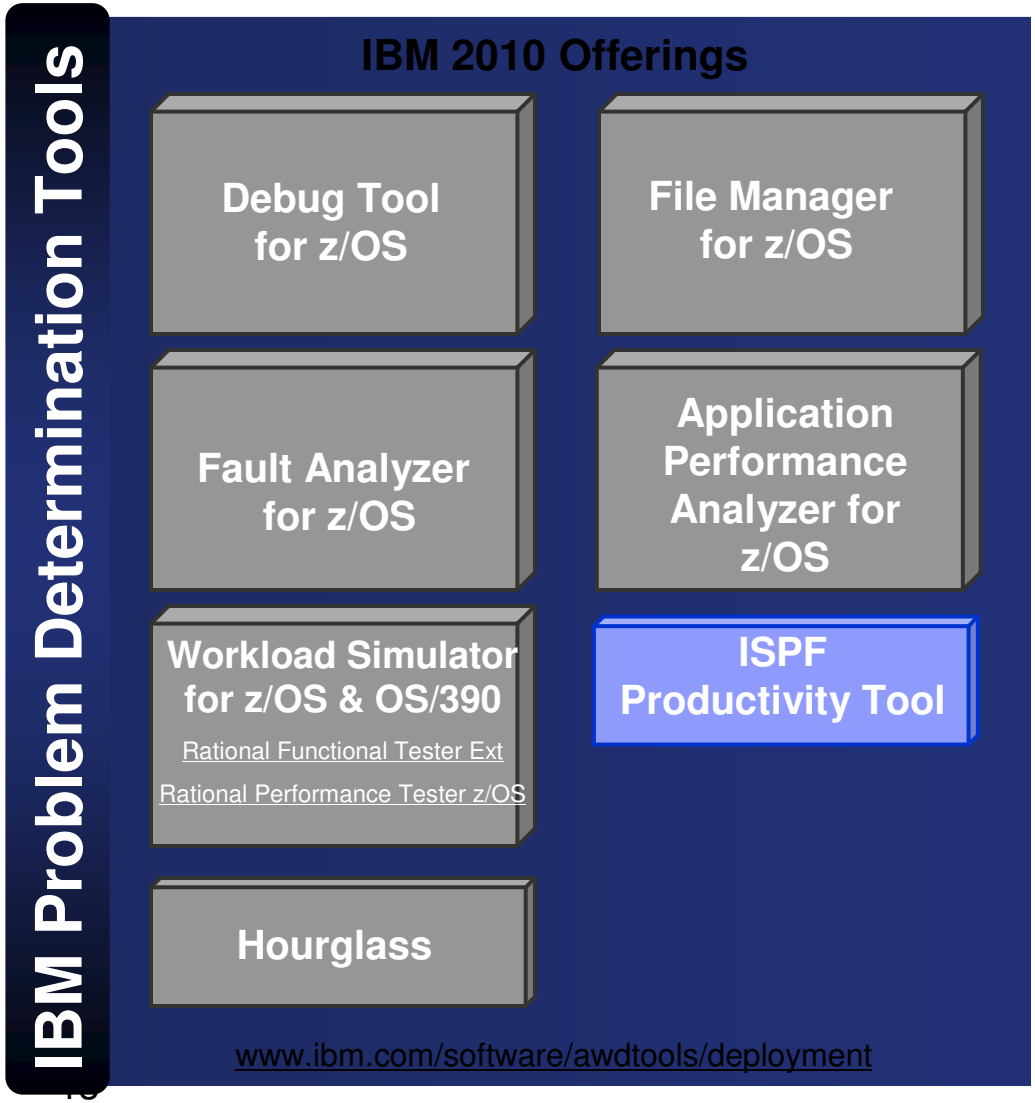
Workload Simulator for zOS and OS/390

- Drive regression, performance, stress, function, and capacity testing
- Simulate on-line users with smart scripts

Hourglass

- Alter the date and time returned to an application when a time request is made

The IBM Problem Determination Tools Suite for z/OS

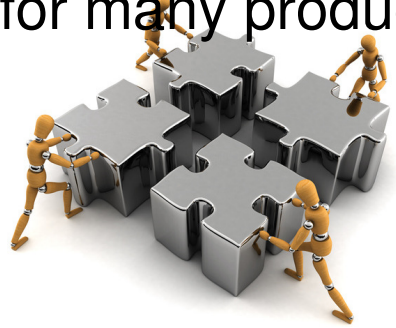


ISPF Productivity Tool

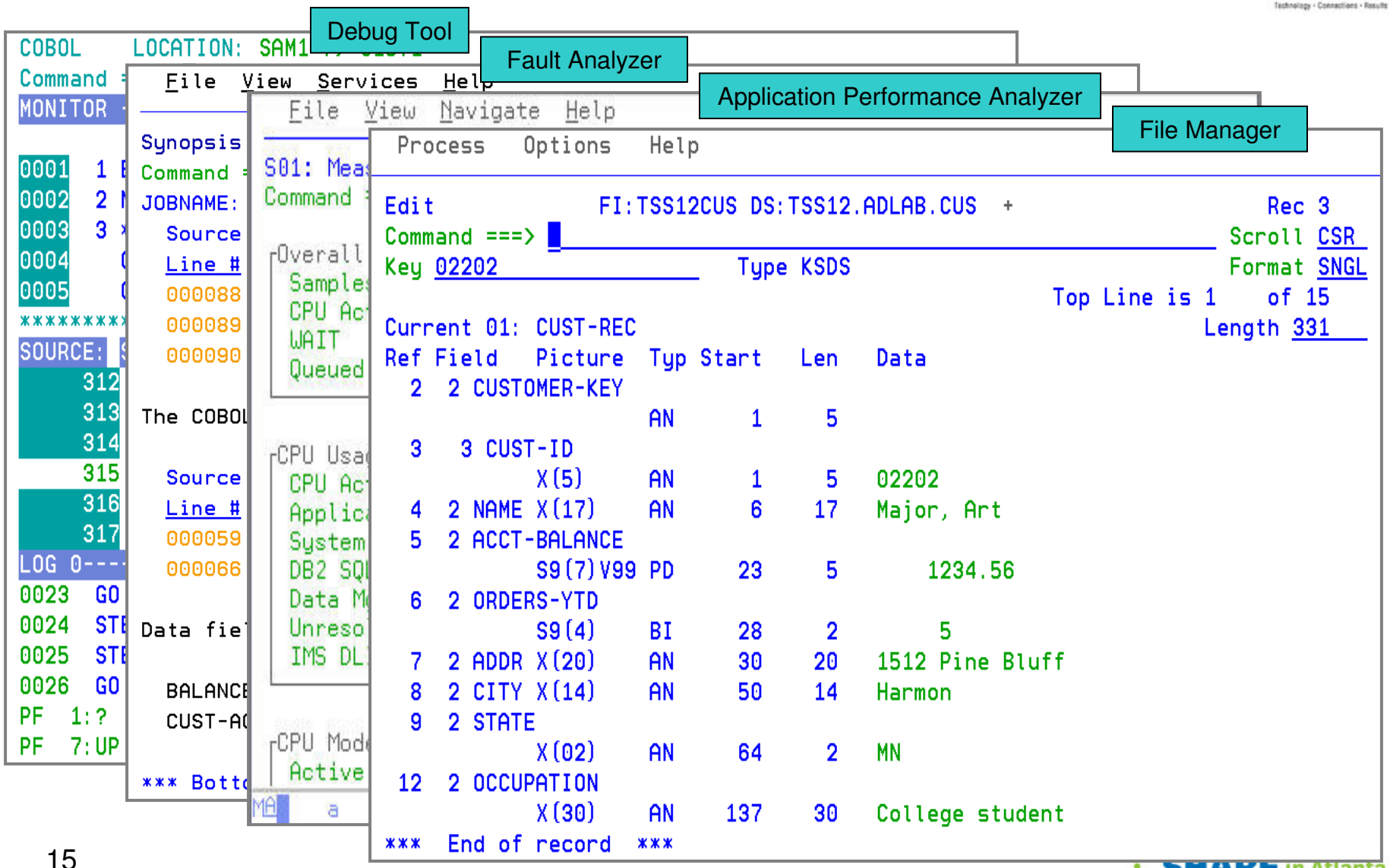
- Turbo-charge the productivity of your ISPF users
- Turn IBM ISPF into a centralized, object-oriented development center

Support a variety of application developer skills

- Reusing mainframe application assets requires expertise in both traditional and new zOS technologies
- To maximize productivity:
  - Some developers may do best with traditional interfaces
  - Others may be more productive with GUI interfaces
- **The IBM Problem Determination Tools provide the best of both worlds**:
  - Proven traditional 3270-based interfaces
  - Eclipse-based GUI interfaces for many products



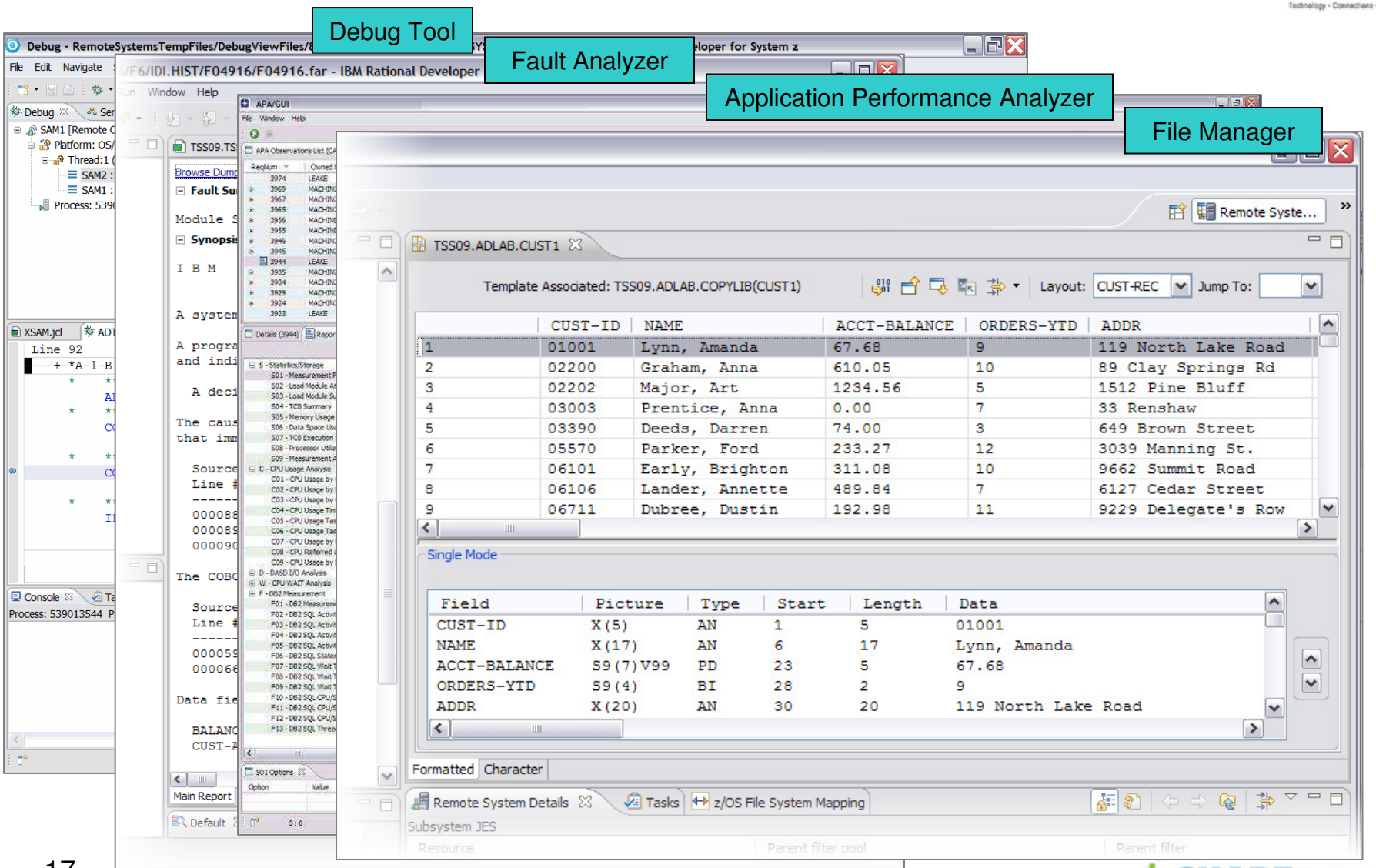
The tools provide excellent, traditional 3270-based interfaces



Agenda

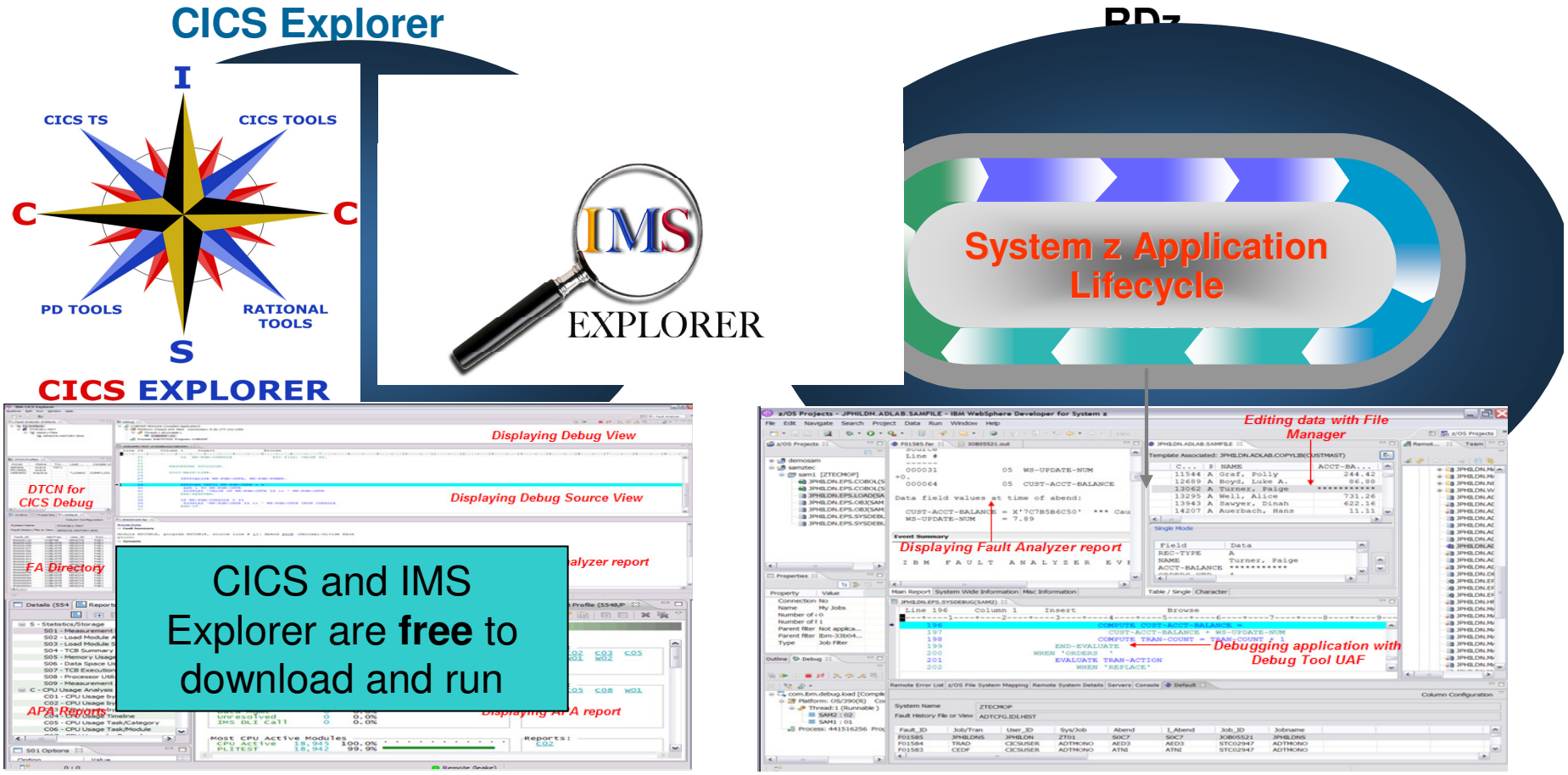
- Development challenges
- Introduction to the IBM Problem Determination Tools
- The new GUI face of the IBM PD Tools
  - Eclipse plug-ins with RDz or CICS Explorer or IMS Explorer
- Introduction to Rational Developer for System z
- Demonstrations of the Problem Determination tools
- Conclusion

In addition, many of the tools provide GUI interfaces



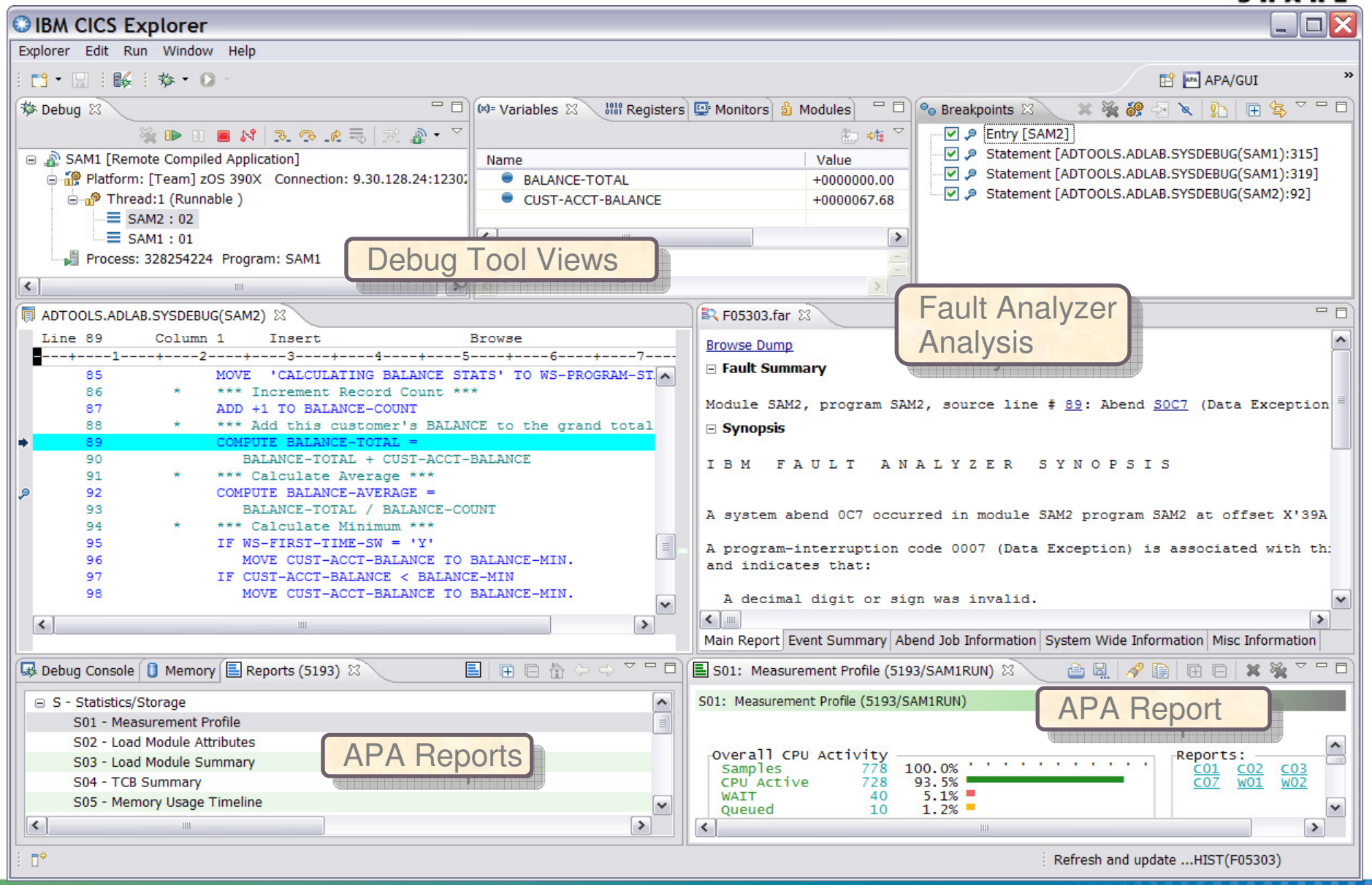
Eclipse plug-ins for the IBM Problem Determination Tools

- Graphical interfaces are provided as plug-ins for Eclipse platforms, such as Rational Developer for System z, CICS Explorer, and IMS Explorer



Debug Tool, Fault Analyzer, Application Performance Analyzer, and File Manager plug-ins

Problem Determination Tools plug-ins in CICS Explorer



IBM

New!

IBM CICS Explorer™

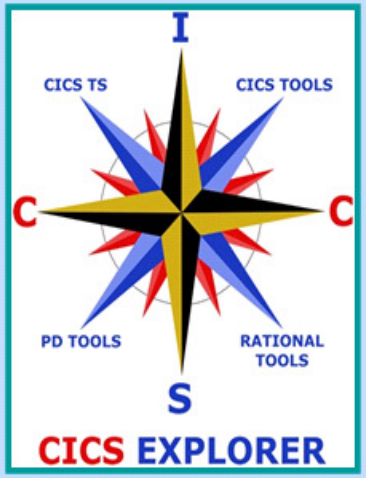
Configuration
Status
Control, Test
MQ

Status
Situations
Topology
XE

VSAM and PDS
File Management
Access Ability
FM

Threadsafe, File, CPU,
Response Time analysis
Graphical and
Sheet views
PA

Daemon &
Connection
Status & Test
TG



Develop
Test
Etc
RDz

CICS, IMS, DB2,
& z/OS
Abend Reporting
& Diagnosis
FA

Execution Tree
Dependencies
Queries
Command Flow
IA

Deployment
Discovery,
Visualization,
Automation & Control
DA

CICS, IMS,
DB2, & z/OS
Application
Debugging
DT

CICS, IMS, DB2,
& z/OS
Observation Requests
& Reporting
APA

CRUD/Install
History, Audit
Backout
Search, Compare
CM

CRUD/Install
Control, Filter
Topology
Events, ATOM
SM

ibm.com/cics/tools
ibm.com/cics/explorer
ibm.com/cics/explorer/download

SM	CICS Transaction Server
IA	CICS Interdependency Analyzer
PA	CICS Performance Analyzer
CM	CICS Configuration Manager
DA	CICS Deployment Assistant
TG	CICS Transaction Gateway

CICS TS

APA FA DT FM	Application Performance Analyzer Fault Analyzer Debug Tool File Manager
MQ	WebSphere MQ
XE	OMEGAMON XE for CICS
RDz	Rational Developer for System z


Address http://www-01.ibm.com/software/htp/cics/explorer/
Go
Links IBM Business Transformation Homepage IBM Standard Software Installer IT Help Central Join World Community Grid
Snagit
IBM
United States [change]
Search
Home Solutions Services Products Support & downloads My IBM
Welcome [IBM Sign in] [Register]
Software > Host Transaction Processing > CICS family > CICS Transaction Server >
CICS Explorer
Features and benefits
System requirements
Library
News
Related links
Warranty info
Product Accessibility Information
CICS Explorer
The new face of CICS
Continuous Innovation Creates Success
CICS EXPLORER
We're here to help
Easy ways to get the answers you need.
Request a quote
E-mail IBM
Or call us at:
877-426-3774
Priority code:
109HH03W
Overview
IBM CICS Explorer - New Face of CICS. Integration point for CICS tooling with rich CICS views, data, methods.
Looking for new ways to accelerate the transfer of knowledge, skills, and best practice to the next generation of technical staff/experts? Need to maintain productivity and protect service-levels? CICS Explorer and System z lead the way to platform simplification.
Common, intuitive, Eclipse-based environment for architects, developers, administrators, system programmers, and operators
Learn more
Features & benefits
System requirements
Downloads
Demo (ZIP, 31.4MB)
CICS Explorer
Podcast series

Agenda

- Development challenges
- Introduction to the IBM Problem Determination Tools
- The new GUI face of the IBM PD Tools
  - Eclipse plug-ins with RDz or CICS Explorer
- Introduction to Rational Developer for System z
- Demonstrations of the Problem Determination tools
- Conclusion

Rational Developer for System z (RDz)

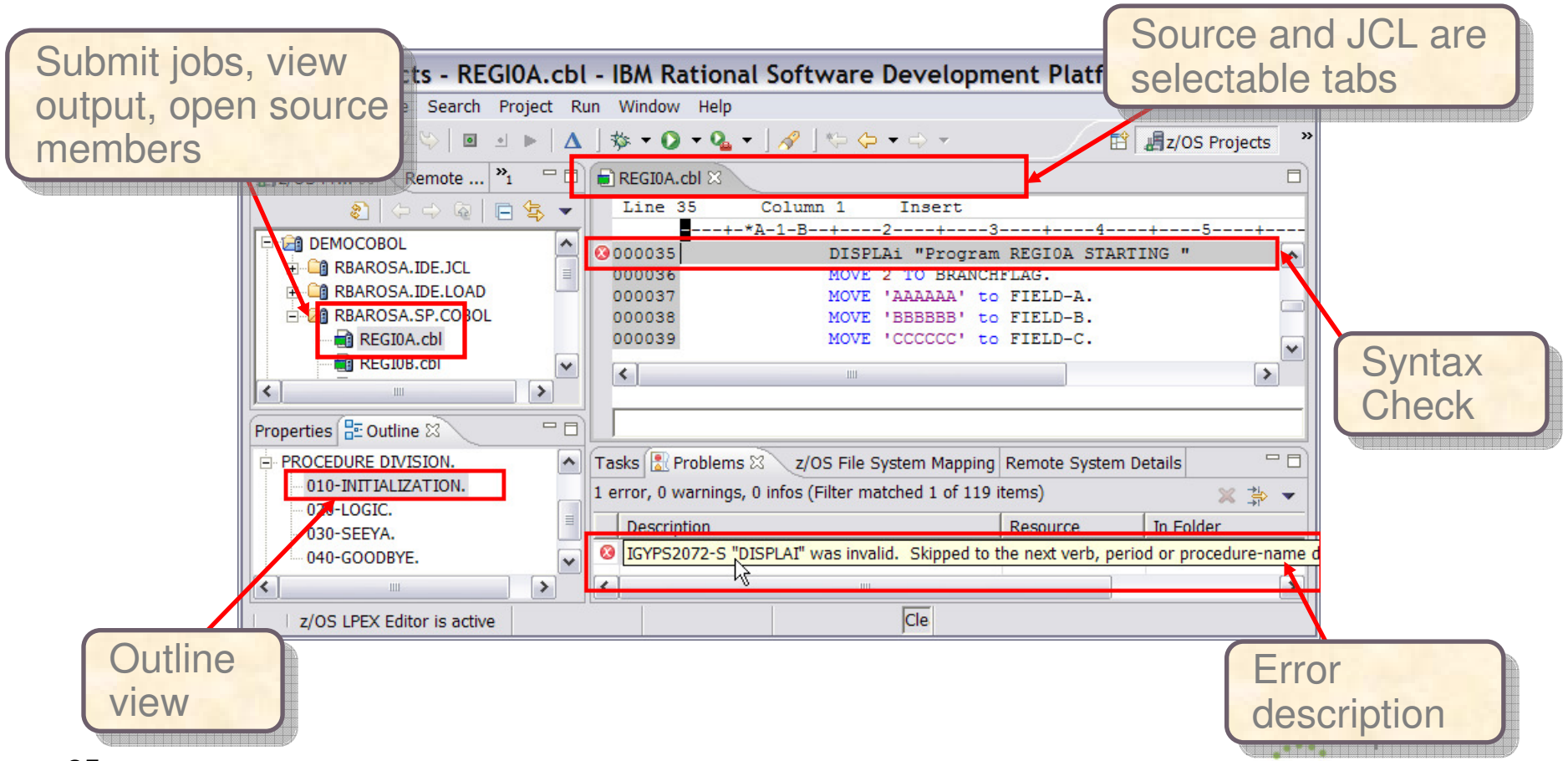
- **An eclipse-based Integrated Development Environment (IDE) for System z applications**
  - Common IDE for COBOL, PL/I, C, C++, HLASM, Java, EGL and web services
- **Supports Enterprise Modernization and SOA**
  - Interactive access to z/OS for development, job generation, submission, monitoring, debugging
  - Enables CICS and IMS applications for web services and SOA
  - Supports for J2EE, JCA, XML, web services

ISPF-based development

- Multiple screens / sessions and varied tools
- Limited screen height and width

RDz-based development

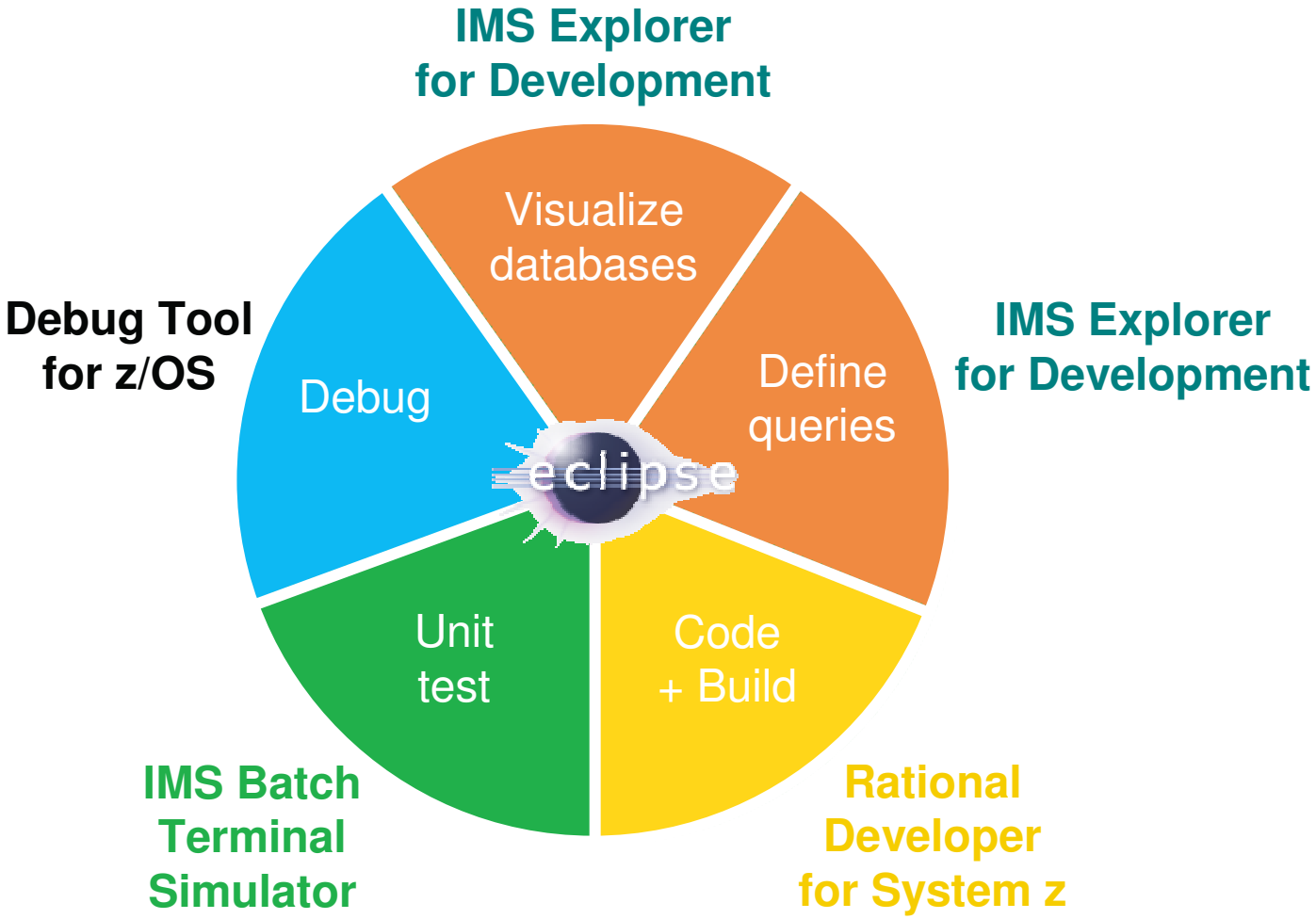
- Simplified development with more information at your fingertips



Agenda

- Development challenges
- Introduction to the IBM Problem Determination Tools
- The new GUI face of the IBM PD Tools
  - Eclipse plug-ins with RDz or CICS Explorer
- Introduction to Rational Developer for System z
- Introducing the new plug-in for BTS (Batch Terminal Simulator) and IMS Explorer
- Conclusion

Integrated Eclipse Environment for IMS Development, Testing, and Debugging



27

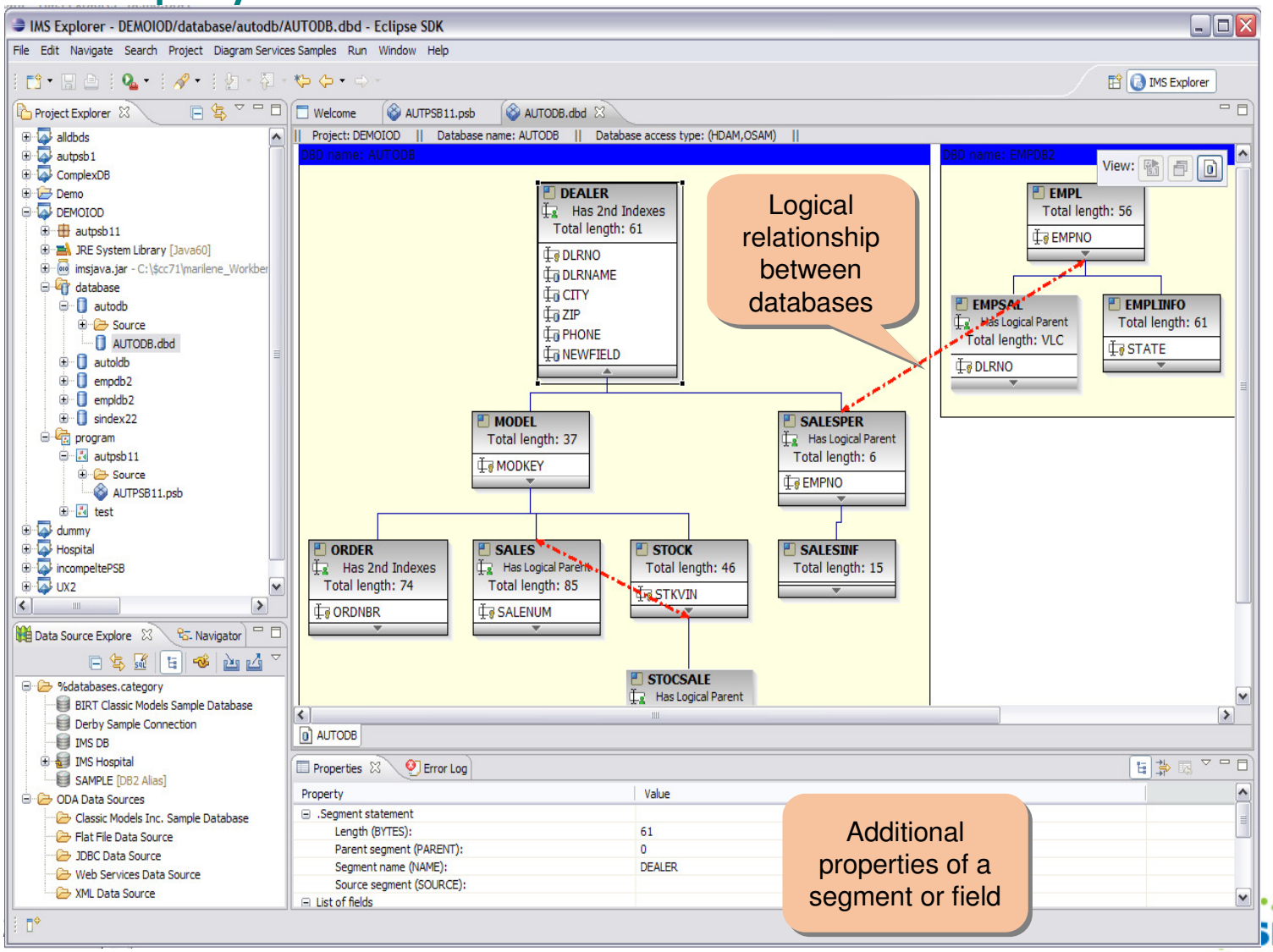
IMS Explorer for Development

- Visualize and change IMS database (DBDs) and program definition sources (PSBs)
  - View IMS database hierarchical structures
  - View, create, and edit PSBs
  - Change/add fields in DBDs
  - Import COBOL copybooks and PL/I structures to a database segment
  - Generate DBD source and PSB source
- Access IMS data using SQL statements
  - Leverage IMS V11 Universal JDBC driver
- Connect to the z/OS system
  - Browse data sets
  - Submit JCL and inspect the output in JES
  - Import and export DBD and PSB source files from a data set to the IMS Explorer, and vice-versa

28

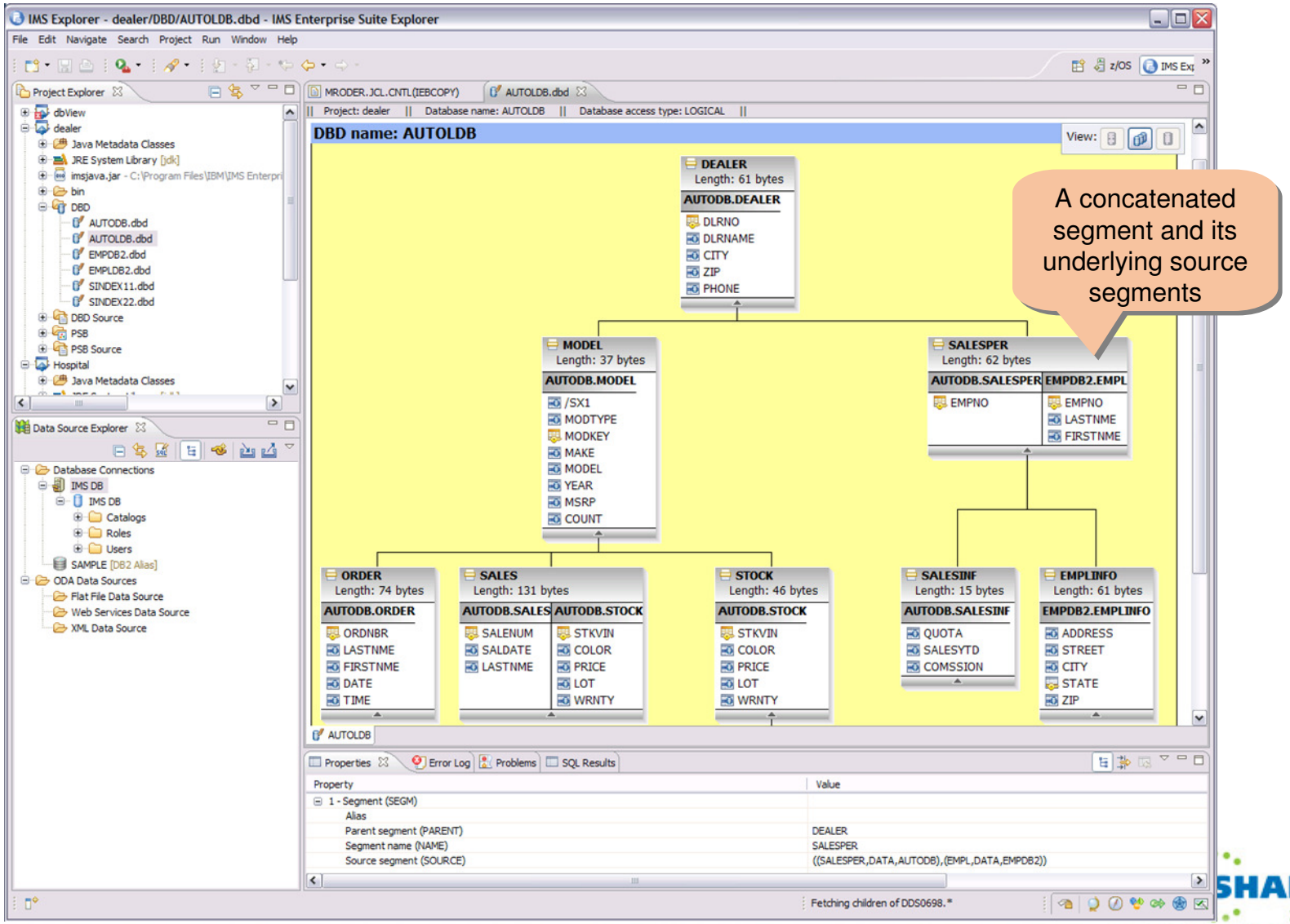
IMS Explorer for Development

View physical IMS database structure



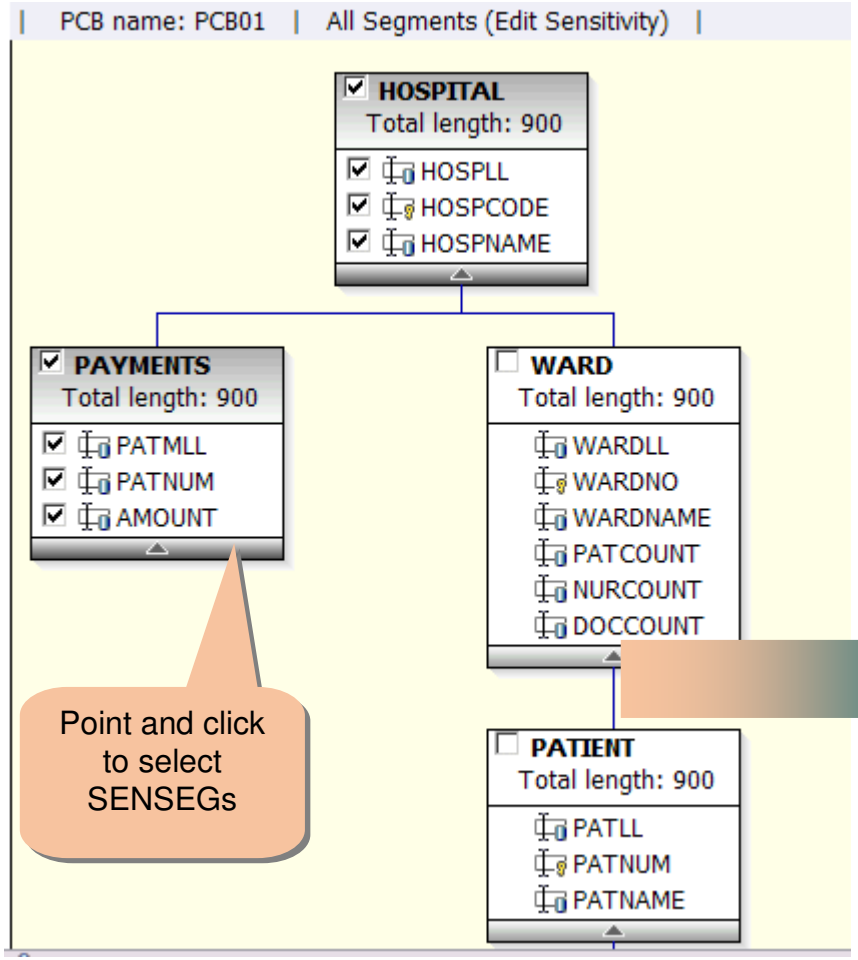
IMS Explorer for Development

View logical IMS database structure



IMS Explorer for Development

Build PCB definition



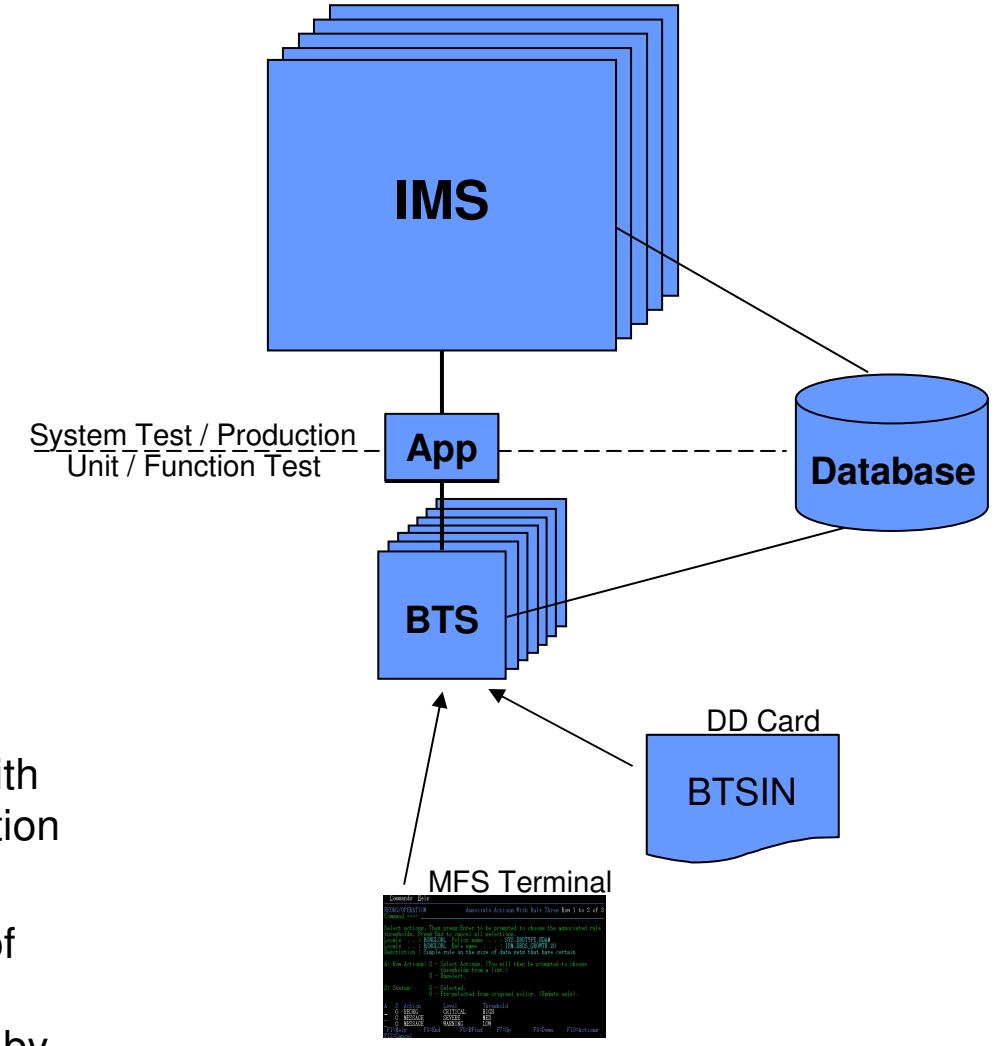

```
*********************************************************
*       PCB NUMBER 5       DB    DEDBJN21
*********************************************************
PCB        TYPE=DB,DBNAME=DEDBJN21,POS=M,PROCOPT=A,KEYLEI
              PCBNAME=PCB01
   SENSEG   NAME=HOSPITAL,PARENT=0
   SENSEG   NAME=PAYMENTS,PARENT=HOSPITA
   SENSEG   NAME=WARD,PARENT=HOSPITAL
   SENSEG   NAME=PATIENT,PARENT=WARD
   SENSEG   NAME=ILLNESS,PARENT=PATIENT
   SENSEG   NAME=TREATMNT,PARENT=ILLNESS
   SENSEG   NAME=DOCTOR,PARENT=TREATMNT
   SENSEG   NAME=BILLING,PARENT=PATIENT
*********************************************************
*         PCB NUMBER 6       DB    IVPDB1
*********************************************************
```

Generated PSB source

31

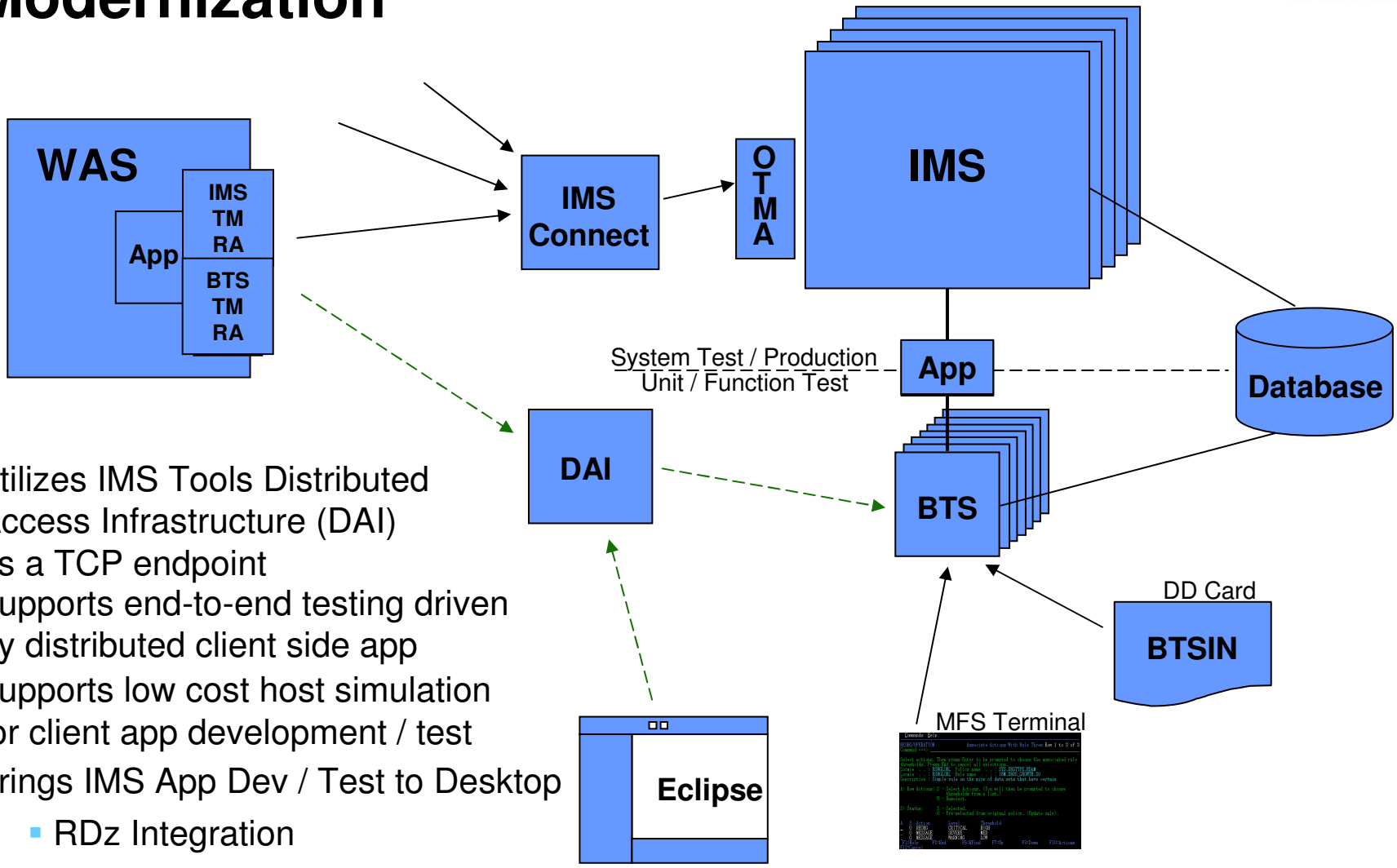
IMS Batch Terminal Simulator Modernization

- Tool for Unit Testing and Debugging IMS applications
  - Runs in TSO, IMS Batch (DL/I DBB) or IMS Online (BMP/JBP)
  - Simulates IMS DC calls
- Driven through BTSIN control statements or interactive full screen TSO simulation
- Playback utility
  - Generates BTSIN with the corresponding MFS associated with the MOD name from IMS transaction log records ('01'X, '03'X).
  - Prints containing screen images of transactions including data, USER/LTERM identification, time by extracting from BTSOUT



32

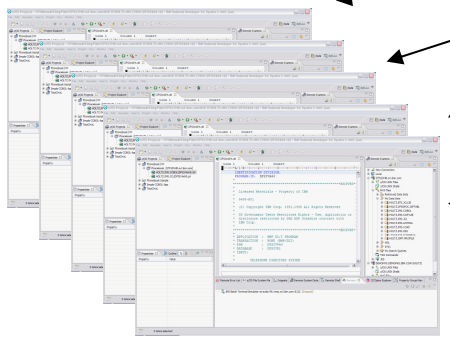
IMS Batch Terminal Simulator Modernization



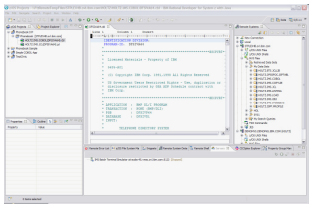
- Utilizes IMS Tools Distributed Access Infrastructure (DAI) as a TCP endpoint
- Supports end-to-end testing driven by distributed client side app
- Supports low cost host simulation for client app development / test
- Brings IMS App Dev / Test to Desktop
  - RDz Integration
  - Debug Tool Integration

33

Application Development Modernization with BTS



XML



1 System Programmer defines common datasets, attributes, etc (similar to PROC) and exports as XML

2 Developers import XML and use to create Launch Configuration… to test applications

DAI

BTS

BTS

BTS

XML

3 Export Launch Configuration as XML for BTSRA

WAS

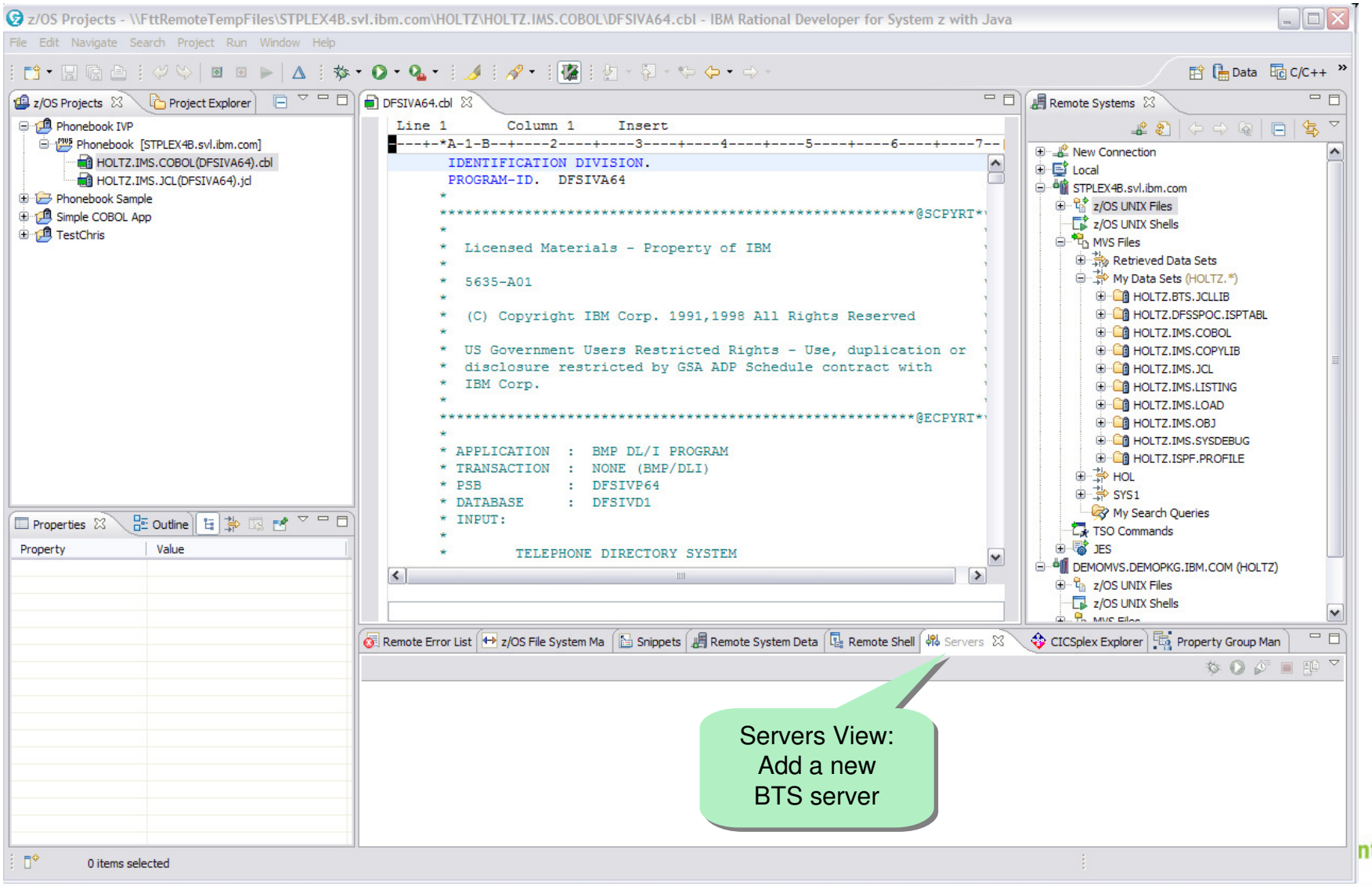
BTS TM RA

App

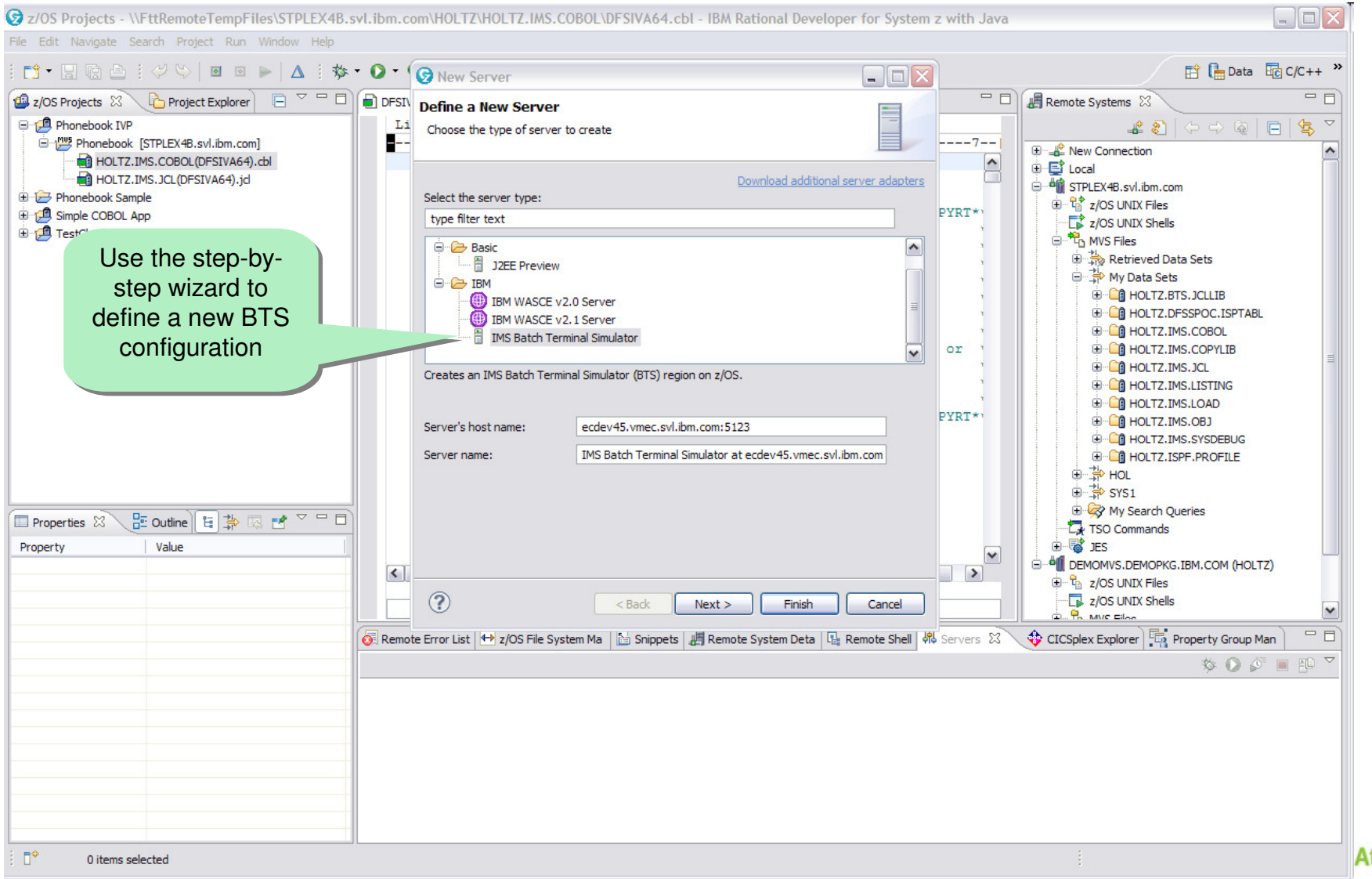
34

IMS Batch Terminal Simulator

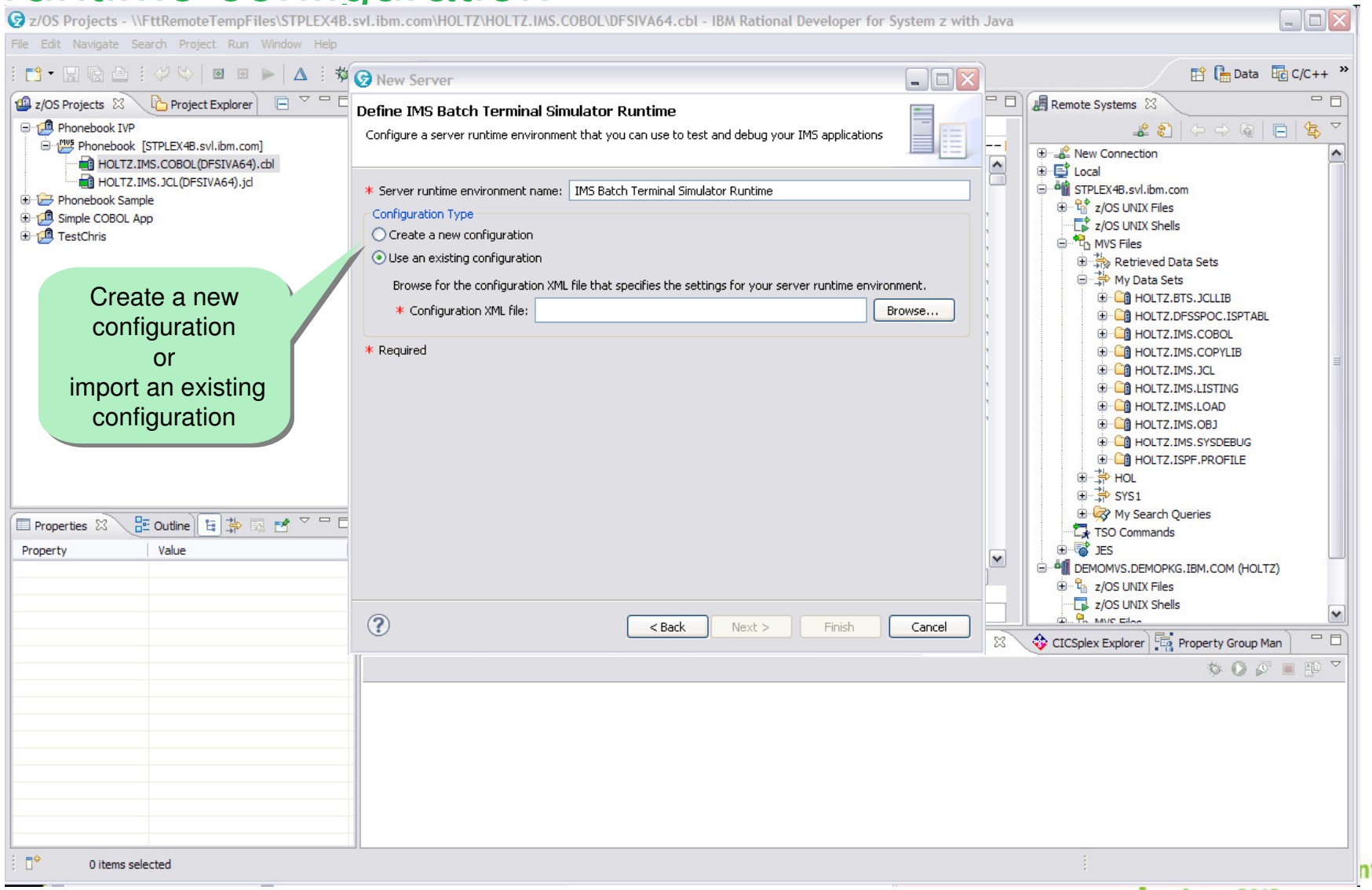
Add BTS server



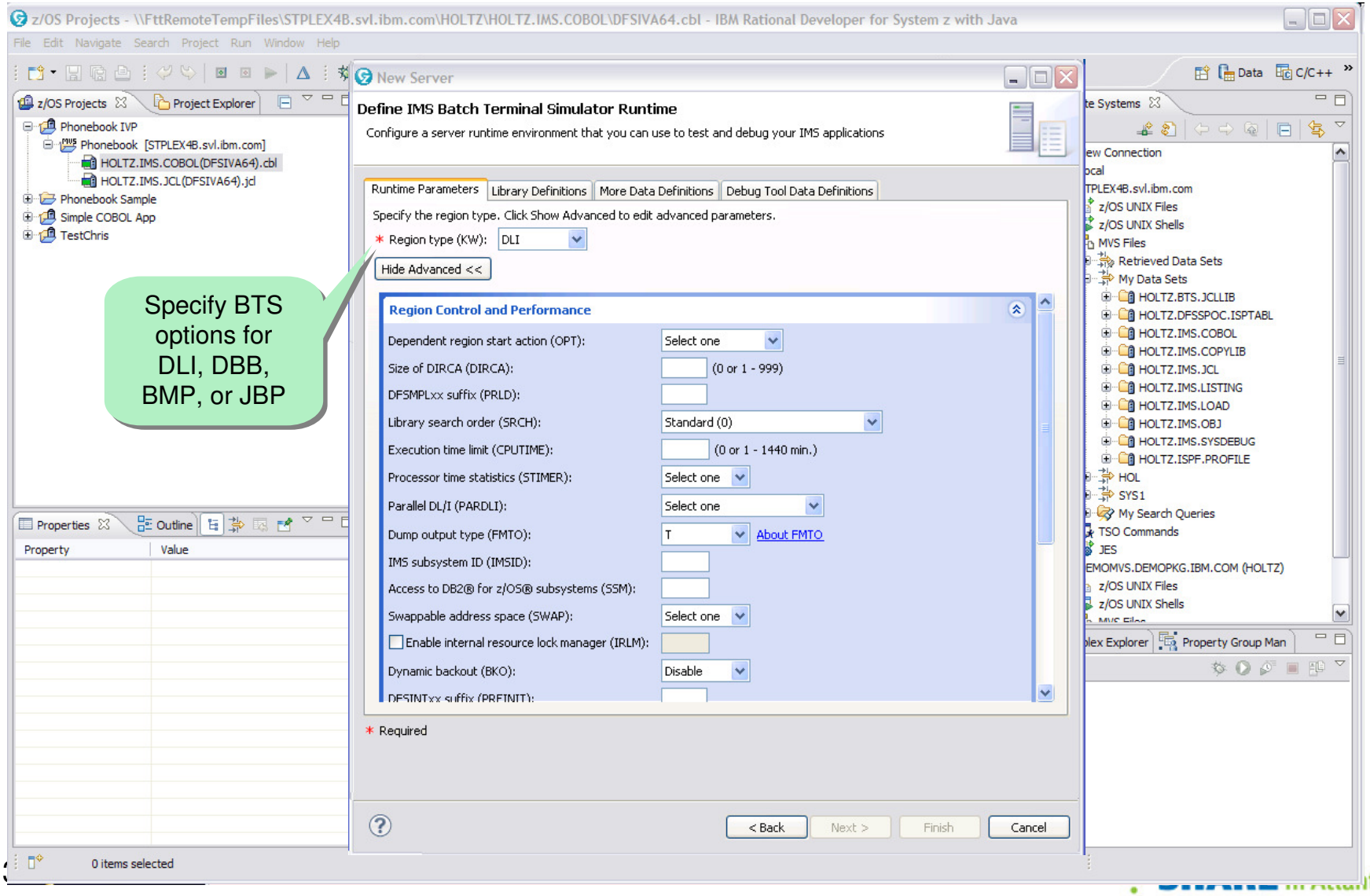
Define BTS server using the step-by-step wizard



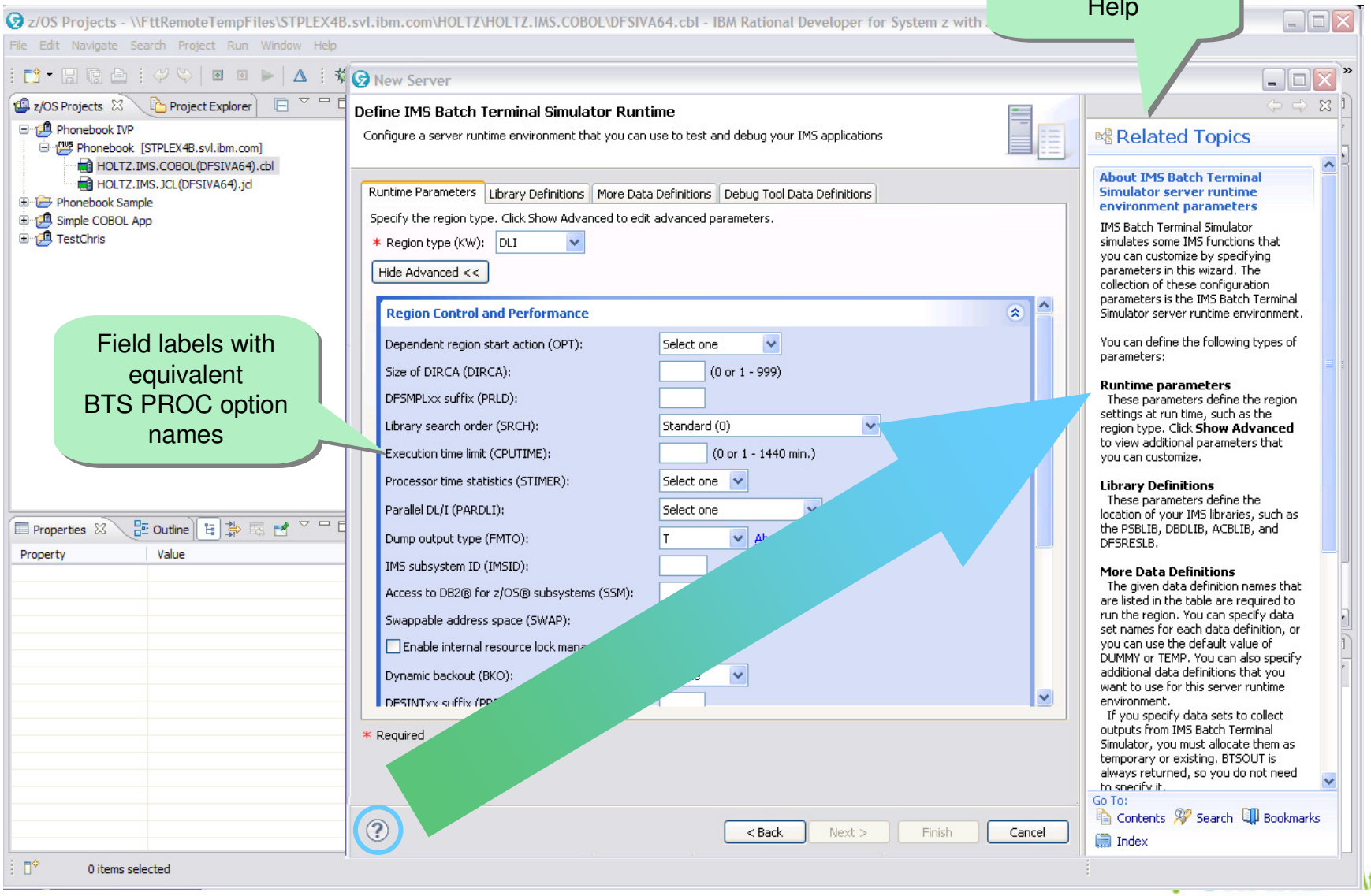
Create new or import existing server runtime configuration



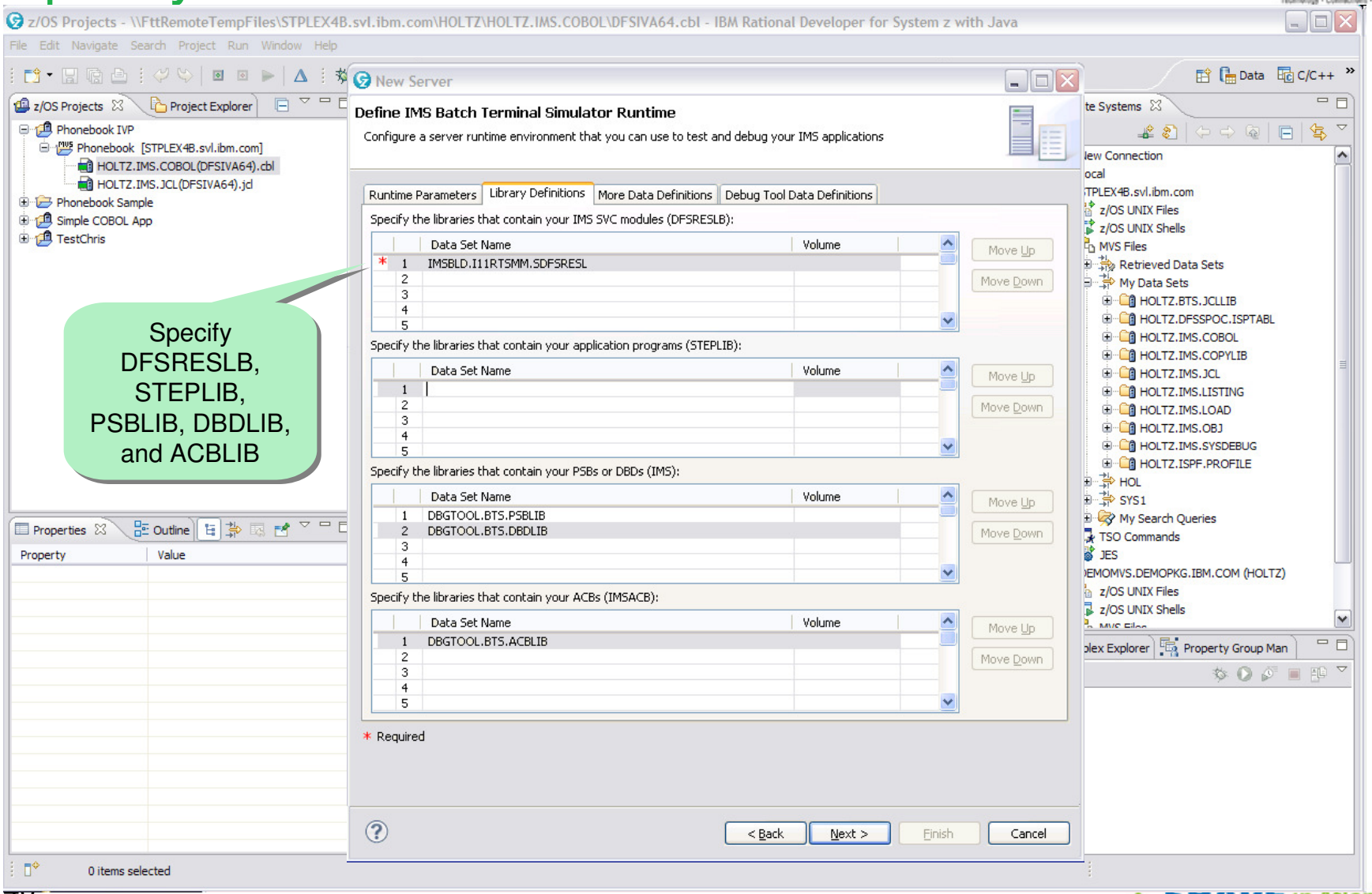
Specify BTS options



Use embedded assistance

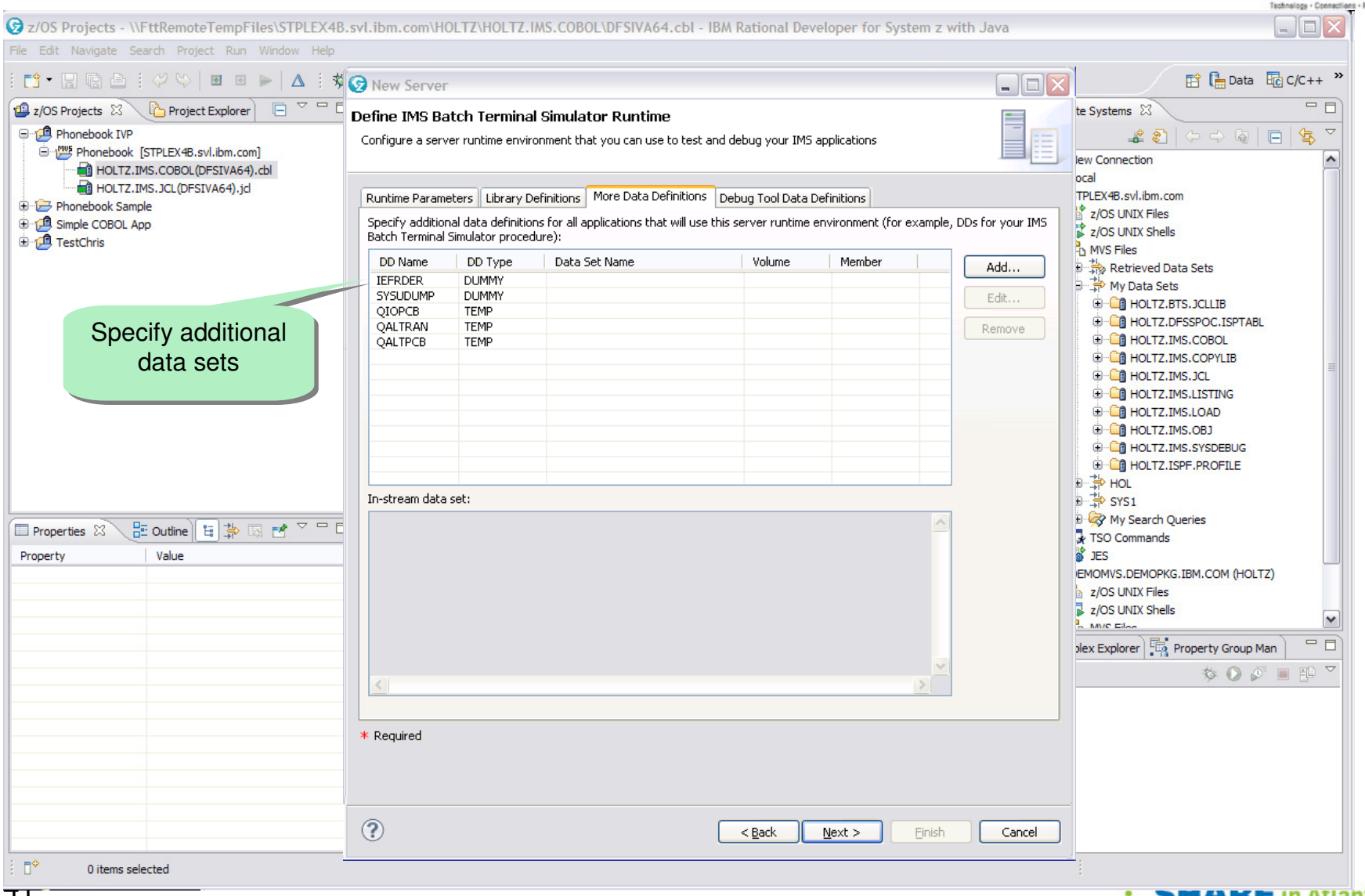


Specify libraries

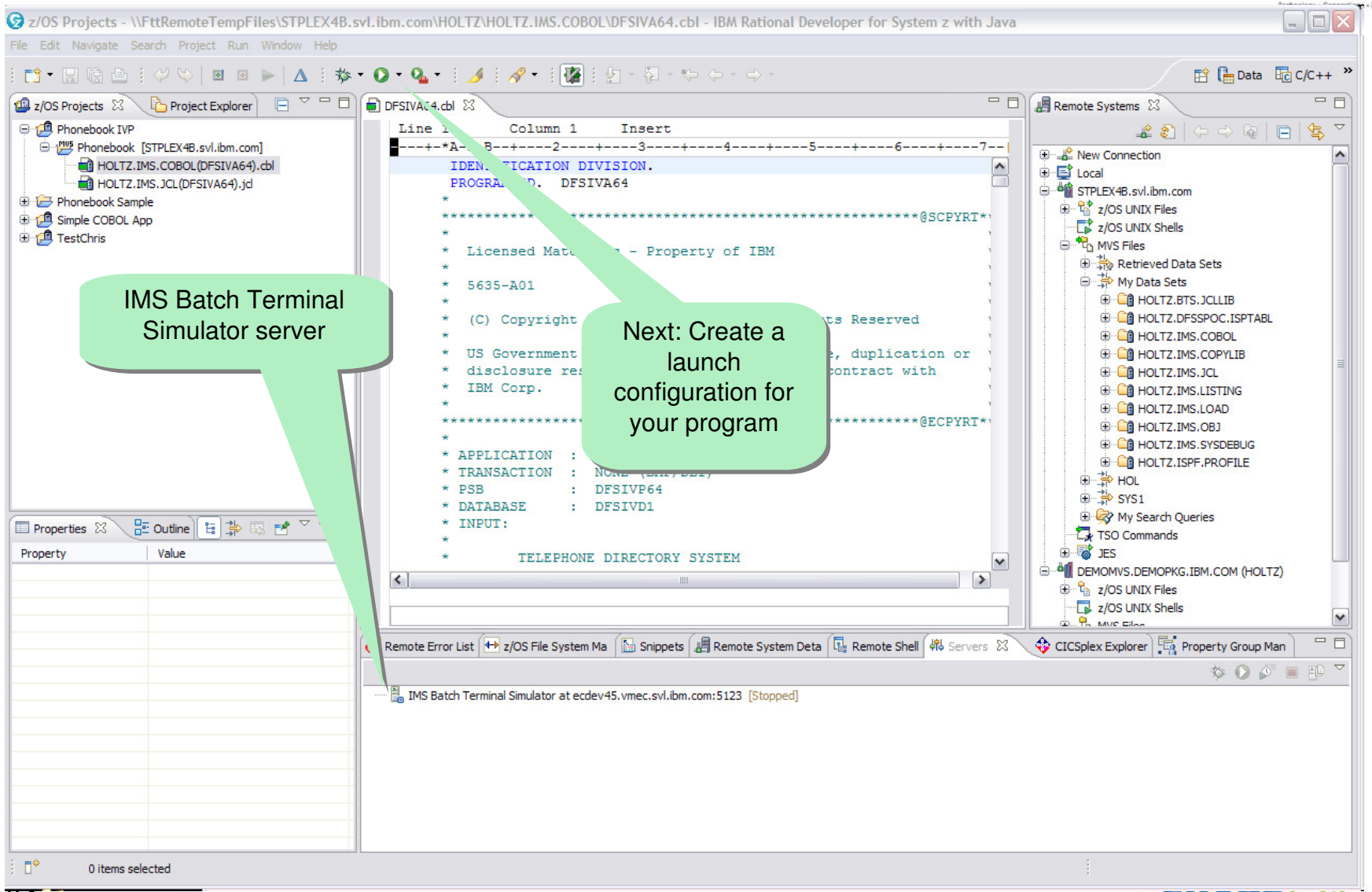


Specify DFSRESLB, STEPLIB, PSBLIB, DBDLIB, and ACBLIB

Specify additional data sets

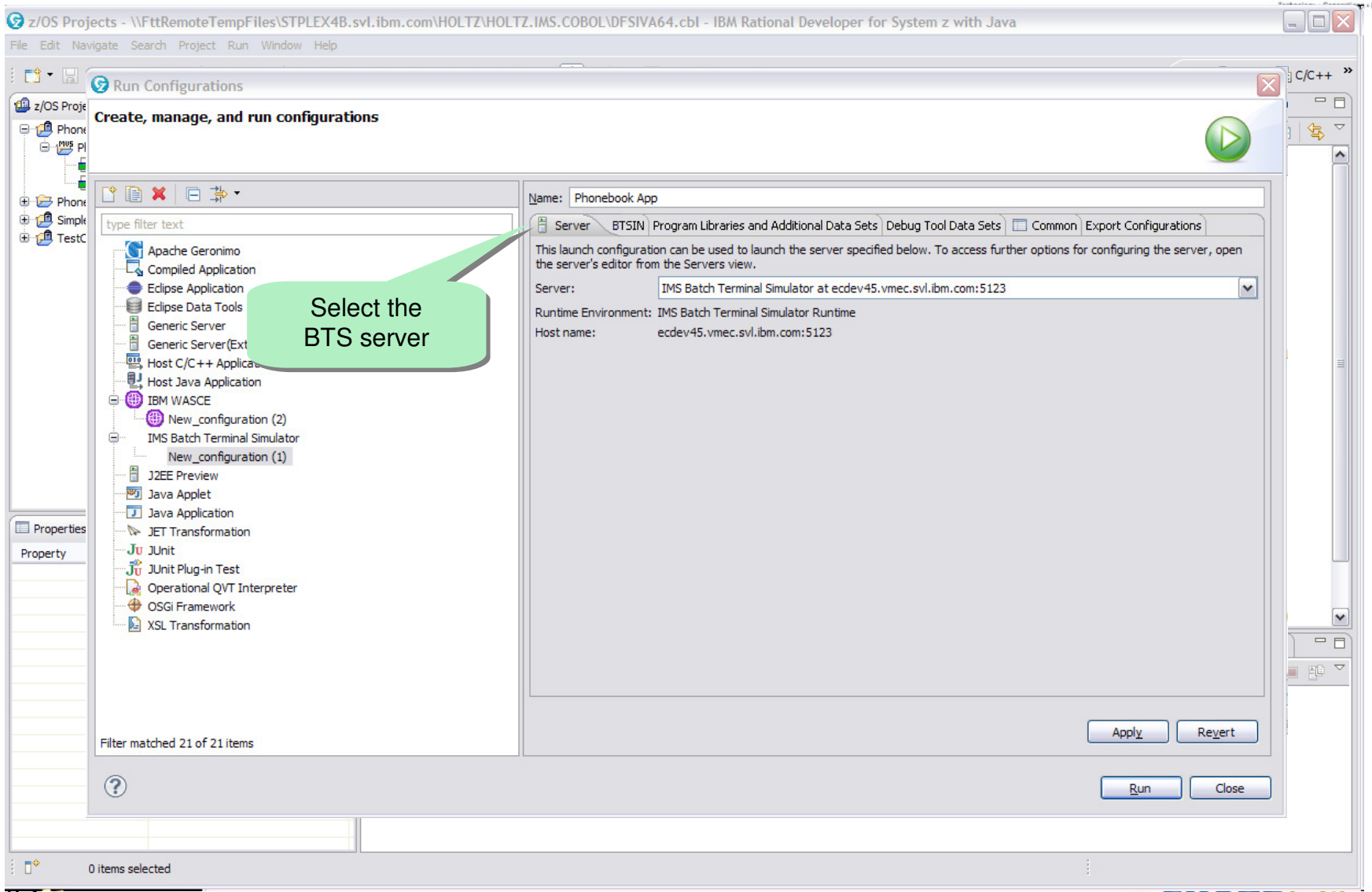


Create launch configuration

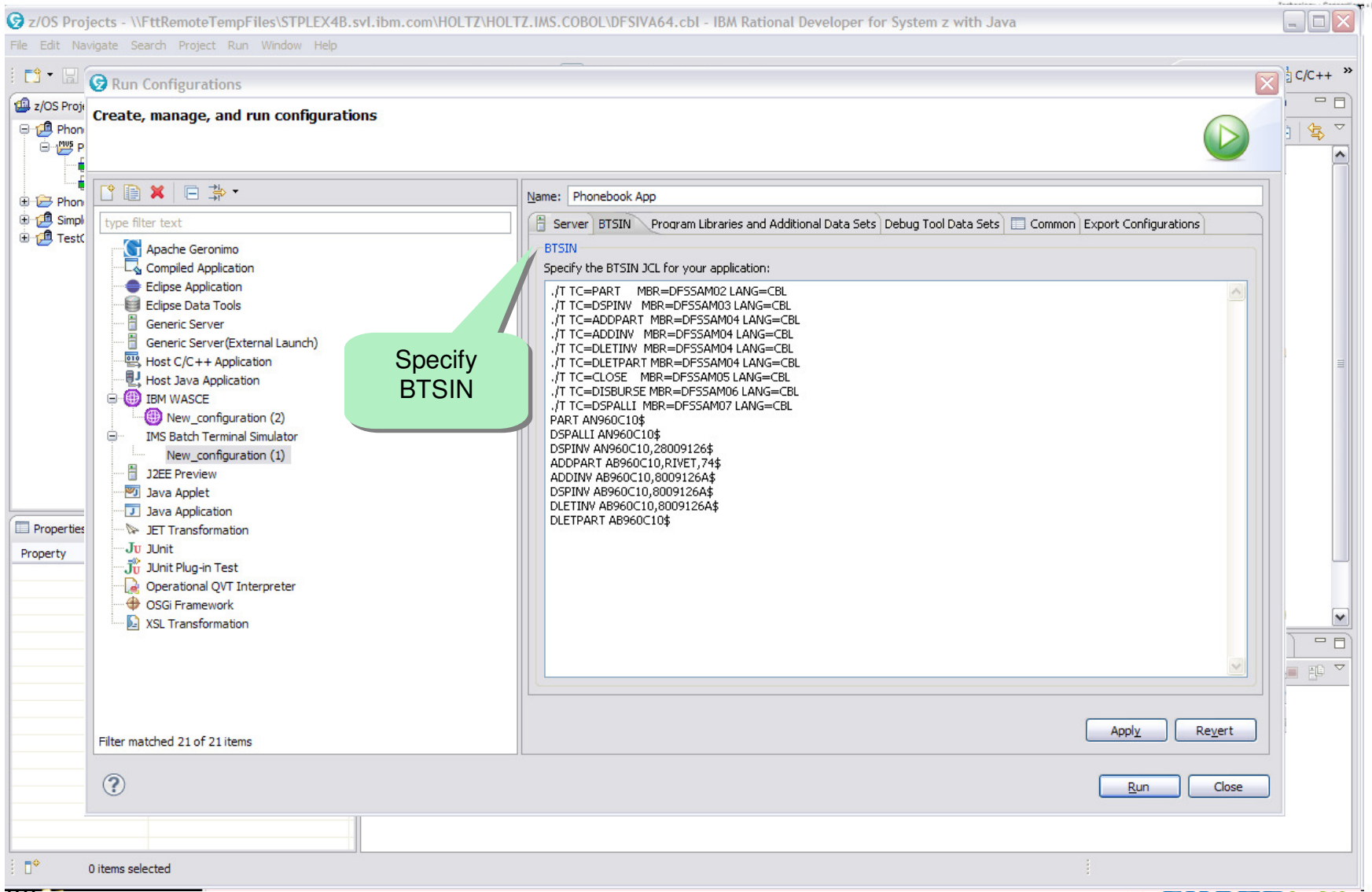


42

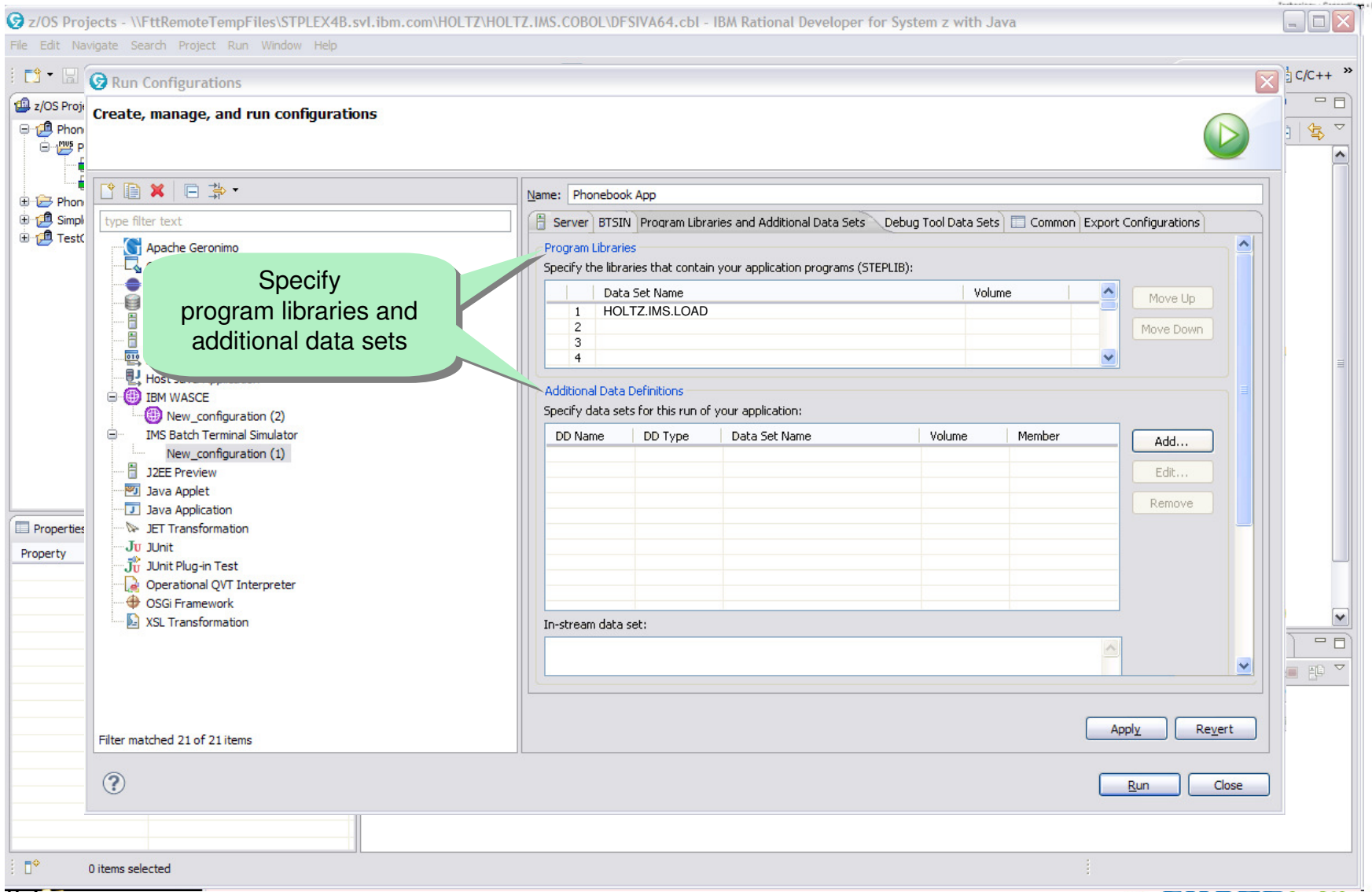
Select BTS server



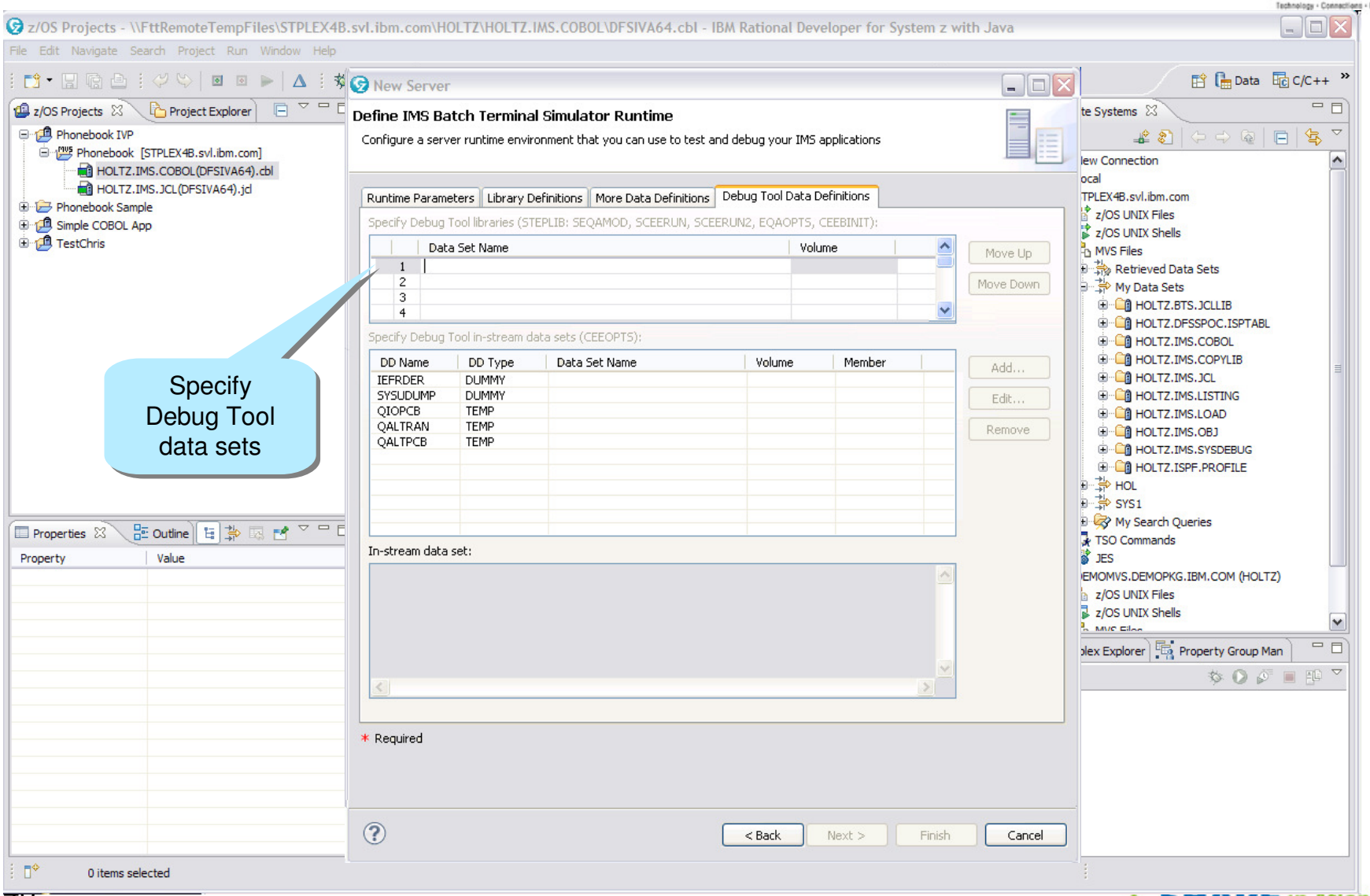
Specify BTSIN



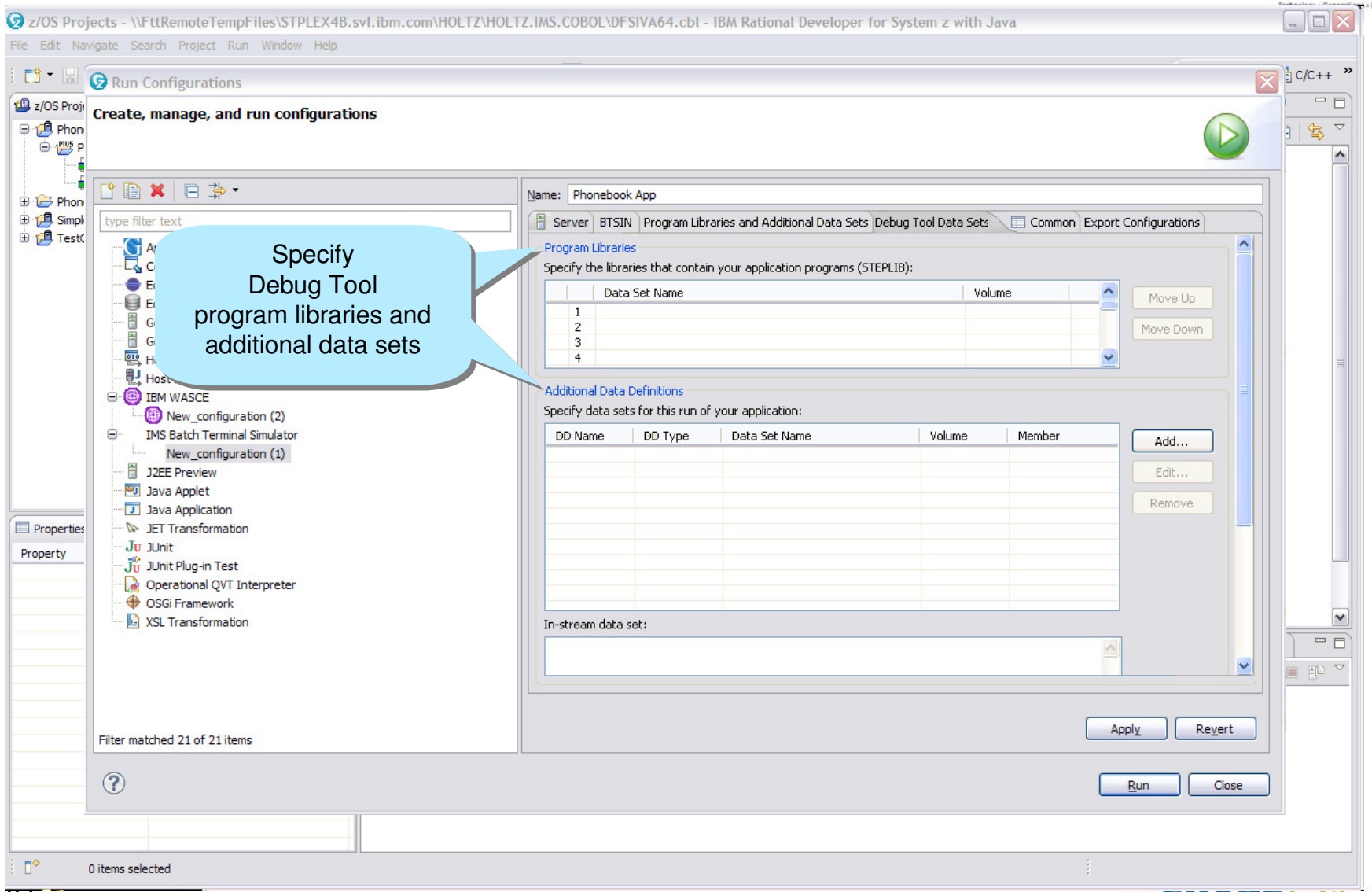
Specify program libraries



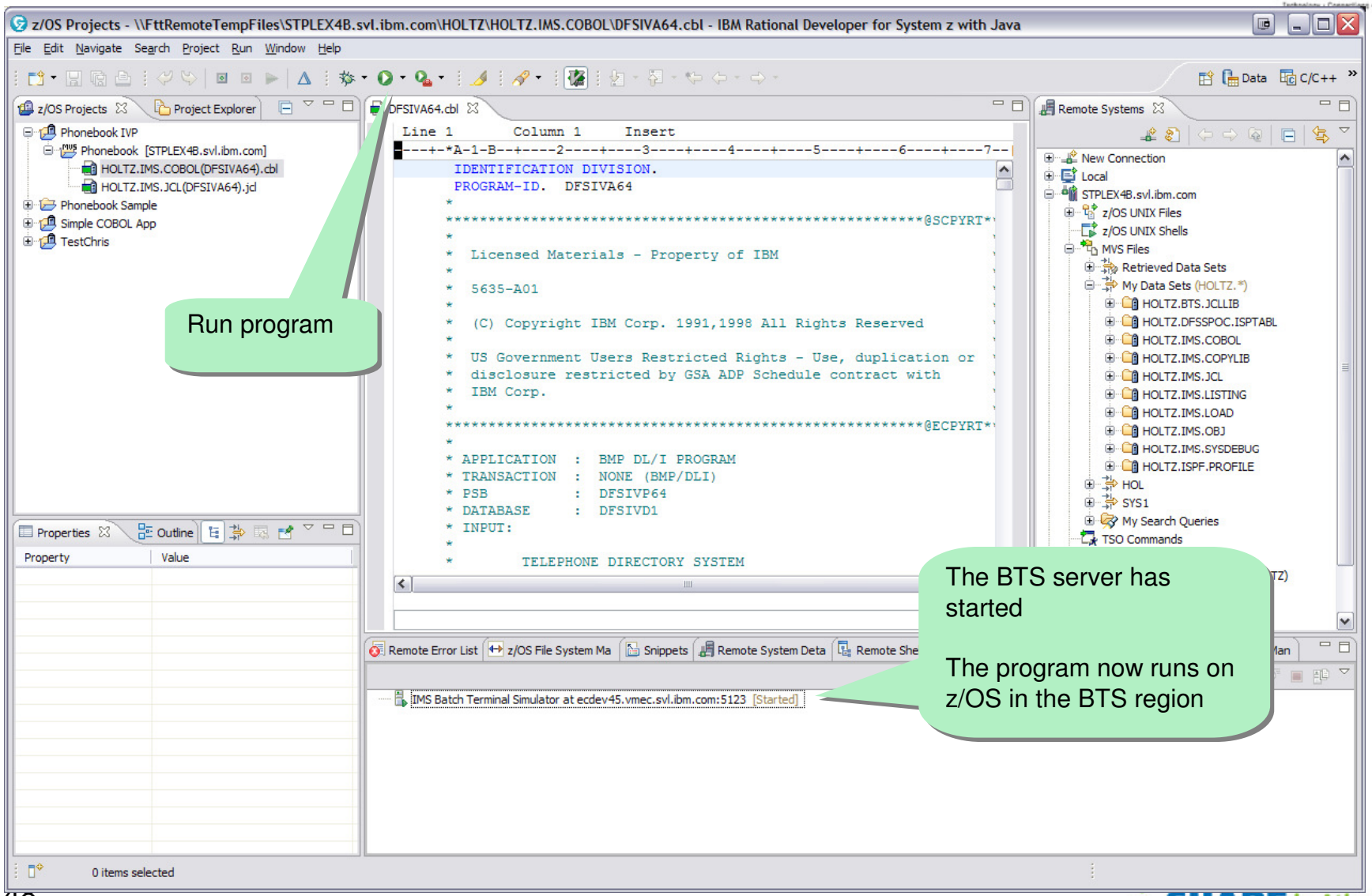
Debug Tool for z/OS
Debug runtime configuration



Debug Tool for z/OS
Debug runtime configuration

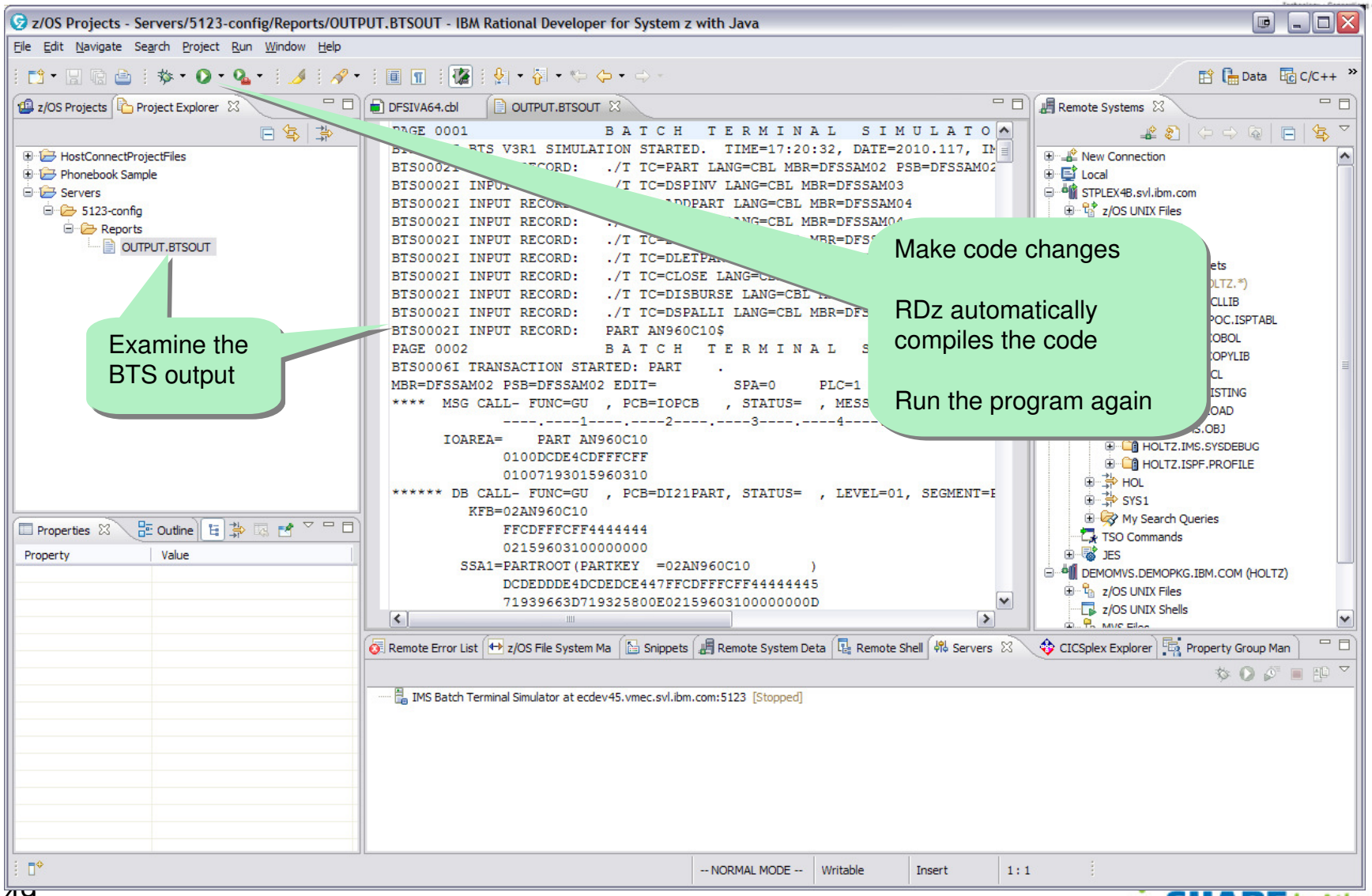


Run program



48

Examine BTS output



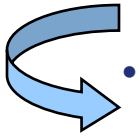
49

Be positioned to take advantage of the latest technologies

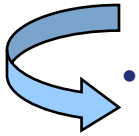
- The changing z development landscape requires modern, more productive and more affordable development tools
- The IBM approach to zSeries tools is an up-to-date toolset that provides:
  - a single set of tools across a broad spectrum of zSeries technologies to support the new generation of complex, composite applications
  - new GUIs and conventional 3270 interfaces
  - Immediate support for new versions of critical software such as DB2, CICS, MQ Series, and IMS

IBM PD Tools for z/OS

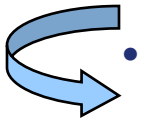
- Reduces the time programmers need to perform common development tasks such as debugging, test data creation/management, testing, and performance analysis
- Shortens application development cycles



- Provides diagnostic tools that provide detailed information about production problems, and tools for rapidly correcting data problems
- Results in reduced production down time, shortened problem resolution time, and fewer problem re-works



- Provides an extensive collection of features and utilities to automate file and data management, copying and reformatting, data scrambling, comparison, etc.



- Reduces lost of time and productivity spent writing in-house utilities
- Simplifies programming tasks during the entire development process
- Lets you spend more development time creating value, instead of struggling through mundane tasks without the right tools
- Increases productivity

Agenda

- Fault Analyzer
- Debug Tool
- Application Performance Analyzer
- File Manager
- Introducing the new plug-in for BTS (Batch Terminal Simulator)
- Conclusion

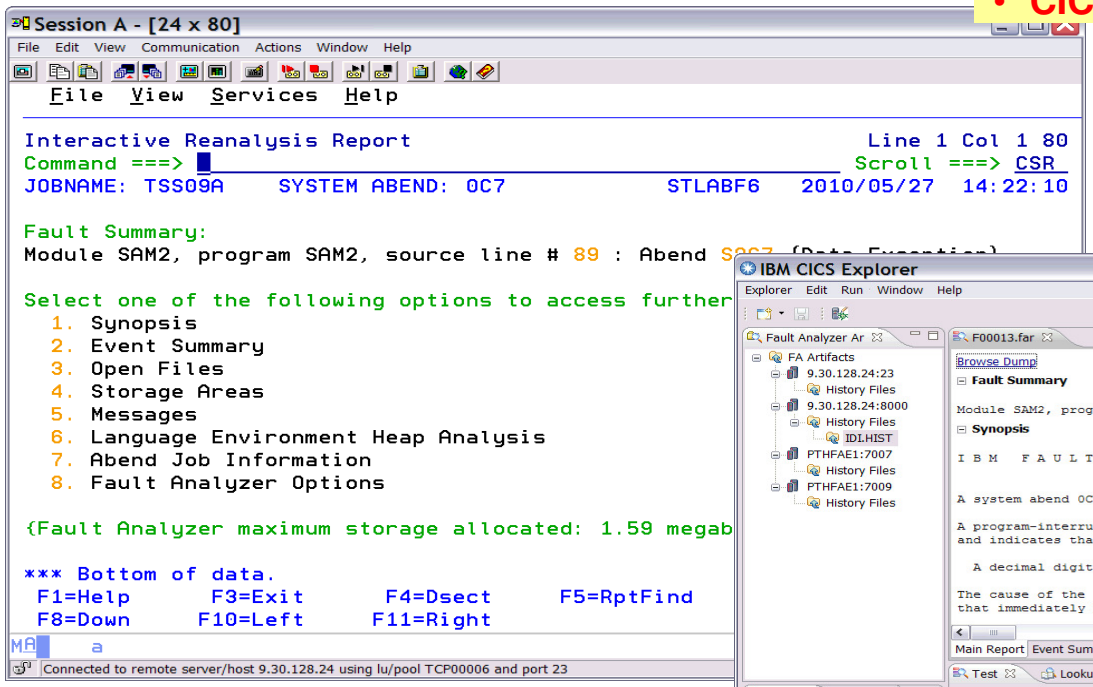
IBM Fault Analyzer
Multiple interfaces and modes of c

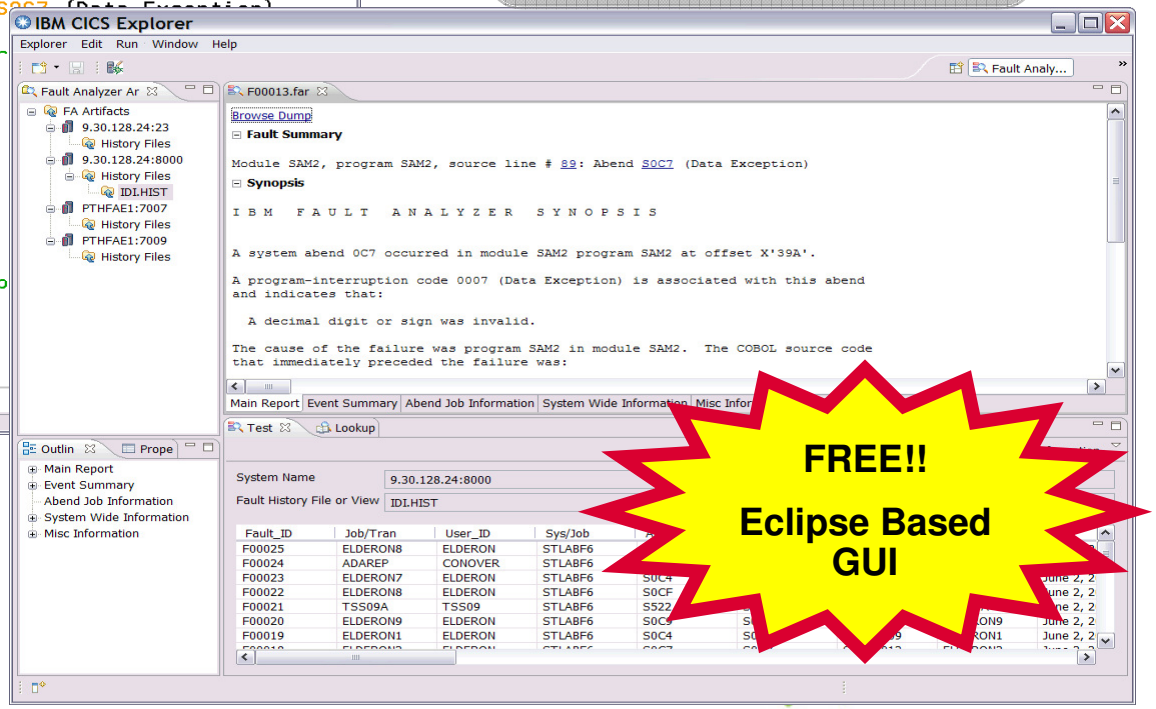
New in FA 11.1

- **Plug-in for CICS Explorer**
- **Java, C/C++, and Enterprise PL/I enhancements**
- **CICS support improvements**

Interfaces

- ISPF
- RDz
- GUI
- CICS
- Web





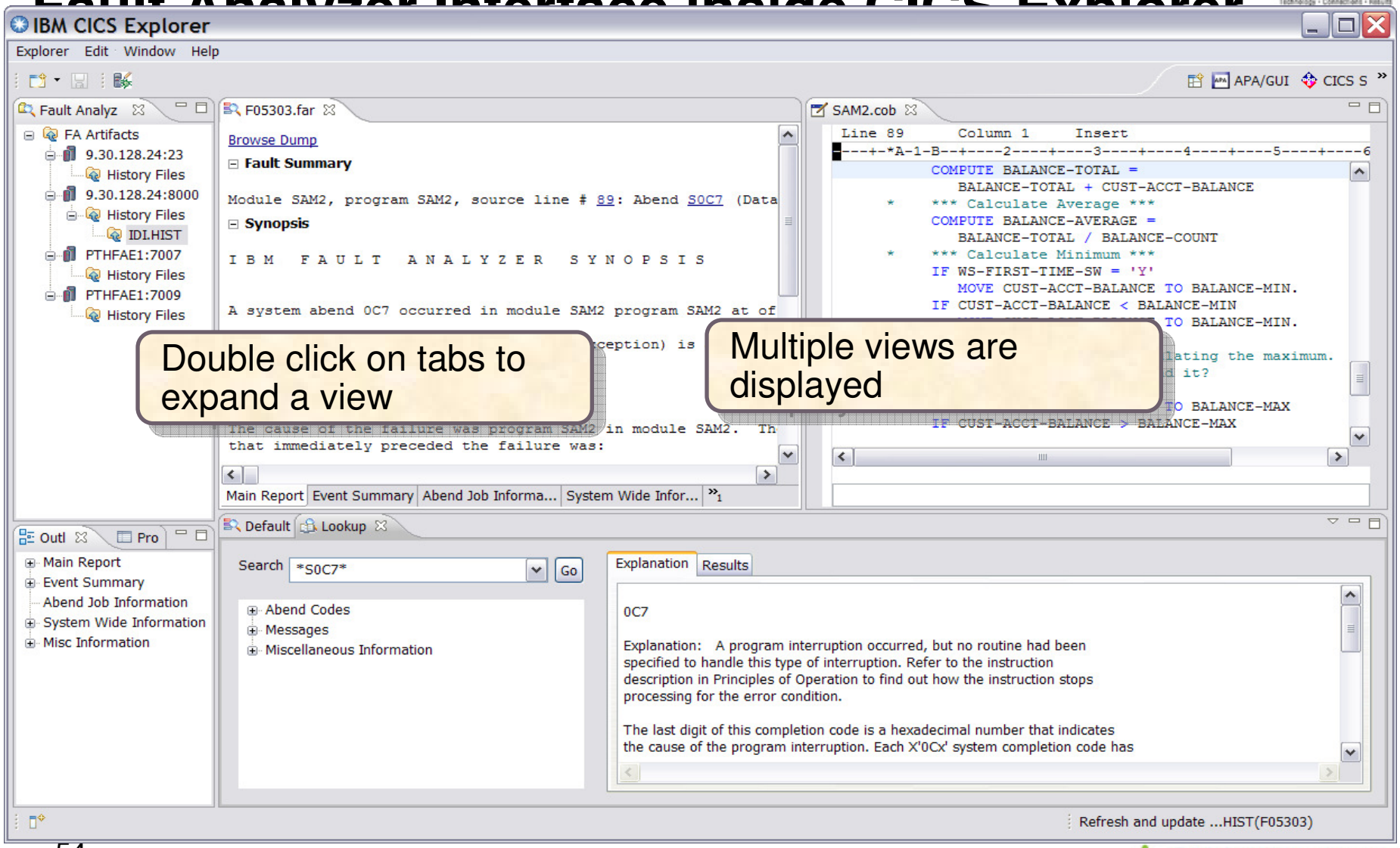
FREE!!

Eclipse Based GUI

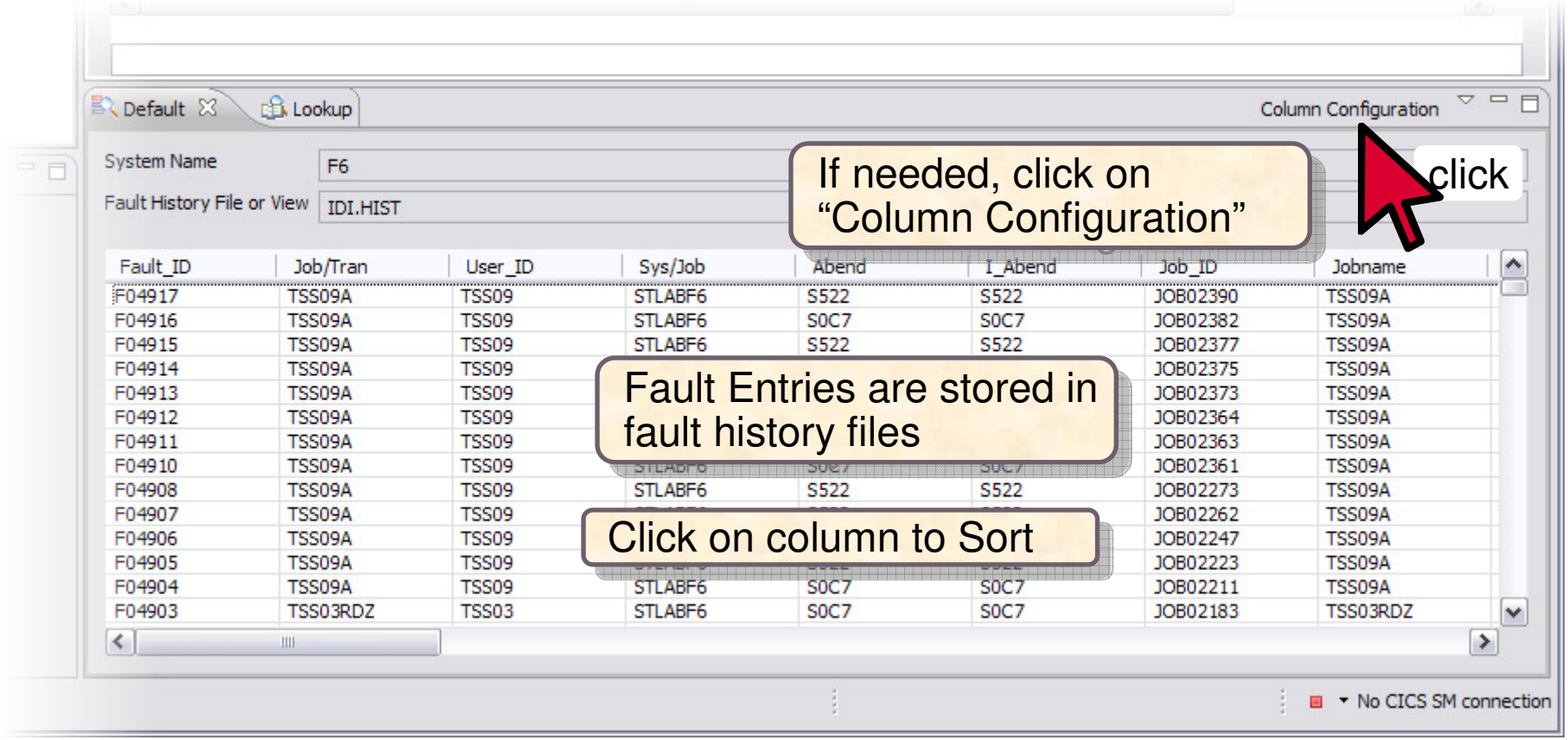
Modes Of Operation

- Real-time analysis
- Batch dump re-analysis
- Interactive dump re-analysis

Fault Analyzer Interface inside CICS Explorer

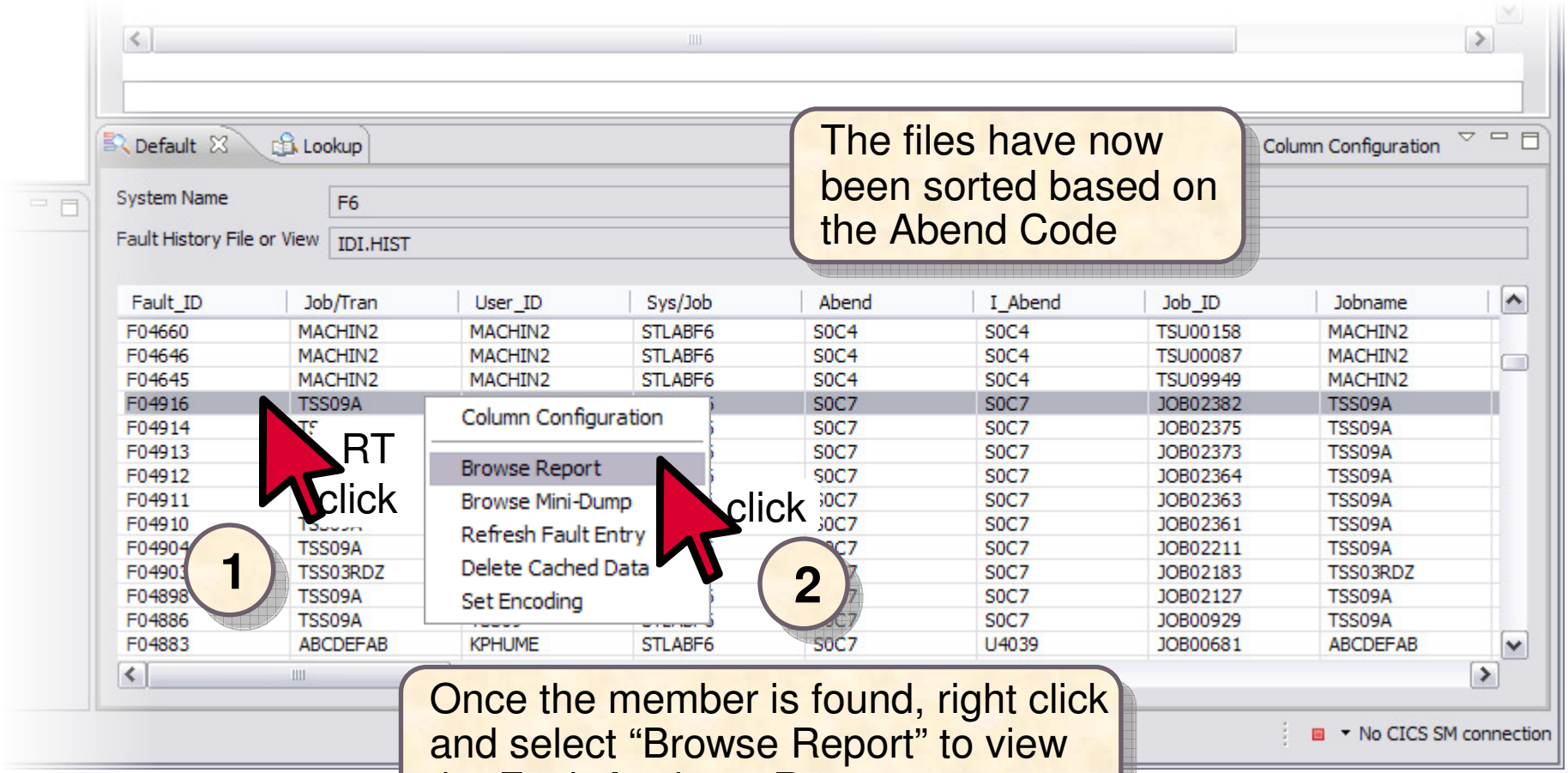


Default Fault History File View



Fault_ID	Job/Tran	User_ID	Sys/Job	Abend	I_Abend	Job_ID	Jobname
F04917	TSS09A	TSS09	STLABF6	S522	S522	JOB02390	TSS09A
F04916	TSS09A	TSS09	STLABF6	S0C7	S0C7	JOB02382	TSS09A
F04915	TSS09A	TSS09	STLABF6	S522	S522	JOB02377	TSS09A
F04914	TSS09A	TSS09				JOB02375	TSS09A
F04913	TSS09A	TSS09				JOB02373	TSS09A
F04912	TSS09A	TSS09				JOB02364	TSS09A
F04911	TSS09A	TSS09				JOB02363	TSS09A
F04910	TSS09A	TSS09				JOB02361	TSS09A
F04908	TSS09A	TSS09	STLABF6	S522	S522	JOB02273	TSS09A
F04907	TSS09A	TSS09				JOB02262	TSS09A
F04906	TSS09A	TSS09				JOB02247	TSS09A
F04905	TSS09A	TSS09				JOB02223	TSS09A
F04904	TSS09A	TSS09	STLABF6	S0C7	S0C7	JOB02211	TSS09A
F04903	TSS03RDZ	TSS03	STLABF6	S0C7	S0C7	JOB02183	TSS03RDZ

Opening a Fault Entry



The files have now been sorted based on the Abend Code

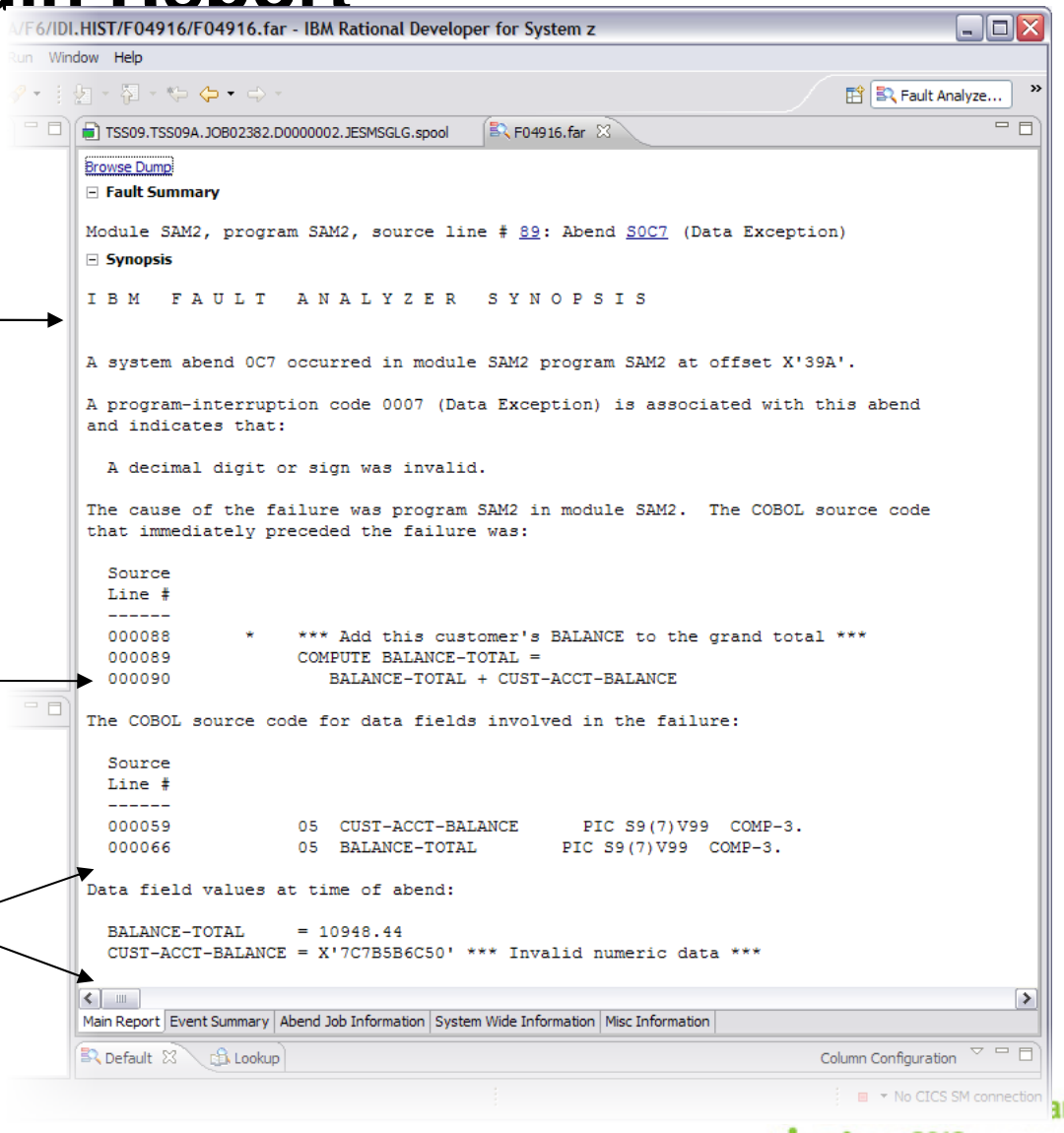
Once the member is found, right click and select “Browse Report” to view the Fault Analyzer Report

Fault Analyzer Main Report

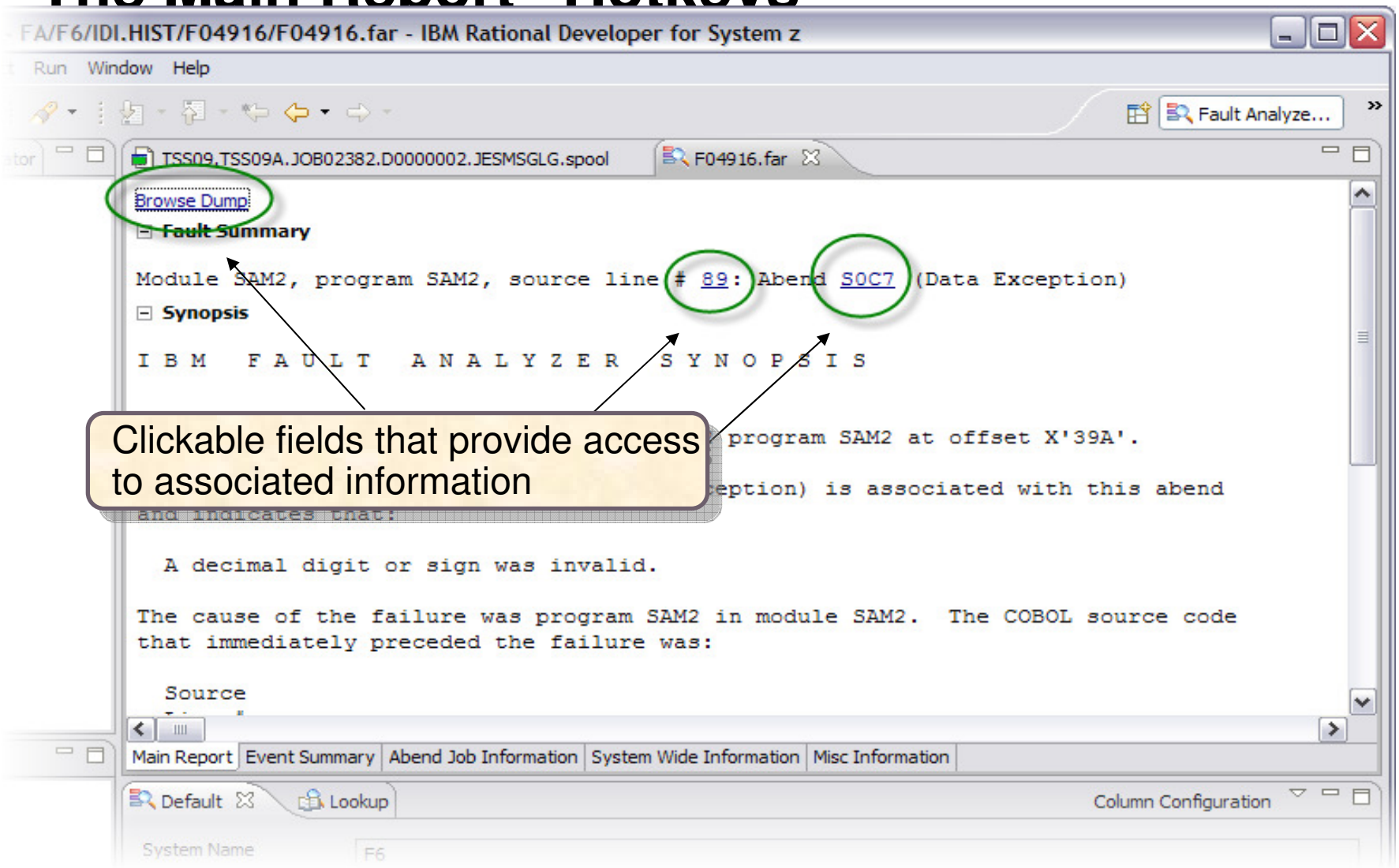
Fault Analyzer Synopsis

Source code that preceded the ABEND

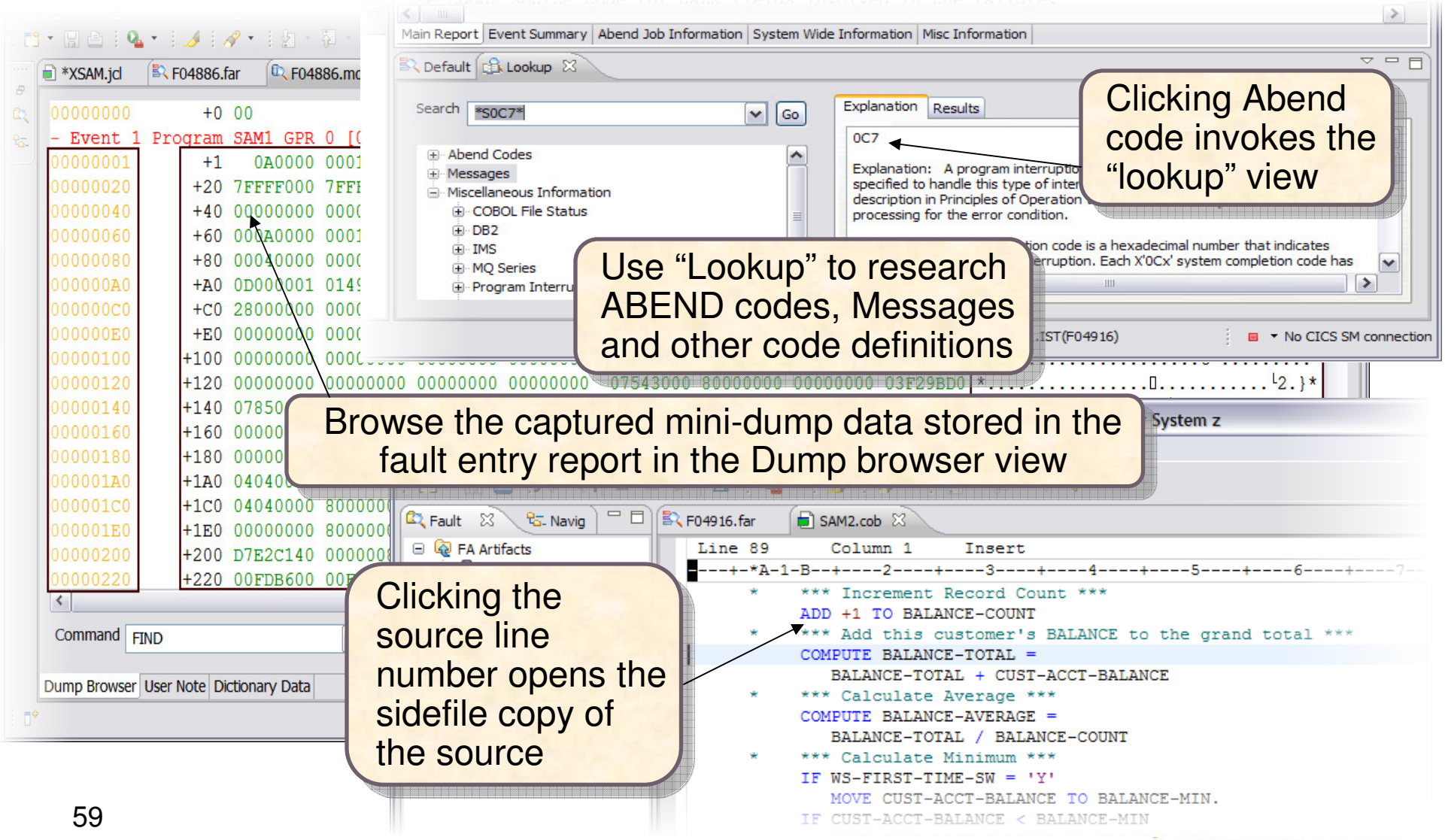
Data Field Information



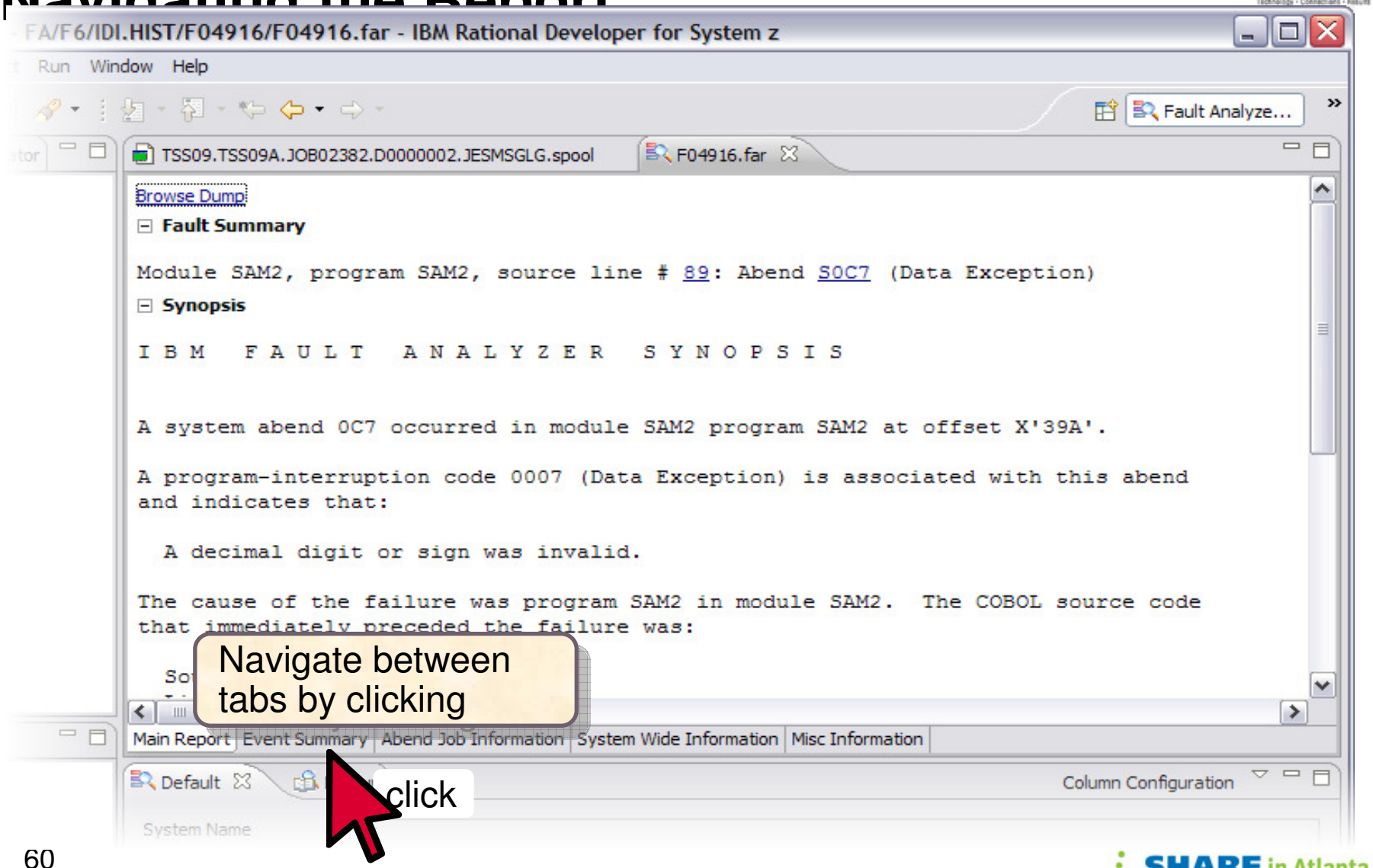
The Main Report "Hotkeys"



Results of clicking Hotkeys



Navigating the Report

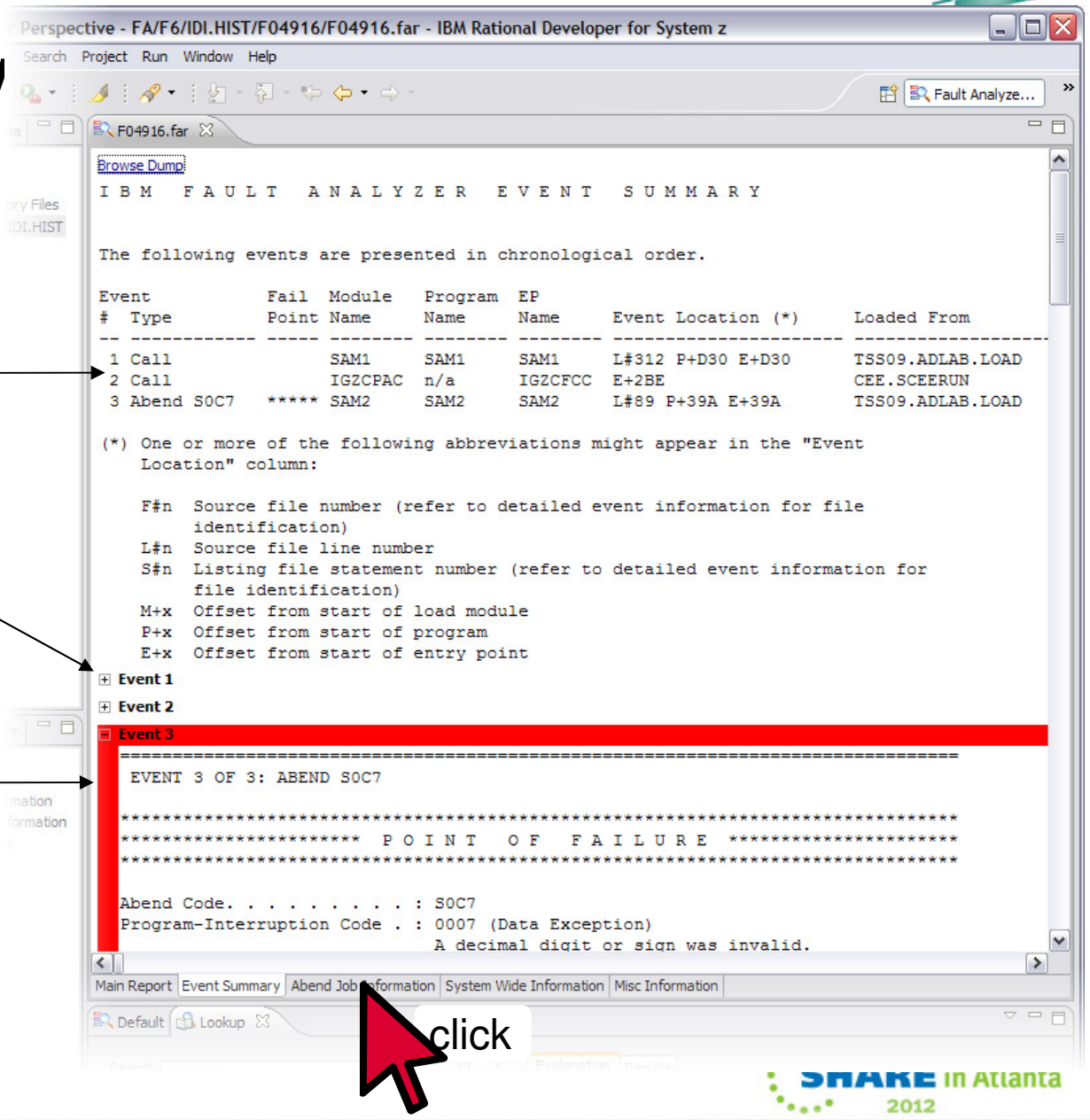


Event Summary

Chronological order of events

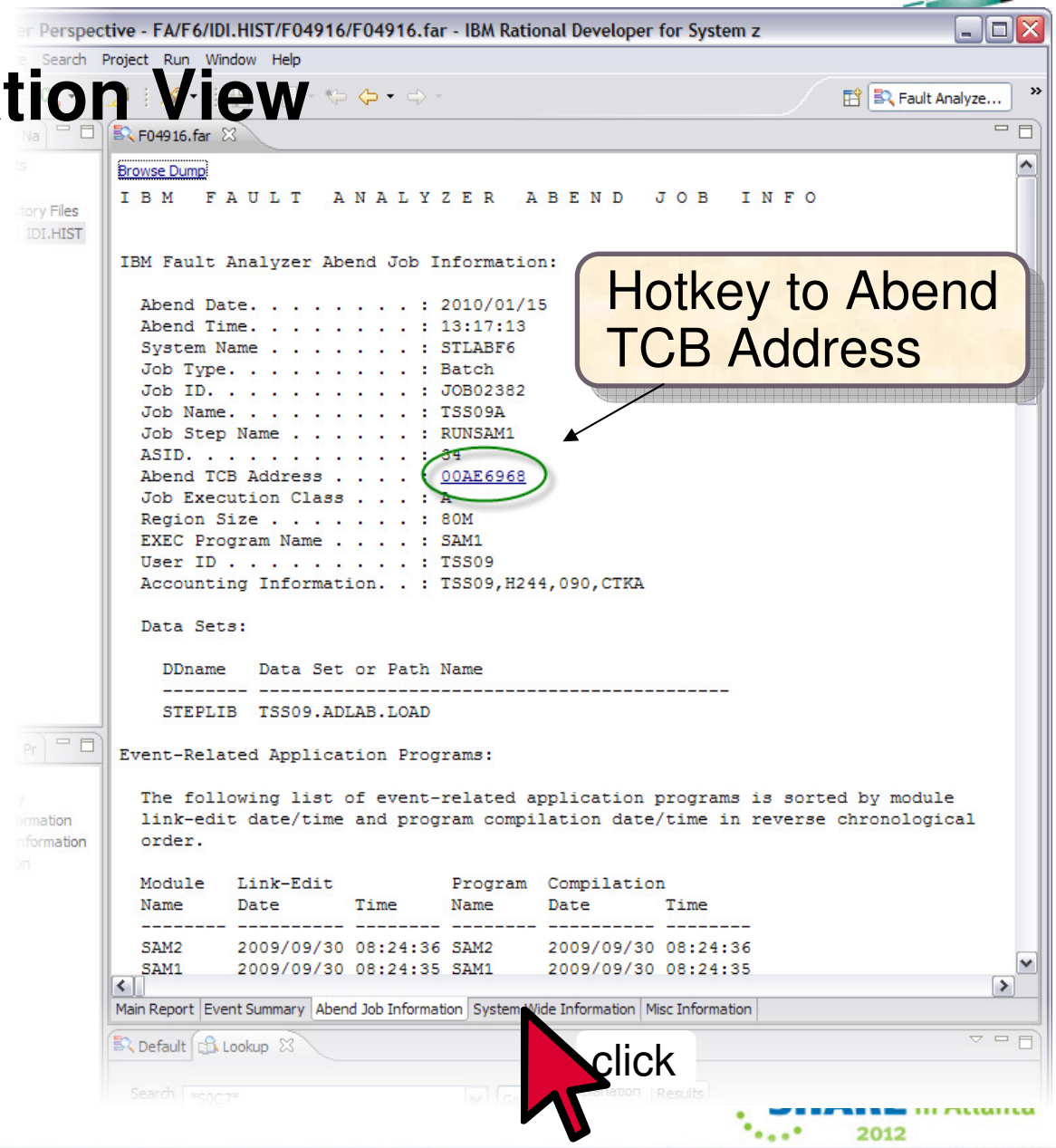
Expandable event details

Highlighted “POINT OF FAILURE” event details

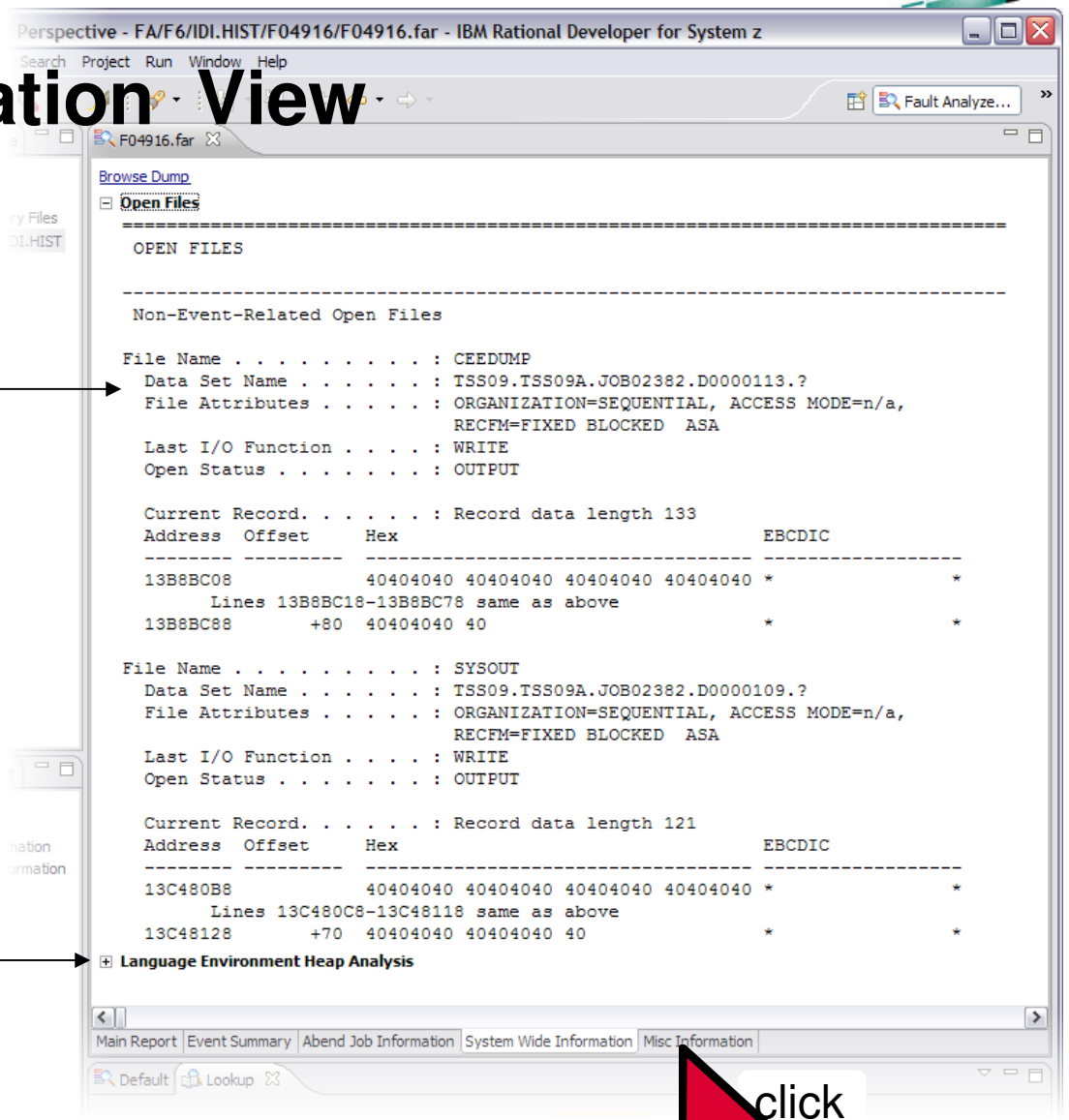


Abend Job Information View

ABEND Job details



System Wide Information View



Open Files

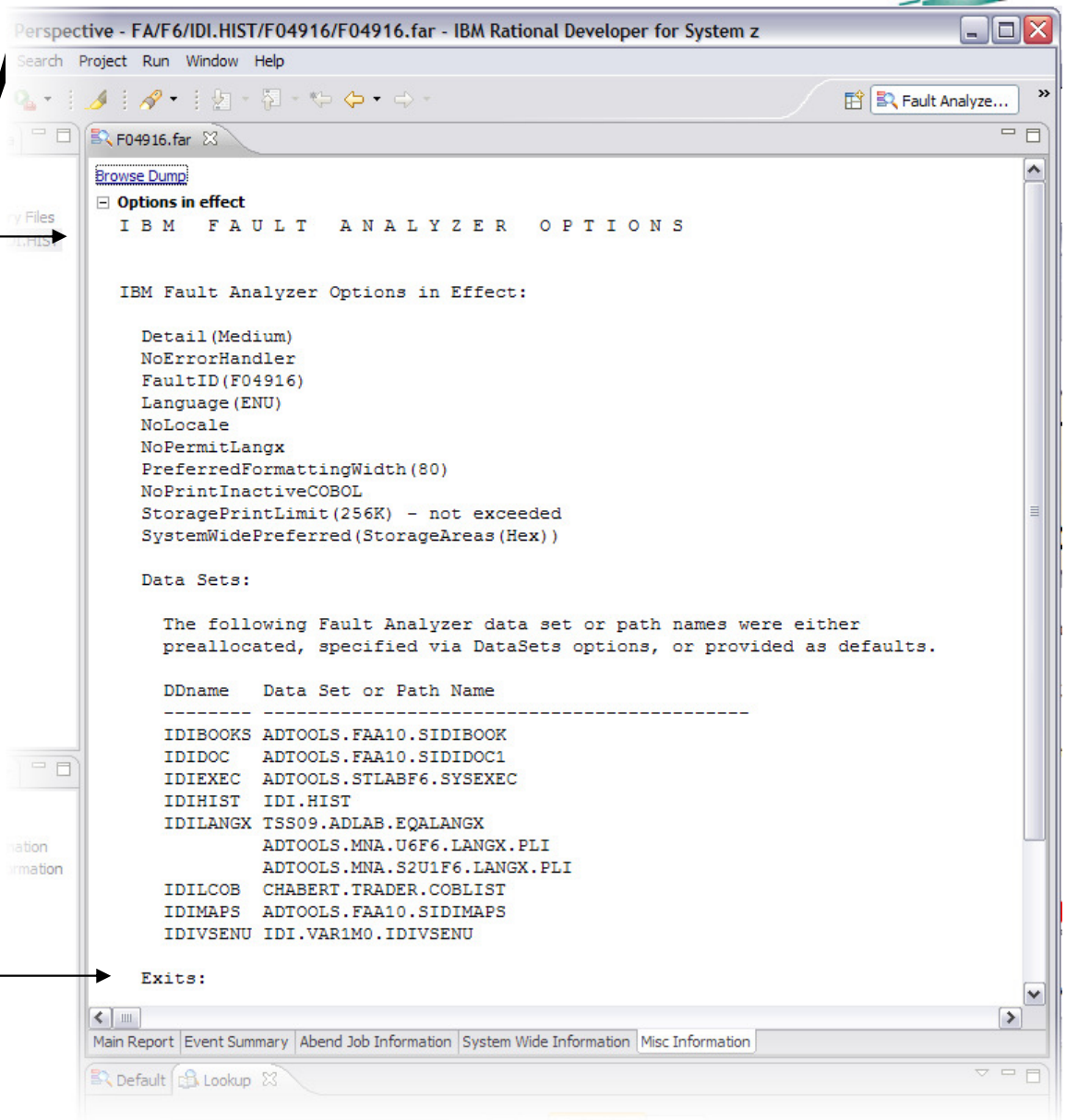
LE Heap Analysis

click

Misc Information V

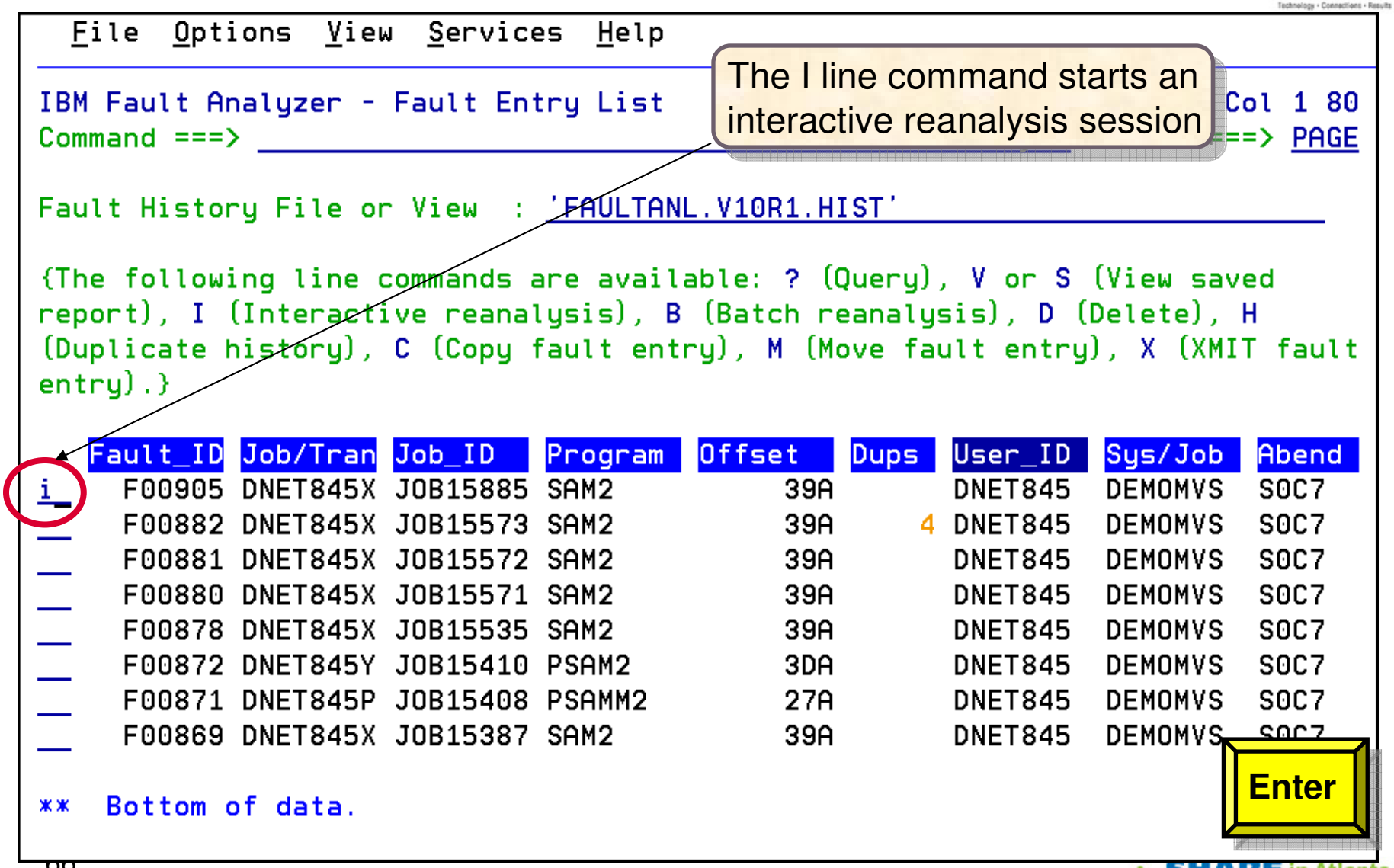
Options in effect

Exits

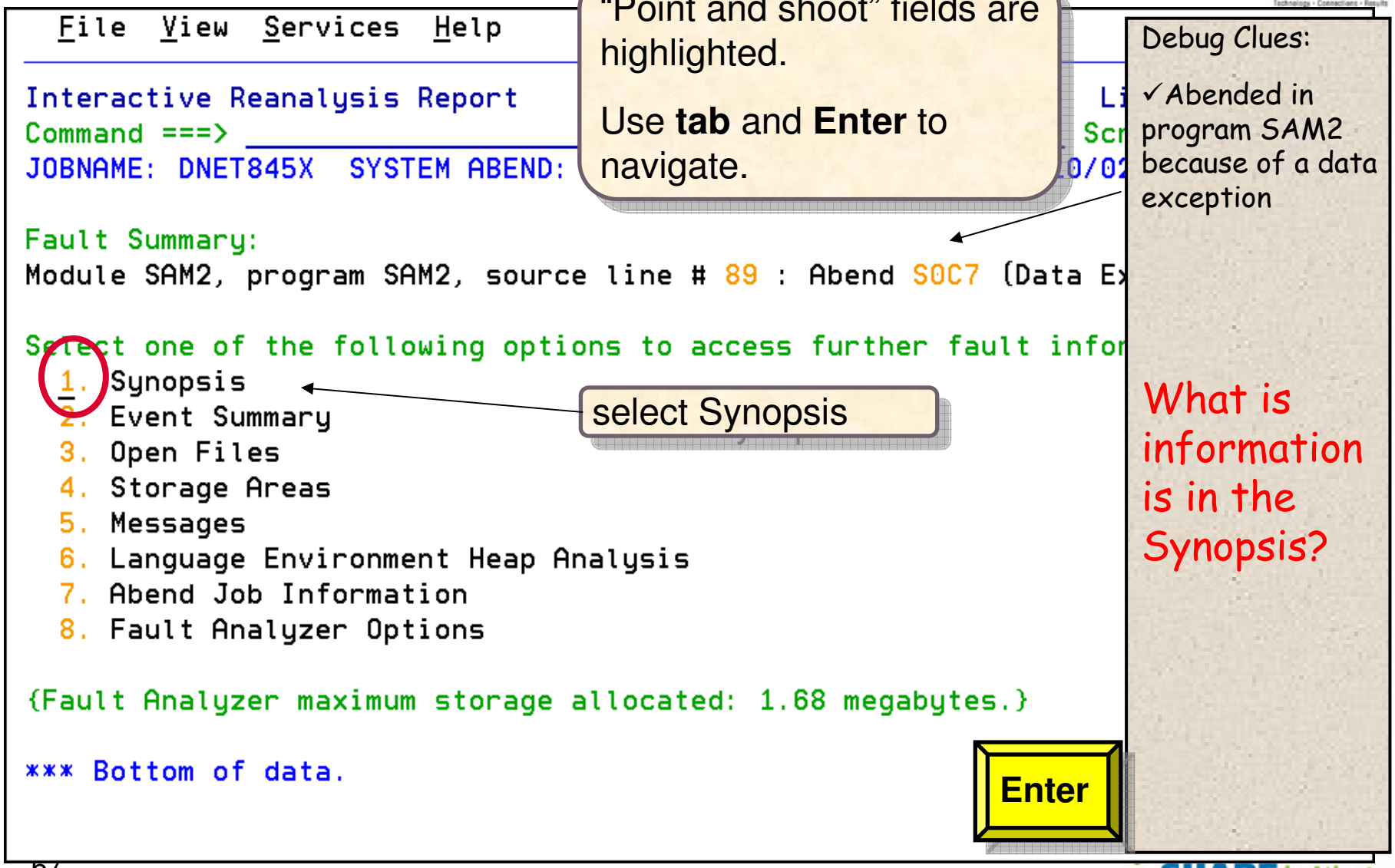


The Fault Analyzer TSO Interface

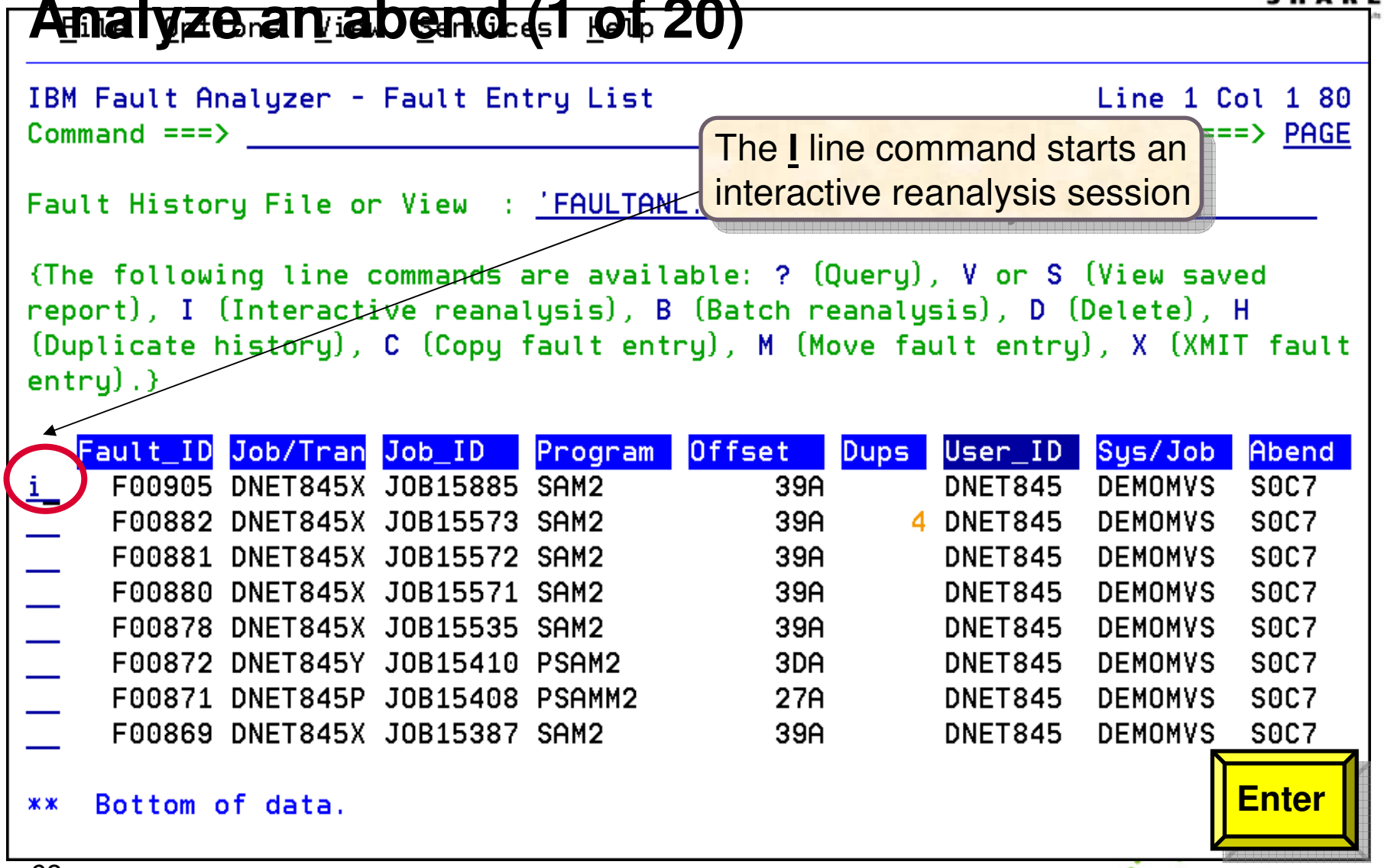
Using Interactive reanalysis **to analyze an abend**



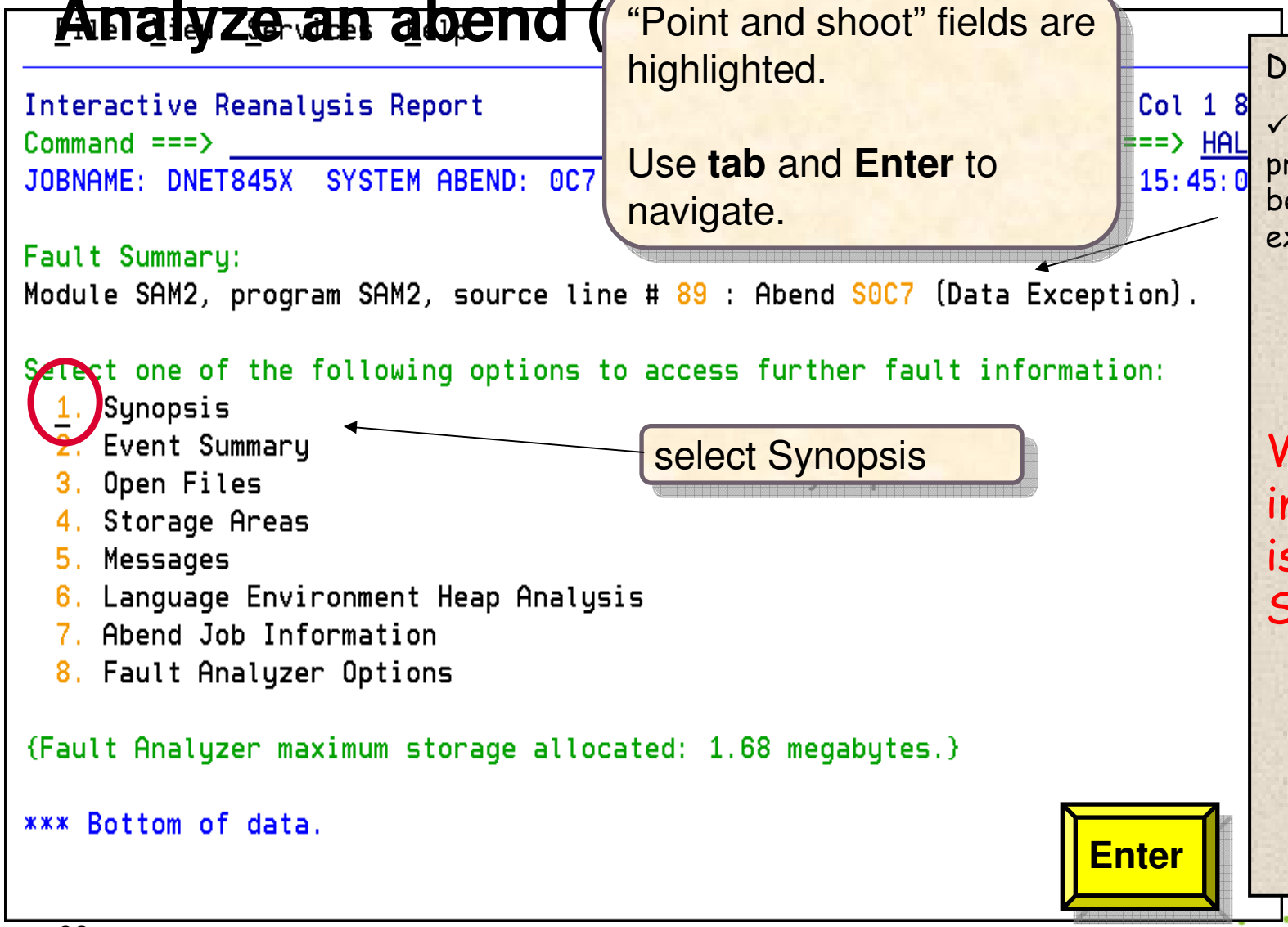
Analyze an abend



Analyze an abend (1 of 20)



Analyze an abend (



"Point and shoot" fields are highlighted.

Use **tab** and **Enter** to navigate.

select Synopsis

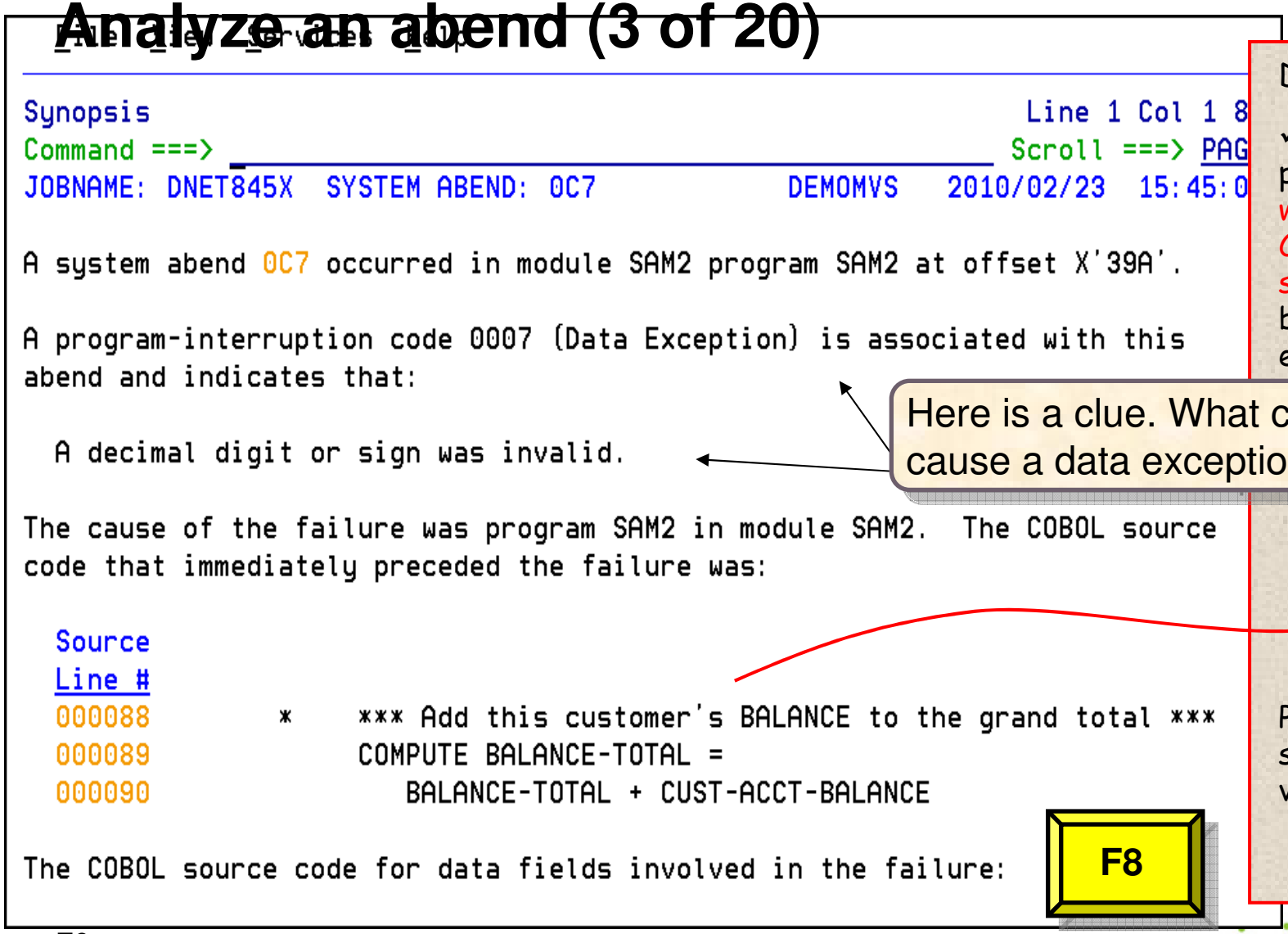
Enter

Debug Clues:

✓Abended in program SAM2 because of a data exception

What is information is in the Synopsis?

Analyze an abend (3 of 20)



Here is a clue. What can cause a data exception?

Debug Clues:

✓Abended in program SAM2 while running a COMPUTE statement because of a data exception

Page forward to see active variables

Analyze an abend (4 of 20)

What variable contained the bad data?

```
Synopsis                                                    2 Col 1 8
Command ===> _____________________________________________ Scroll ===> PAG
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0
  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' *** Invalid numeric data ***

*** Bottom of data.
```

Debug Clues:

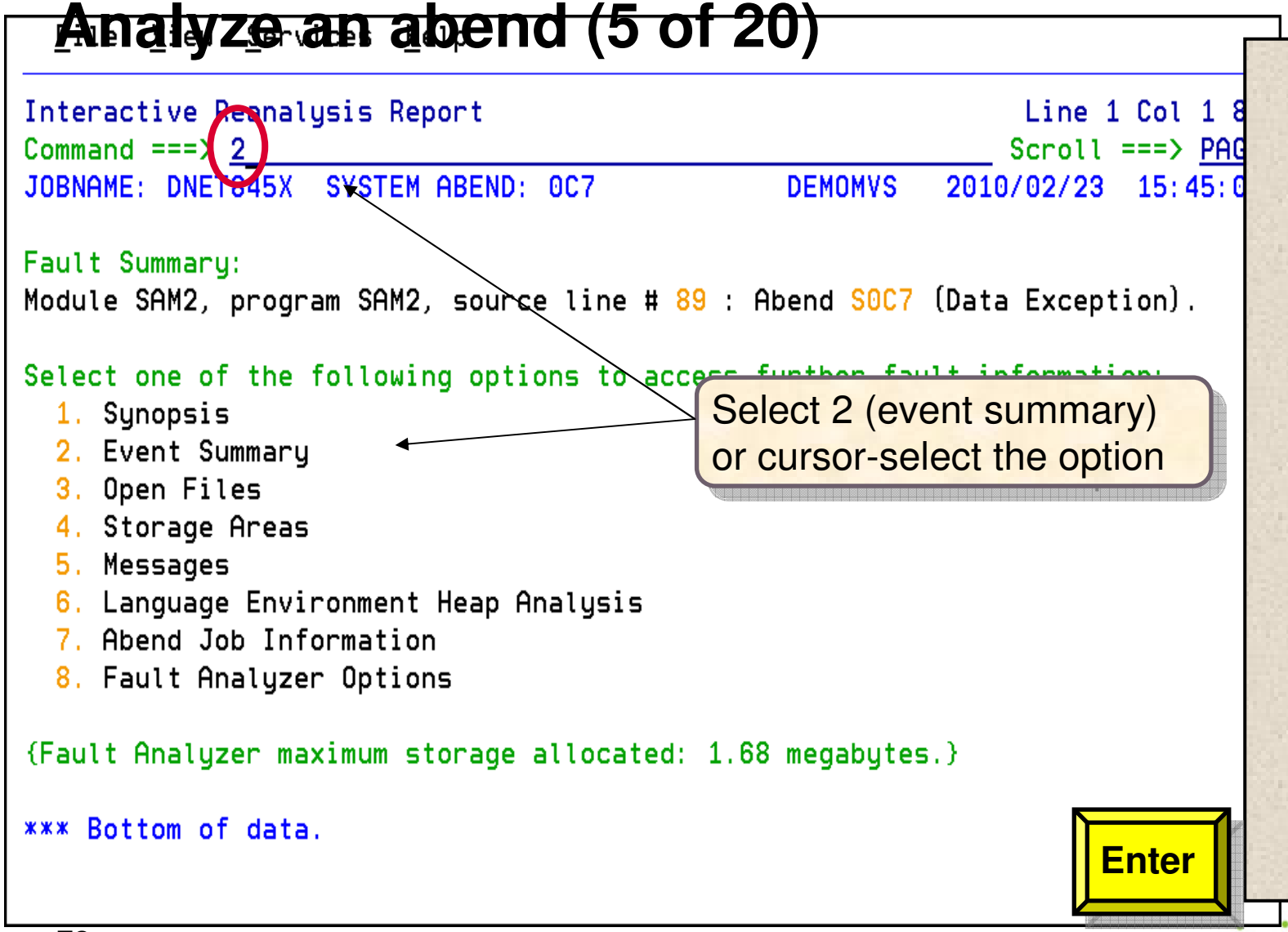
✓Abended in program SAM2 while running a COMPUTE statement because of a data exception

Go look at the bad variable

Return to menu

F3

Analyze an abend (5 of 20)



Select 2 (event summary) or cursor-select the option

Enter

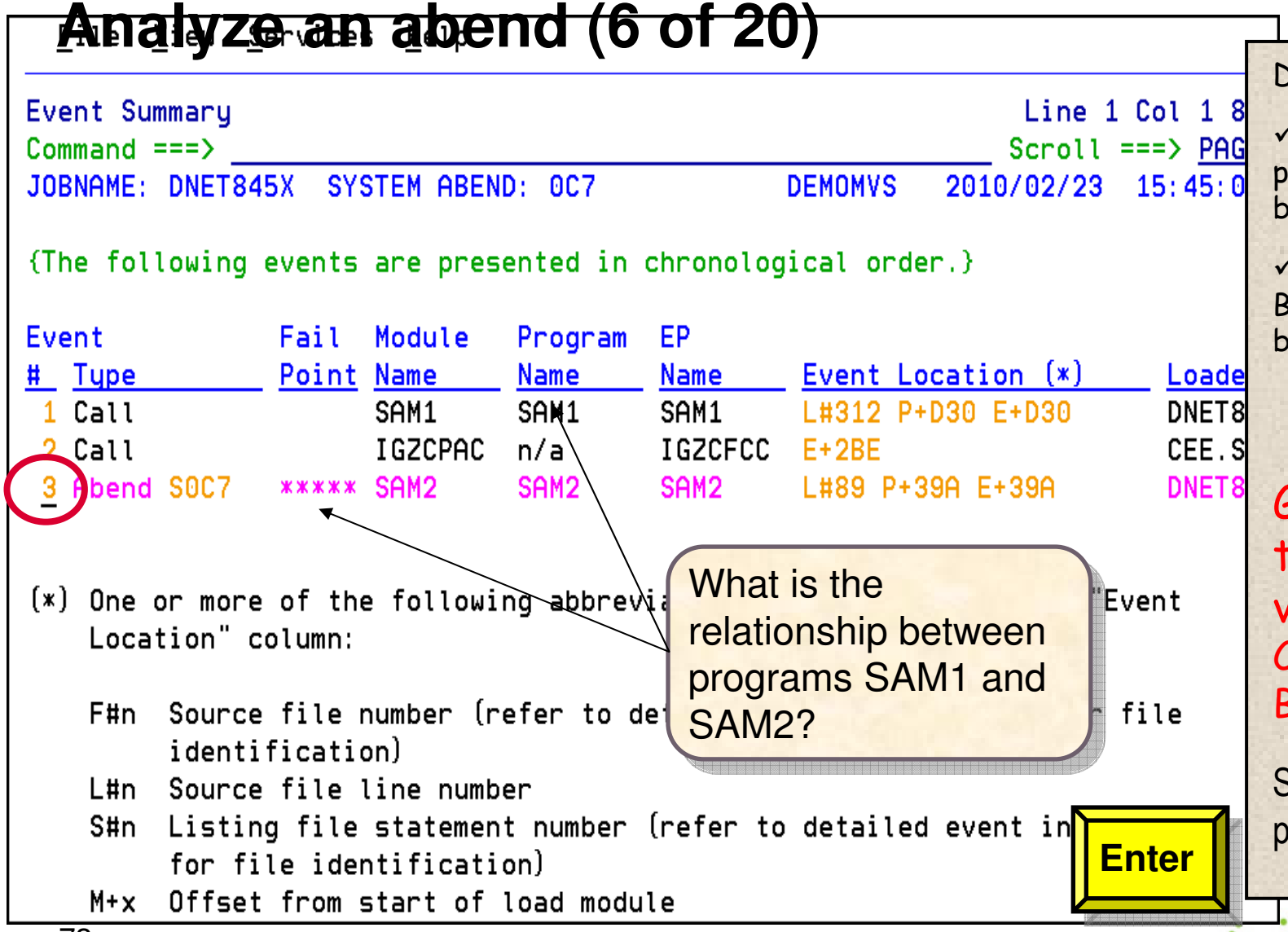
Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data

Go look at the bad variable, CUST-ACCT-BALANCE

Analyze an abend (6 of 20)



What is the relationship between programs SAM1 and SAM2?

Enter

Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data

Go look at the bad variable, CUST-ACCT-BALANCE

Select detail for program SAM2

Analyze an abend (7 of 20)

```
Event 3 of 3: Abend S0C7 *** Point of Failure ***              Line 1 Col 1 8
Command ===> bottom                                          Scroll ===> HAL
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0

Previous Event Details


Abend Code. . . . . . . . . . : S0C7
Program-Interruption Code . . : 0007 (Data Exception)
  A decimal digit or sign was invalid.

The source code below was executed via the following sequence of PERFORM
statements:
  Source
  Line #
  000079             PERFORM 100-CALC-BALANCE-STATISTICS.

COBOL Source Code:
  Source
  Line #
  000088        *    *** Add this customer's BALANCE to the grand to
  000089             COMPUTE BALANCE-TOTAL =
```

Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data

Go look at the bad variable, CUST-ACCT-BALANCE

Go to the bottom of the program detail display

Enter

Analyze an abend (8 of 20)

```
Event 3 of 3: Abend S0C7 *** Point of Failure ***              Line 83 Col 1 8
Command ===> _____________________________________________  Scroll ===> PAG
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0
  R6: 205910C0 (606016 bytes of storage addressable)
  R7: 2050DA20 (1144288 bytes of storage addressable)
  R8: 0003F7C8 (Module SAM2 program SAM2 WORKING-STORAGE SECTION BLW=0000 +
               X'0', symbol WS-FIELDS, source line # 36 )
  R9: 0003F5D0 (Module SAM2 program SAM2 + X'5D0', source line # 116 )
  R10: 0003F11C (Module SAM2 program SAM2 + X'11C')
  R11: 0003F29C (Module SAM2 program SAM2 + X'29C')
  R12: 0003F0FC (Module SAM2 program SAM2 + X'FC')
  R13: 20594458 (592808 bytes of storage addressable)
  R14: 8003F392 (Module SAM2 program SAM2 + X'392', source line # 89 )
  R15: 8003F224 (Module SAM2 program SAM2 + X'224')

Associated Messages

CEE3207S The system detected a data exception (System Completion Code=0C7).

Associated Storage Areas

*** Bottom of data.
```

What is shown here?

Enter

Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data

Go look at the bad variable, CUST-ACCT-BALANCE

Analyze an abend (9 of 20)

```
Associated Storage Areas                                    Line 1 Col 1 8
Command ===> f cust-acct-balance                           Scroll ===> PAG
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0

Task Global Table (TGT) at address 0003F5D0 for length 376

WORKING-STORAGE SECTION
     - Collapse hex
 Off Hex Value                          Data Value          Source (Starting
BLW=0000 at address 0003F7C8
                                                            01  WS-FIELDS.
   0 C3C1D3C3 E4D3C1E3 C9D5C740 C2C1D3C1 *CALCULATING BALA*      05  WS-PROGR
  10 D5C3C540 E2E3C1E3 E2404040 4040     *NCE STATS       *
  1E D5                                  *N               *      05  WS-FIRST
  1F 0000000C                            0                       05  WS-WORK-
  23 0000000C                            0                       05  WS-WORK-
  27 0000000C                            0                       05  WS-WORK-
  2B 0000000C                            0                       05  WS-WORK-
  2F 0000000C                            0                       05  WS-WORK-

LINKAGE SECTION
```

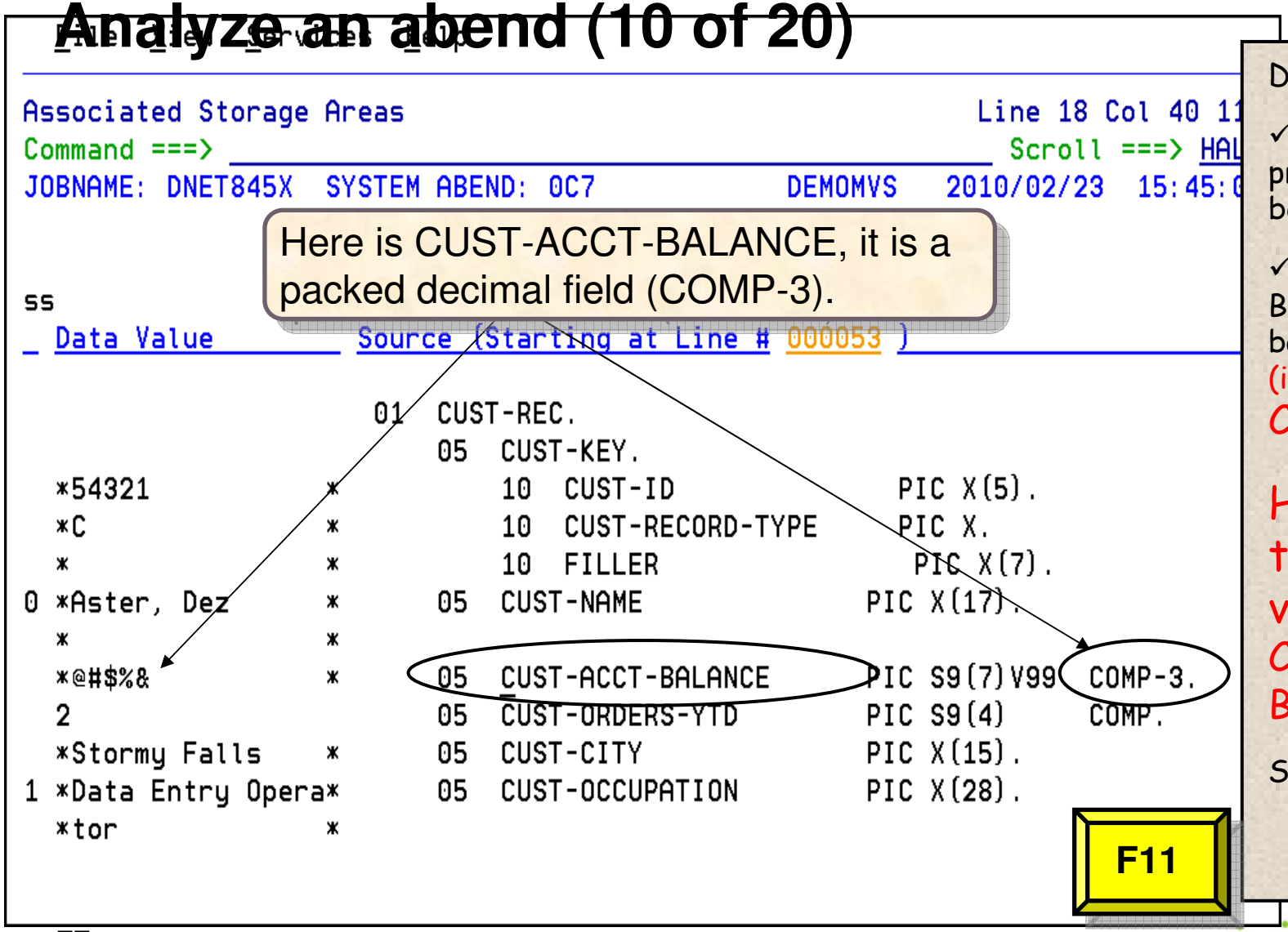
Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data

Go look at the bad variable, CUST-ACCT-BALANCE

Analyze an abend (10 of 20)



Debug Clues:

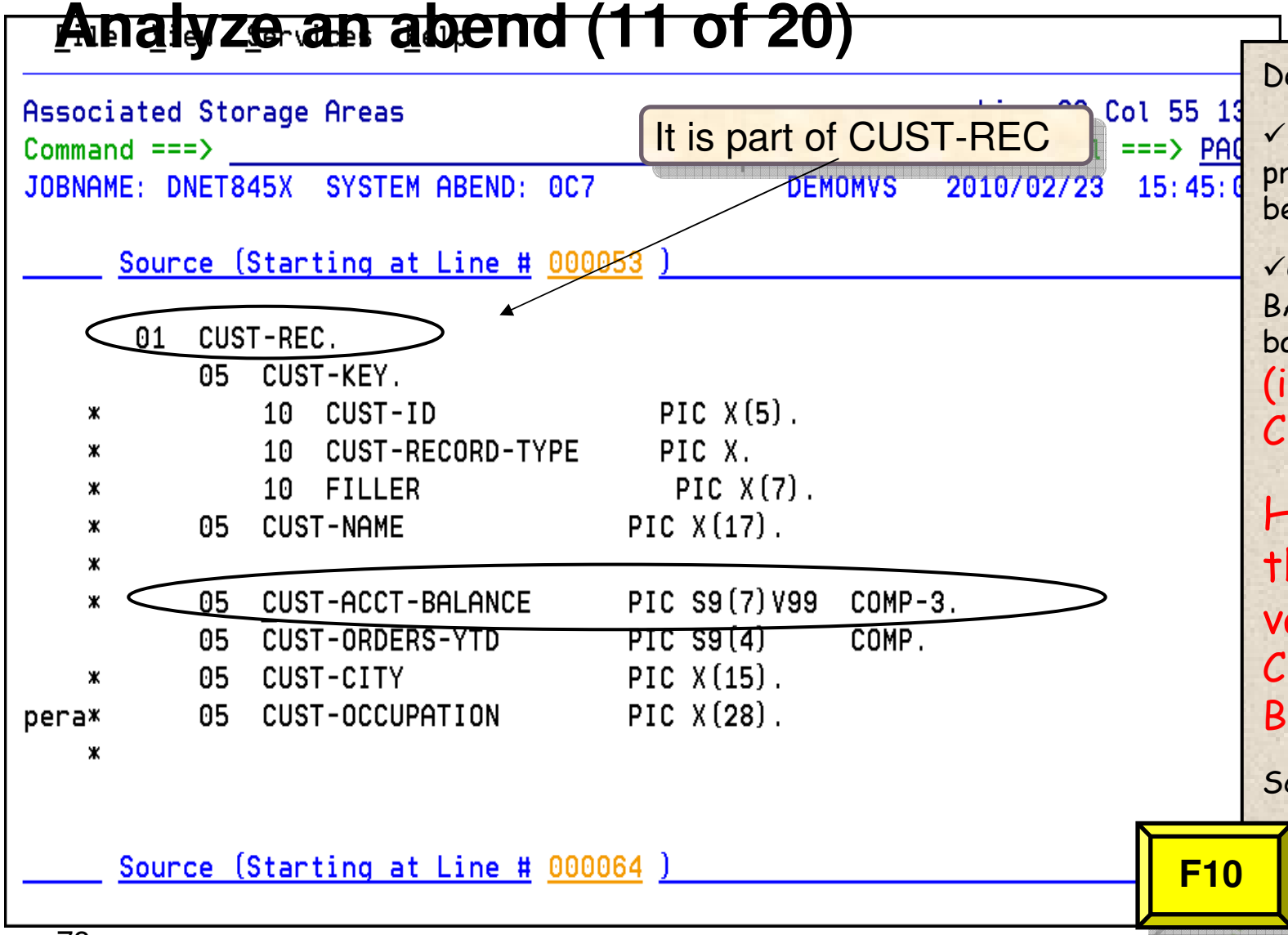
✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data (it is part of CUST-REC)

Here is the bad variable, CUST-ACCT-BALANCE

Scroll right

Analyze an abend (11 of 20)



It is part of CUST-REC

Debug Clues:

✓Abended in program SAM2 because:

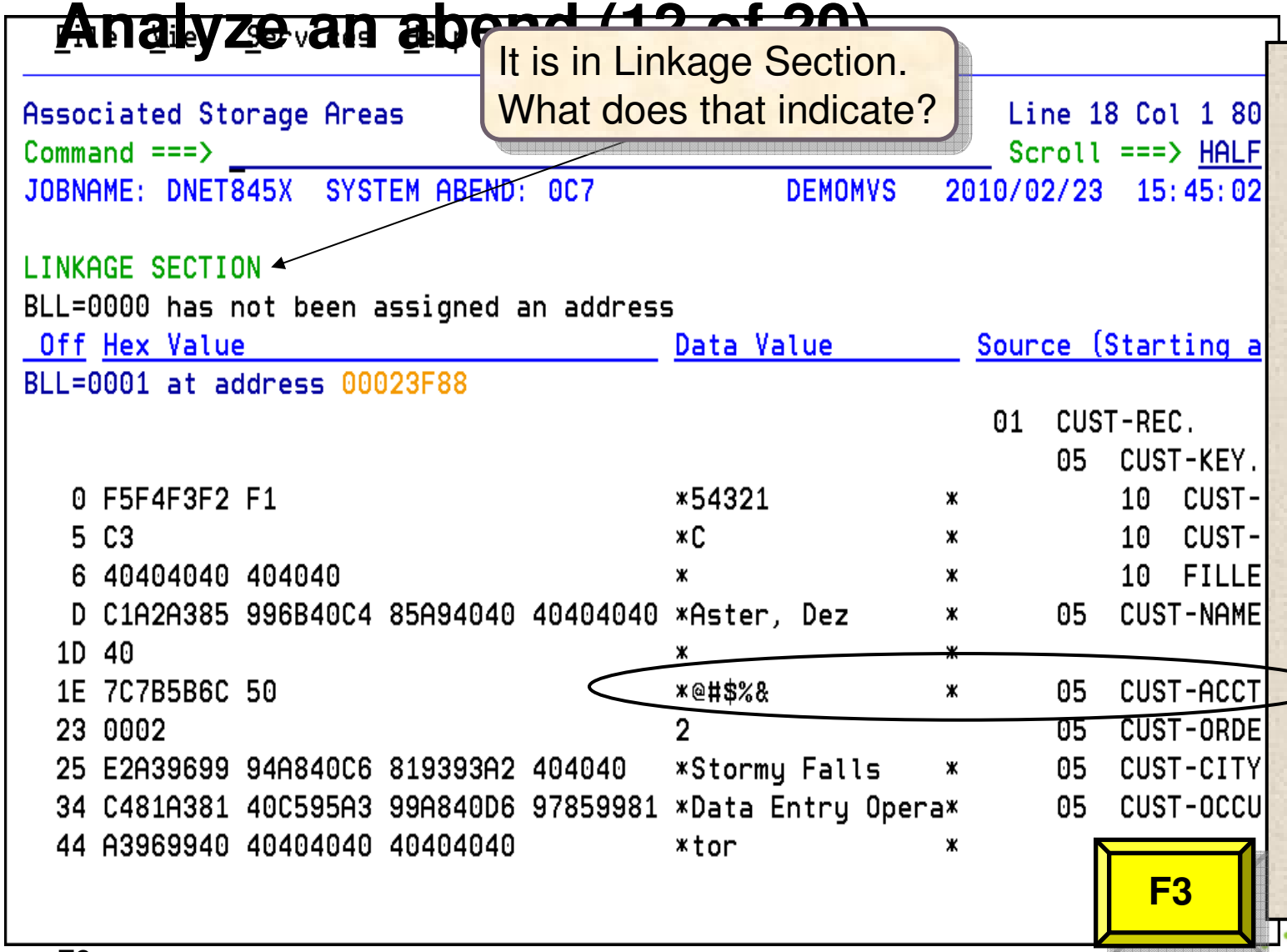
✓CUST-ACCT-BALANCE had bad numeric data (it is part of CUST-REC)

Here is the the bad variable, CUST-ACCT-BALANCE

Scroll left

F10

Analyze an abend (12 of 20)



Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE had bad numeric data (it is part of CUST-REC)

Here is the the bad variable, CUST-ACCT-BALANCE

Return to program detail

Analyze an abend (13 of 20)

```
Event 3 of 3: Abend S0C7 *** Point of Failure ***              Line 83 Col 1 8
Command ===> _____________________________________________  Scroll ===> HAL
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0
  R6: 205910C0 (606016 bytes of storage addressable)
  R7: 2050DA20 (1144288 bytes of storage addressable)
  R8: 0003F7C8 (Module SAM2 program SAM2 WORKING-STORAGE SECTION BLW=0000 +
               X'0', symbol WS-FIELDS, source line # 36 )
  R9: 0003F5D0 (Module SAM2 program SAM2 + X'5D0', source line # 116 )
  R10: 0003F11C (Module SAM2 program SAM2 + X'11C')
  R11: 0003F29C (Module SAM2 program SAM2 + X'29C')
  R12: 0003F0FC (Module SAM2 program SAM2 + X'FC')
  R13: 20594458 (592808 bytes of storage addressable)
  R14: 8003F392 (Module SAM2 program SAM2 + X'392', source line # 89 )
  R15: 8003F224 (Module SAM2 program SAM2 + X'224')

Associated Messages

CEE3207S The system detected a data exception (System Completion Code=0C7).

Associated Storage Areas

*** Bottom of data.
```

F3

Debug Clues:

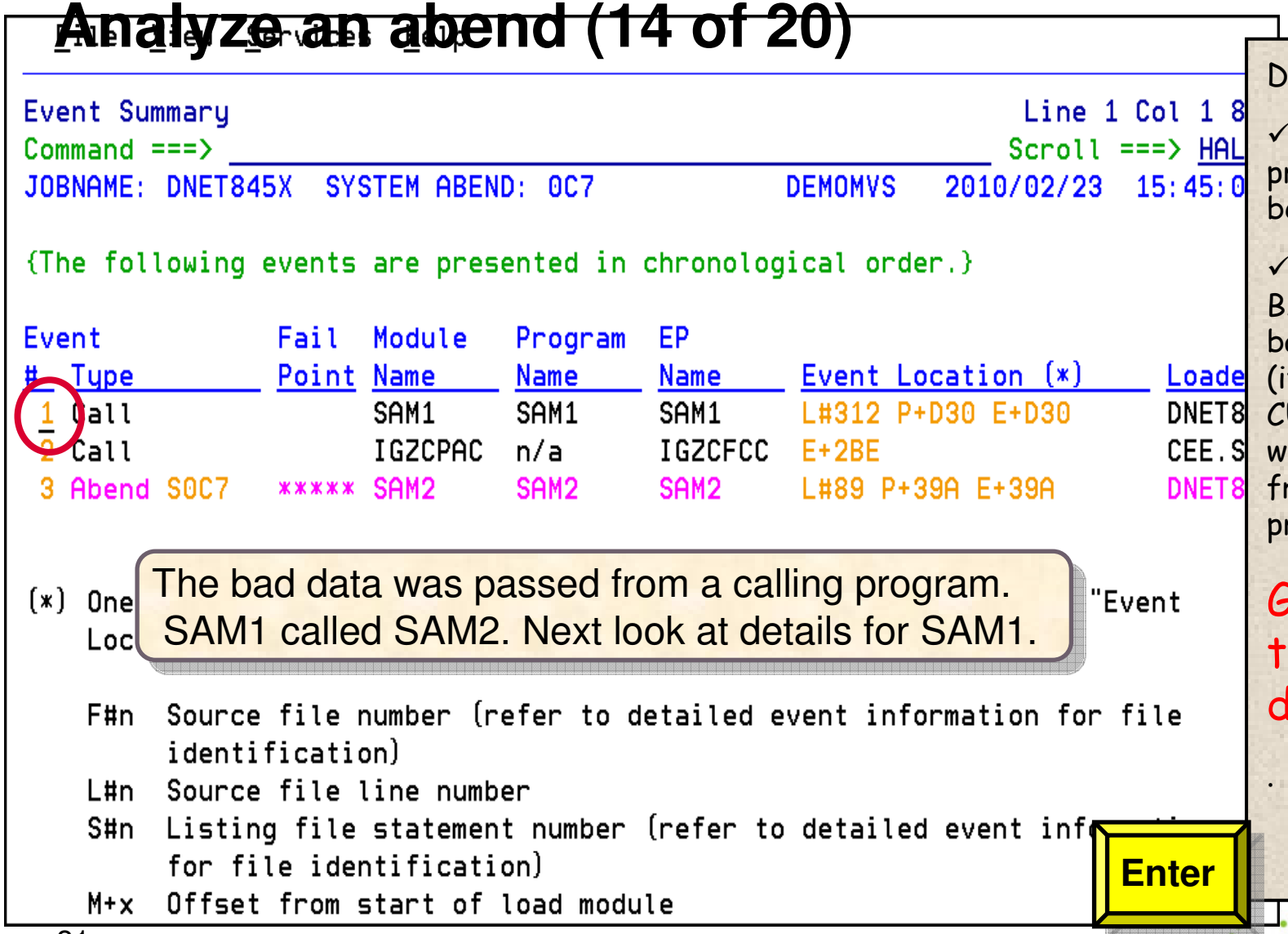
✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE has bad numeric data (it is part of CUST-REC), which was passed from a calling program

Go look at the passed data

Return to the events list

Analyze an abend (14 of 20)



The bad data was passed from a calling program. SAM1 called SAM2. Next look at details for SAM1.

Enter

Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE has bad numeric data (it is part of CUST-REC), which was passed from a calling program

Go look at the passed data

.

Analyze an abend (15 of 20)

```
Event 1 of 3: Call (DSA Address 2059                    1 Col 1 8
Command ===> bottom                                     ll ===> HAL
JOBNAME: DNE1845X  SYSTEM ABEND: 0C7                    23  15:45:0


The source code below was executed via the following sequence of PERFORM
statements:
  Source
  Line #
  000261              PERFORM 100-PROCESS-TRANSACTIONS
  000278                        PERFORM 200-PROCESS-PRINT-TRAN
  000299              PERFORM 210-PROCESS-CUSTFILE-RECORD

COBOL Source Code:
  Source
  Line #
  000311        *       SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
  000312                CALL 'SAM2' USING CUST-REC,
  000313                       CUSTOMER-BALANCE-STATS

Data Field Declarations:
  Source
```

What passed variable contained the bad data?

This is the CALL statement

Enter

Debug Clues:

- ✓Abended in program SAM2 because:
- ✓CUST-ACCT-BALANCE has bad numeric data (it is part of CUST-REC), which was passed from a calling program
- ✓SAM1 called SAM2

Go look at the passed data

Analyze an abend (16 of 20)

```
Event 1 of 3: Call (DSA Address 20594030)                    Line 68 Col 1
Command ===> _____________________________________________  Scroll ===> P
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45
  R10: 00007124 (Module SAM1 program SAM1 + X'124')
  R11: 00007798 (Module SAM1 program SAM1 + X'798')
  R12: 000070FC (Module SAM1 program SAM1 + X'FC')
  R13: 20594030 (593872 bytes of storage addressable)
  R14: 80007D32 (Module SAM1 program SAM1 + X'D32', source line # 312 )
  R15: A05142B0 (Module IGZCPAC + X'2B0')

Associated Open Files

File Name . . . . . : CUSTFILE
File Name . . . . . : CUSTRPT
File Name . . . . . : TRANFILE

Associated Storage Areas

Next Event Details

*** Bottom of data.
```

Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE has bad numeric data, which was passed from a calling program

✓SAM1 called SAM2

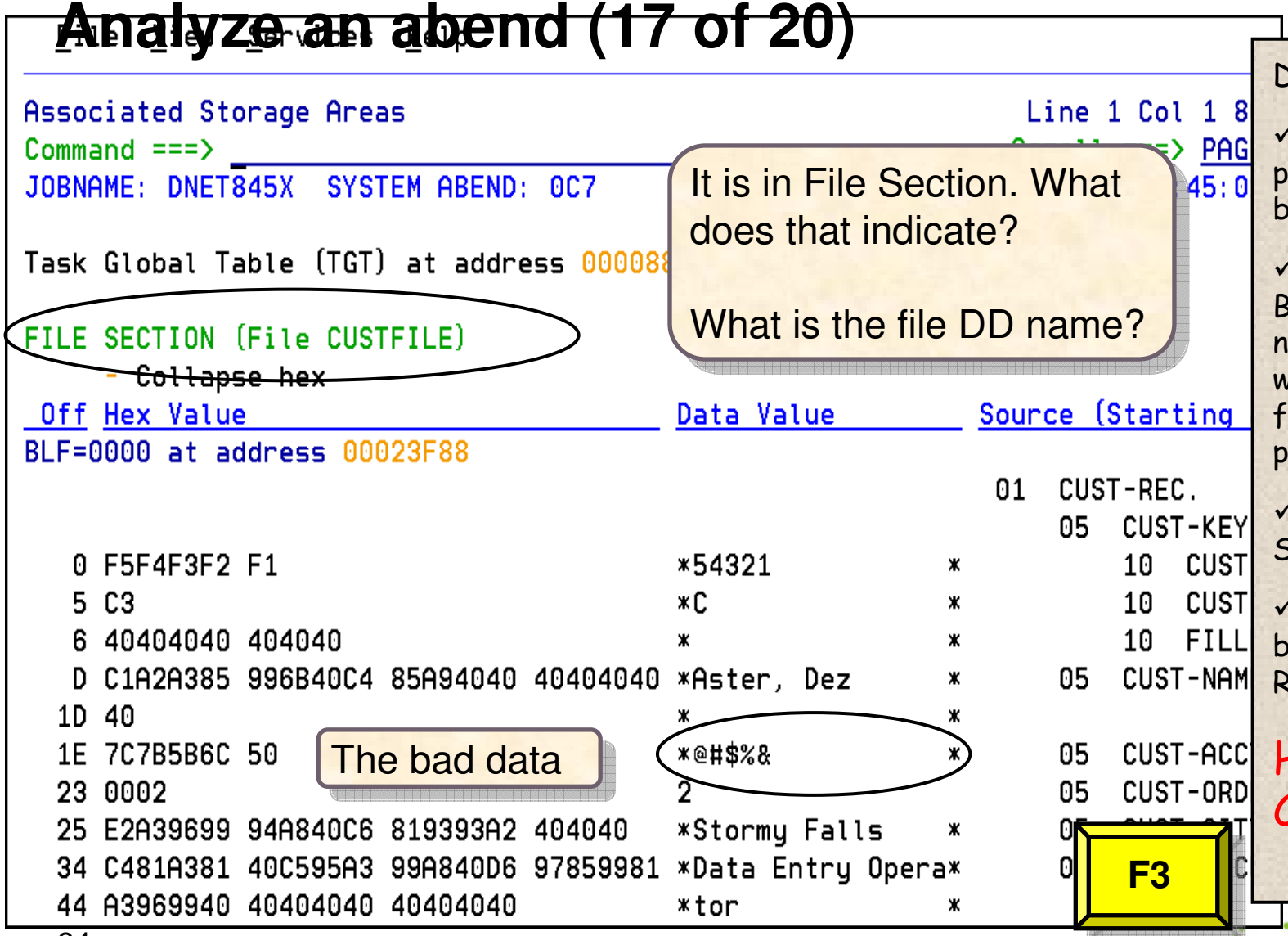
✓SAM1 passed bad data in CUST-REC

Look at CUST-REC

Go to variables and storage for this program

Enter

Analyze an abend (17 of 20)



It is in File Section. What does that indicate?

What is the file DD name?

The bad data

Debug Clues:

✓Abended in program SAM2 because:

✓CUST-ACCT-BALANCE has bad numeric data, which was passed from a calling program

✓SAM1 called SAM2

✓SAM1 passed bad data in CUST-REC

Here is CUST-REC

Analyze an abend (18 of 20)

```
Event 1 of 3: Call (DSA Address 20594030)                    Line 68 Col 1 8
Command ===> _____________________________________________  Scroll ===> PAG
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0
  R10: 00007124 (Module SAM1 program SAM1 + X'124')
  R11: 00007798 (Module SAM1 program SAM1 + X'798')
  R12: 000070FC (Module SAM1 program SAM1 + X'FC')
  R13: 20594030 (593872 bytes of storage addressable)
  R14: 80007D32 (Module SAM1 program SAM1 + X'D32', source line # 312 )
  R15: A05142B0 (Module IGZCPAC + X'2B0')

Associated Open Files

File Name . . . . . : CUSTFILE
File Name . . . . . : CUSTRPT
File Name . . . . . : TRANFILE

Associated Storage Areas


Next Event Details

*** Bottom of data.
```

Cursor select the file name

Enter

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE has bad numeric data, which was passed from a calling program
- ✓ SAM1 called SAM2
- ✓ SAM1 passed bad data in CUST-REC
- ✓ The bad data was read from file CUSTFILE

Analyze an abend (19 of 20)

What is the full name of the CUSTFILE file?

```
File Information                                          Line 1 Col 1 8
Command ===> _                                          Scroll ===> PAG
JOBNAME: DNET845X  SYSTEM ABEND: 0C7          DEMOMVS   2010/02/23  15:45:0

File Name . . . . . : CUSTFILE
  Data Set Name . . . . . : DNET845.ADLAB.FILES(CUST2FA)
  File Attributes . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=SEQUENTIAL
                            RECFM=FIXED
  Last I/O Function . . . : READ
  Open Status . . . . . : INPUT
  File Status Code. . . . : 0

  Previous Record . . . . : Record data length 80
  Address  Offset   Hex                                      EBCDIC
  00023F38          F2F4F0F9 F0D7D7D6 F0F0F9F4 F5D78981 *24090PP000945Pia*
  00023F48      +10 95964040 40404040 40404040 40404040 *no              *
  00023F58      +20 40404040 4040F2F0 F0F560F0 F760F0F5 *      2005-07-05*
  00023F68      +30 0001F2F0 F0F660F1 F260F2F7 40404040 *..2006-12-27    *
  00023F78      +40 40404040 40404040 40404040 40404040 *                *

  Current Record. . . . . : Record data length 80
  Address  Offset   Hex                                      EBCDIC
```

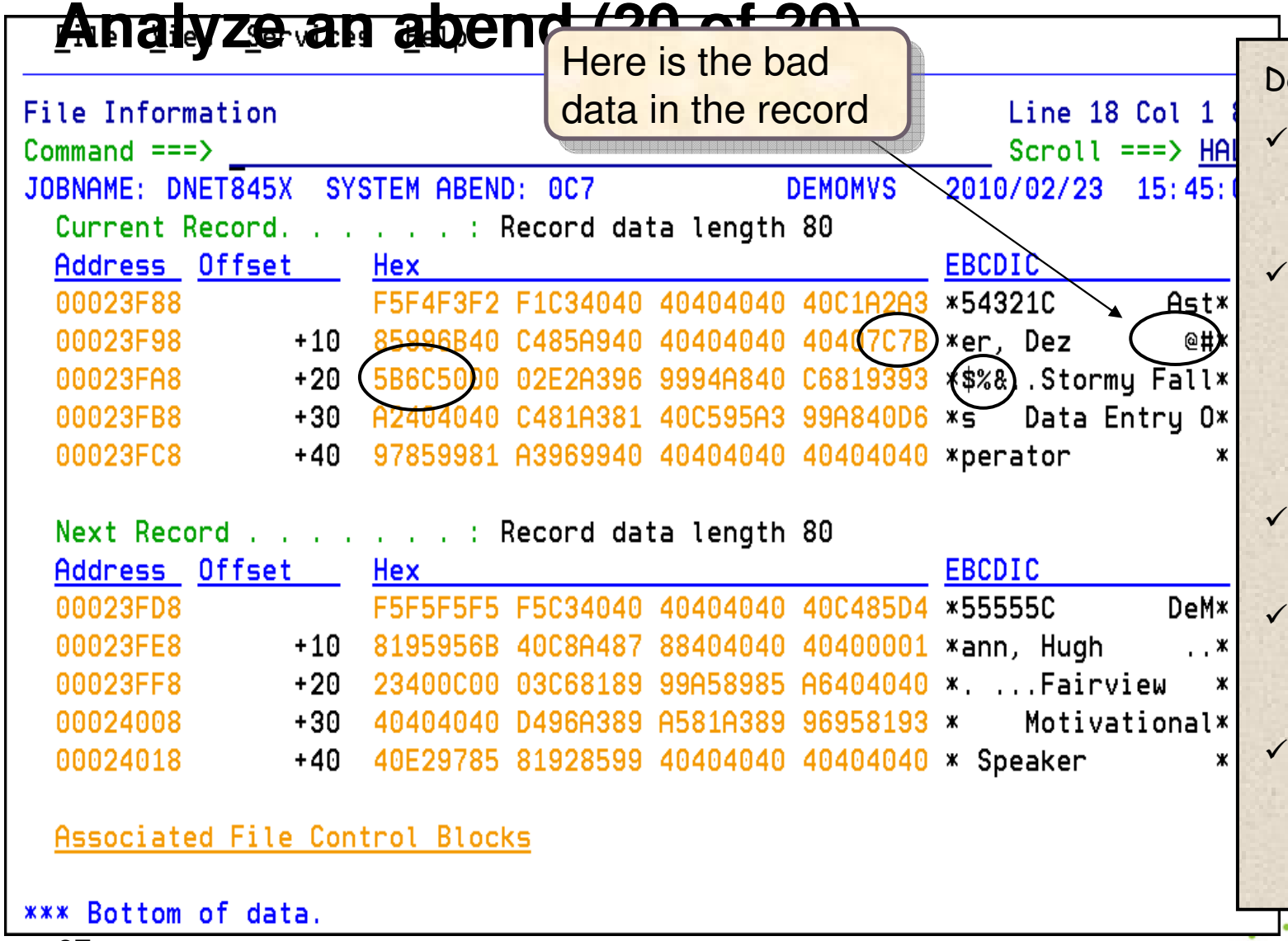
CUSTFILE is an input file

F8

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE has bad numeric data, which was passed from a calling program
- ✓ SAM1 called SAM2
- ✓ SAM1 passed bad data in CUST-REC
- ✓ The bad data was read from file CUSTFILE

Analyze an abend (20 of 20)



Here is the bad data in the record

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE has bad numeric data, which was passed from a calling program
- ✓ SAM1 called SAM2
- ✓ SAM1 passed bad data in CUST-REC
- ✓ The bad data was read from file CUSTFILE

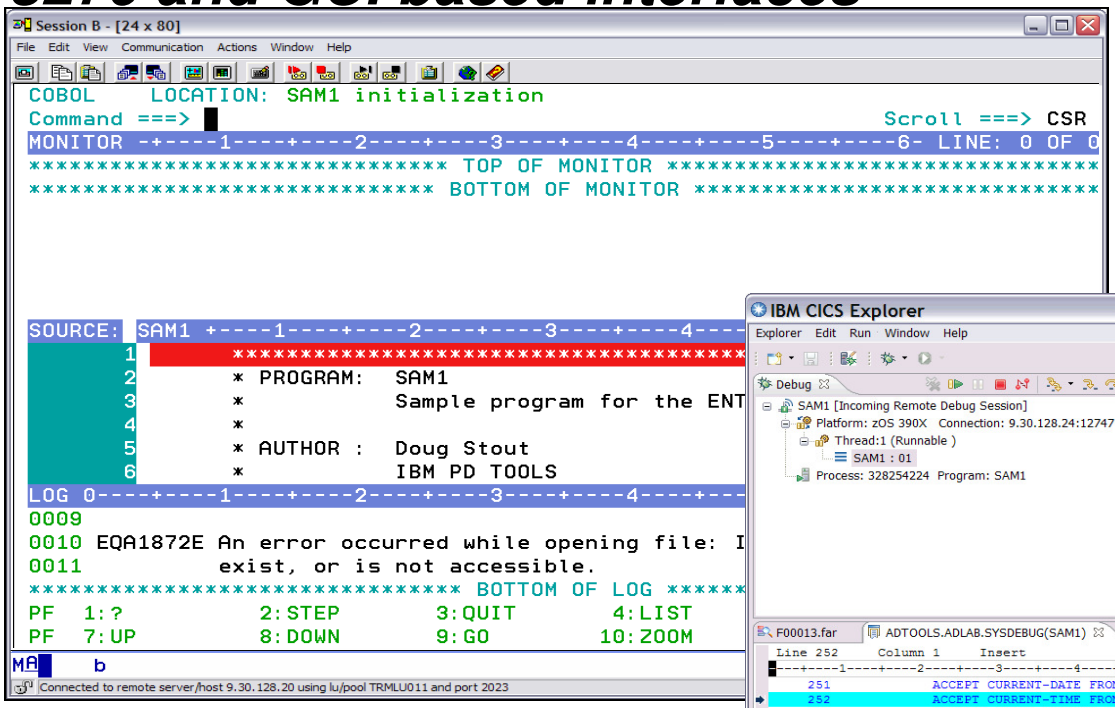
Fault Analyzer: What's new in version 11?

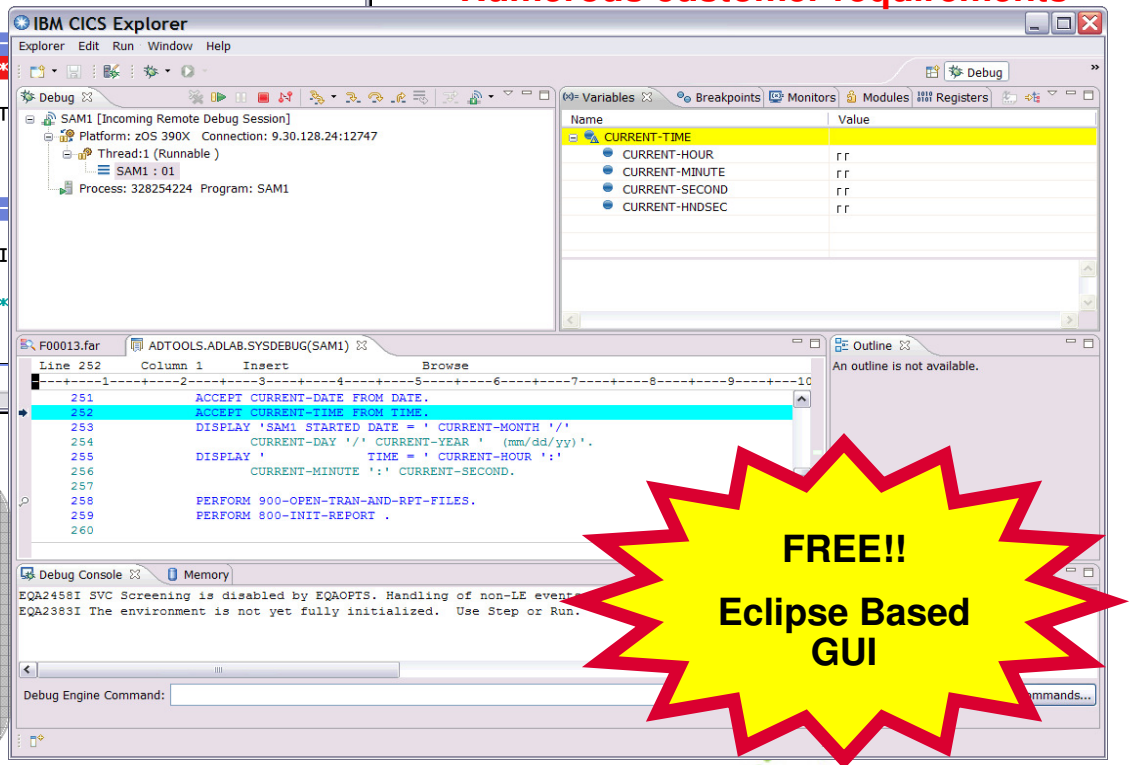
- The IBM Fault Analyzer Plug-in for Eclipse, when integrated with Fault Analyzer for z/OS and CICS Explorer, provides access to problem reports for diagnosing mainframe application errors and abends. Key features include:
  - An interface to manage views and multiple fault history files
  - The ability to browse fault entries that were created during real-time analysisof abending programs
  - A browser for browsing the dump storage associated with a fault entry
  - A source listing of abending programs using side files
- Java support enhancements:
  - Integration of Java stack trace information into the Fault Analyzer event list
  - Java source support where Java source is included in the abending jar file
- C/C++ Dwarf file support.
- Latest Enterprise PL/I sysdebug file support.
- Fault Entry size management improvements:
  - Fault Analyzer will no longer include un-referenced storage pages in the minidump portion of a Fault Entry.
  - An option to control the size of the CICS trace table included from SDUMP analysis.
  - Support for CICS EXCI calls made from the Fault Analyzer listing exit.
  - CICS Auxiliary trace data set interpretation and viewing.
  - Enhancements to ISPF history file management operations to be similar to IDIUTIL.

Agenda

- Fault Analyzer
- Debug Tool
- Application Performance Analyzer
- File Manager

IBM Debug Tool
3270 and GUI based interfaces





New in DT V11.1

- Plug-in for CICS Explorer
- Explicit debug mode
- New UI for Terminal Interface Manager
- IBM zEnterprise 196 support
- Numerous customer requirements

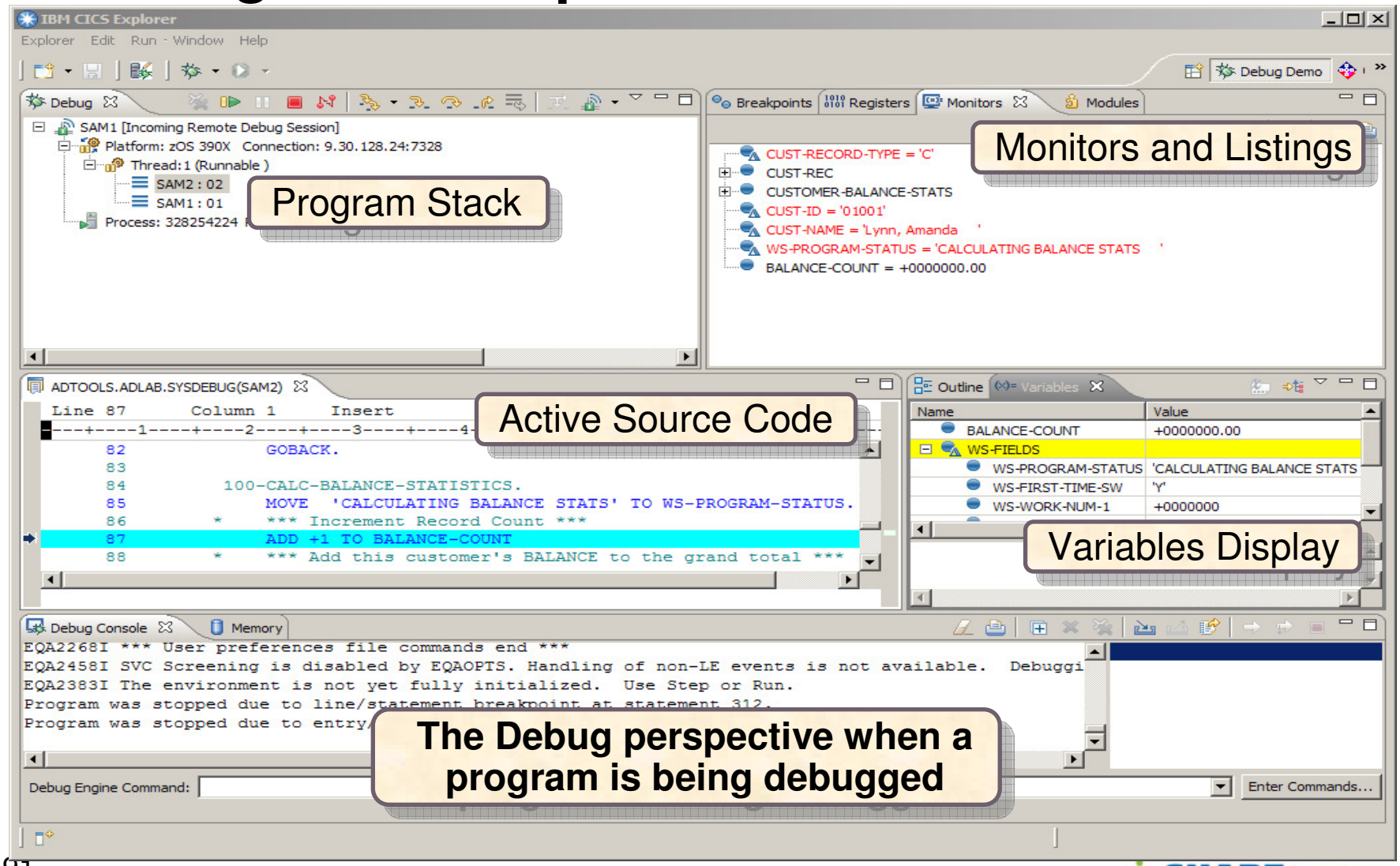
Sample Features

- 64-bit register support Assembler
- Dynamic patching
- Save and restore sessions settings
- Object level disassembly debugging

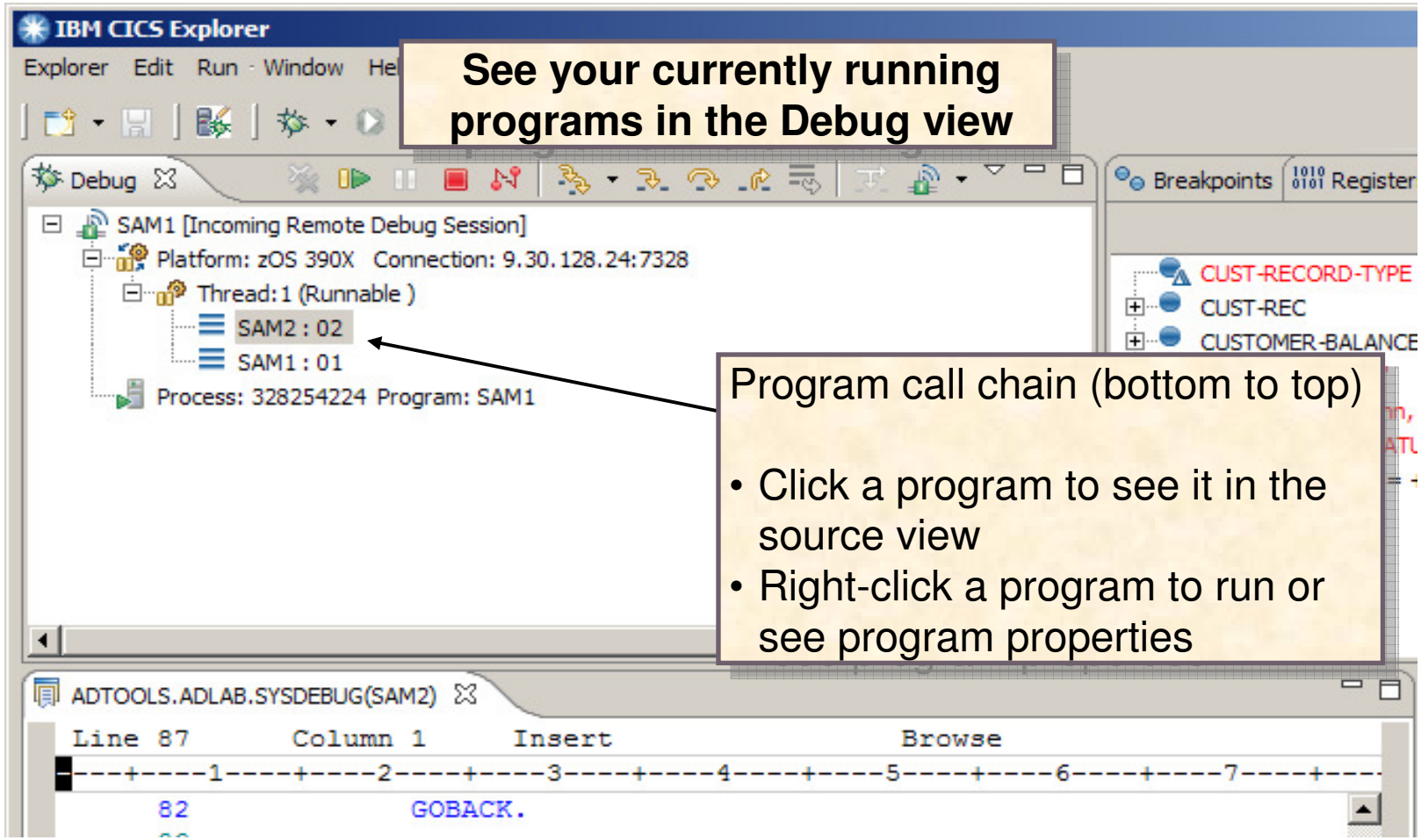
FREE!!

Eclipse Based GUI

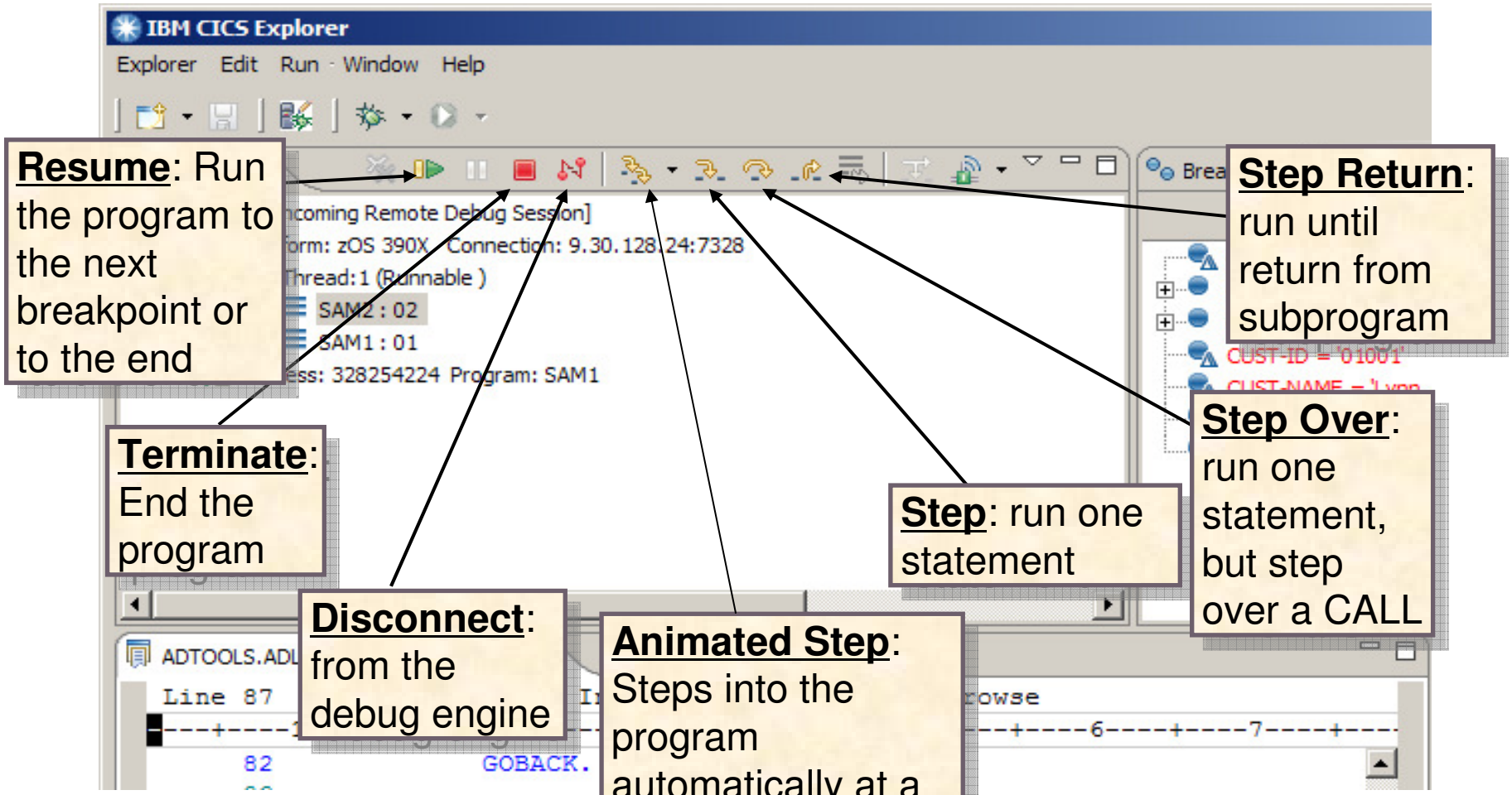
The Debug Tool Perspective



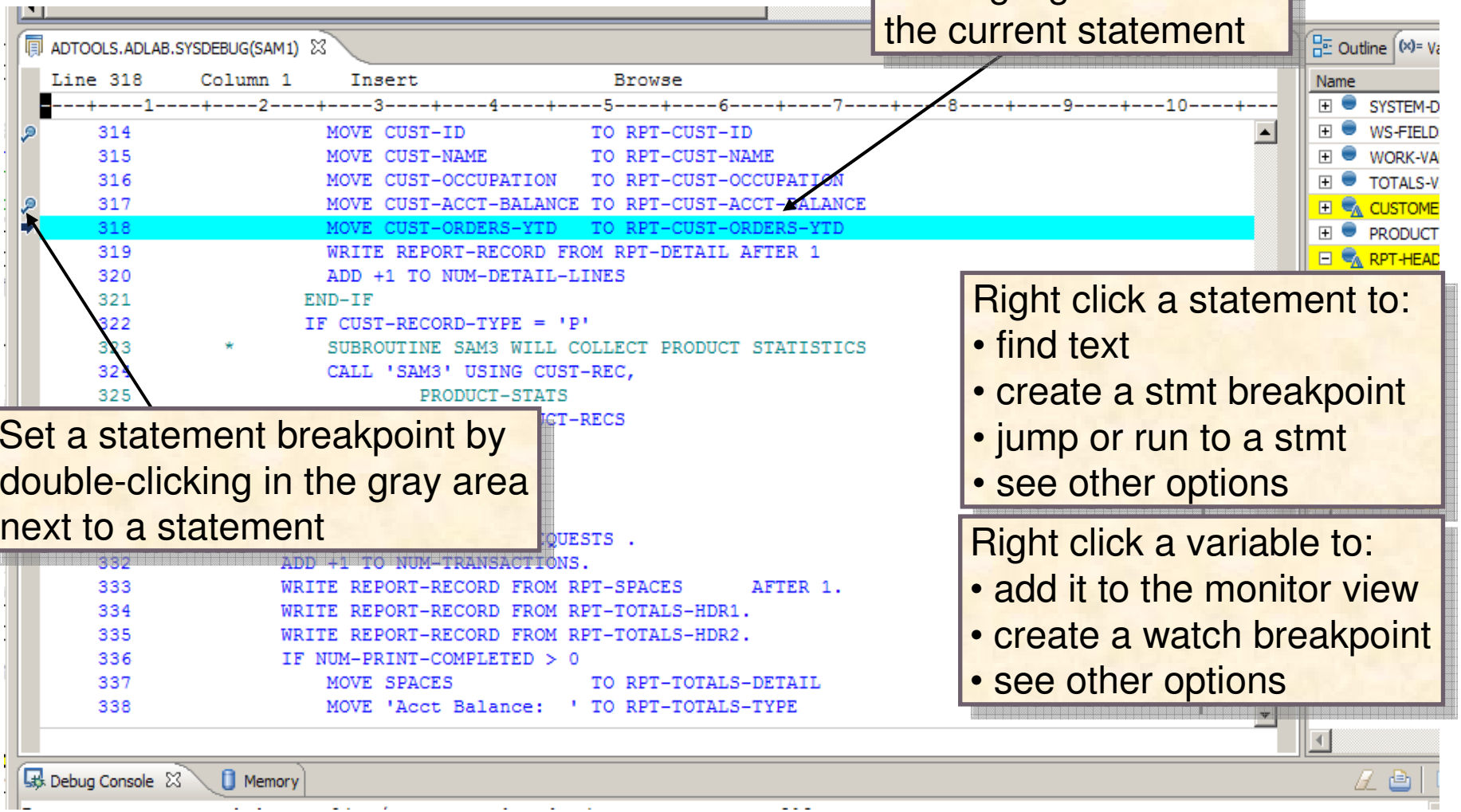
The Debug view



Action bar buttons perform program actions….

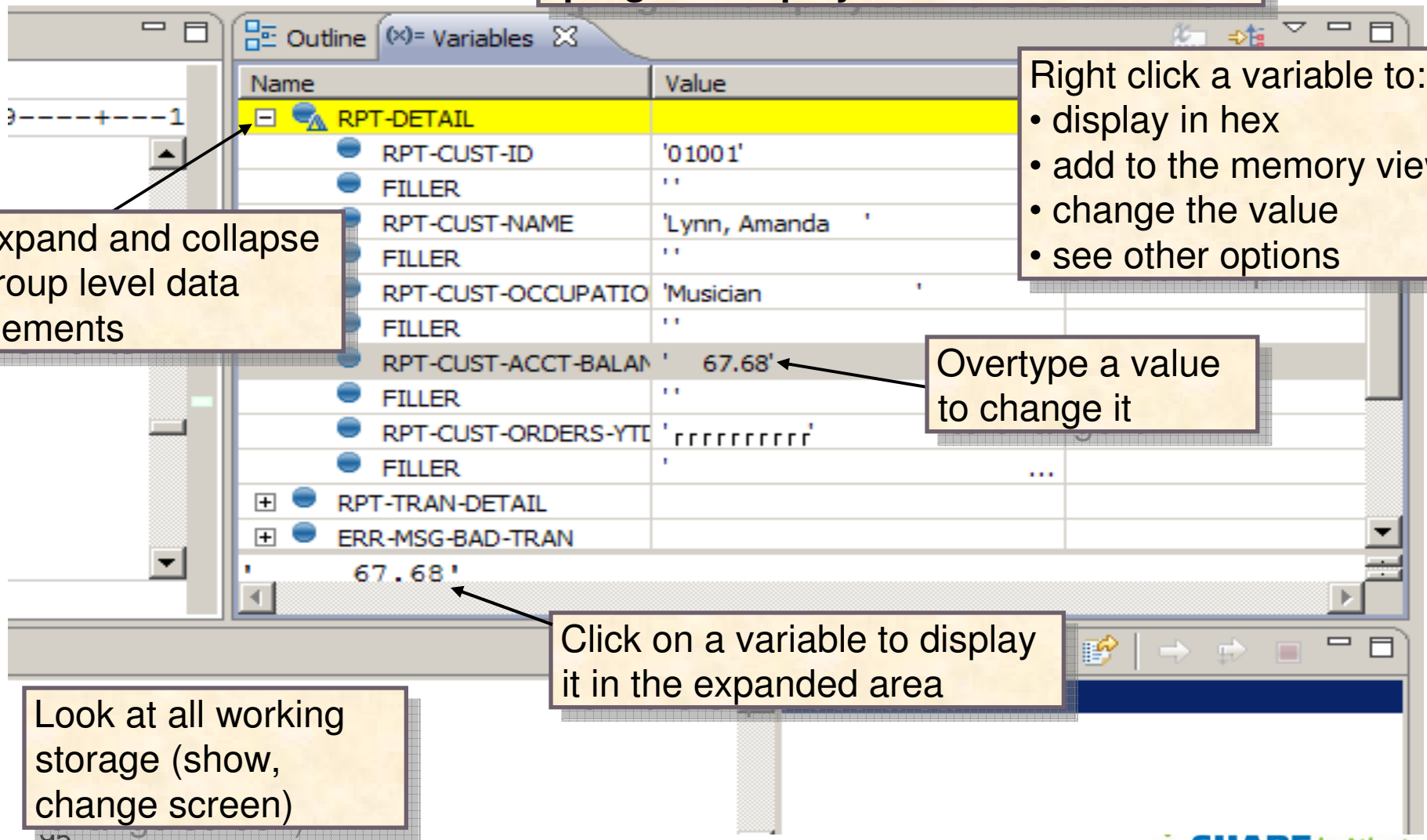


The Program Source view



The Variables View

View and change variables for the program displayed in the source view.



Expand and collapse group level data elements

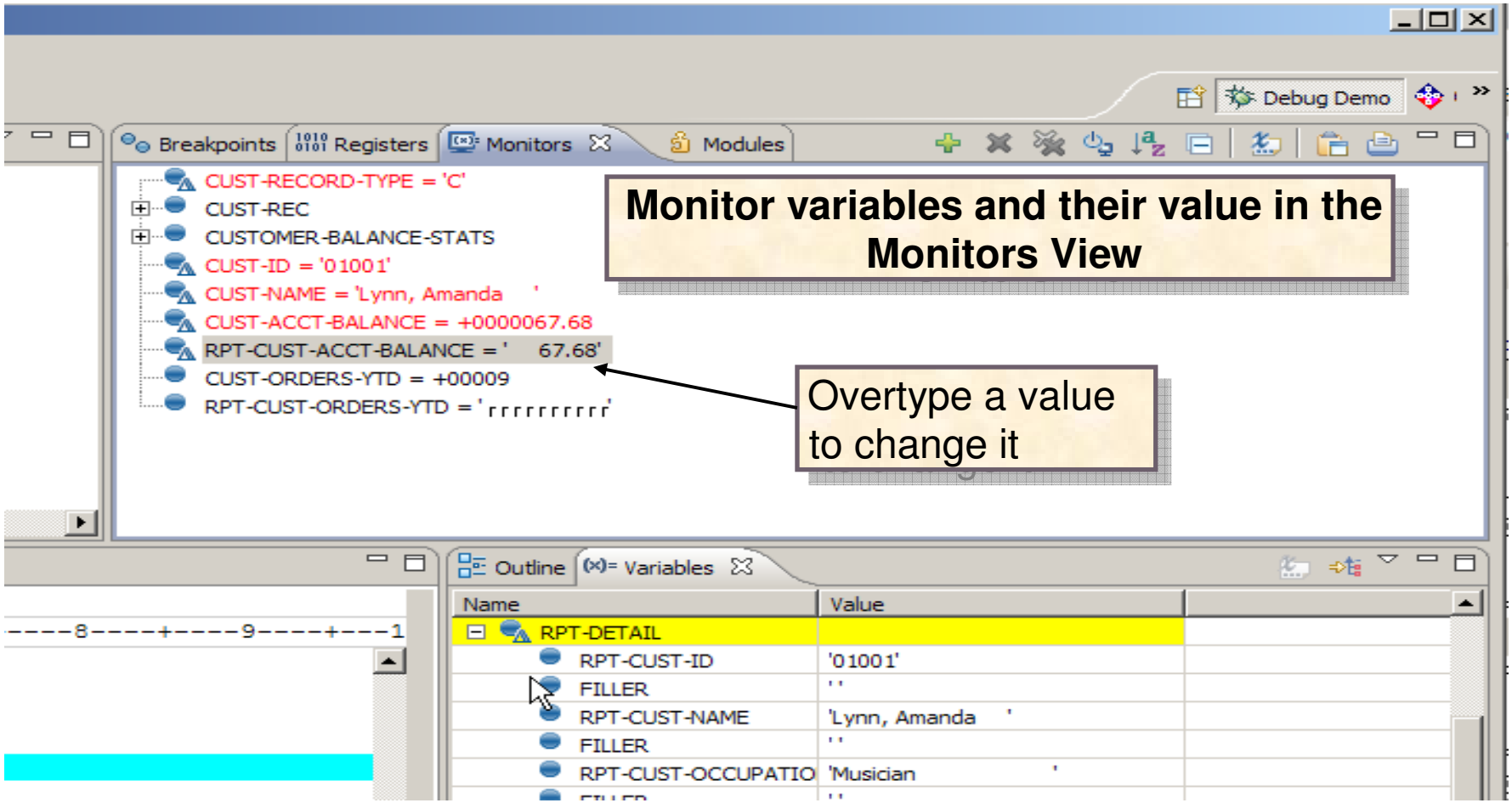
Right click a variable to:
- display in hex
- add to the memory view
- change the value
- see other options

Overtype a value to change it

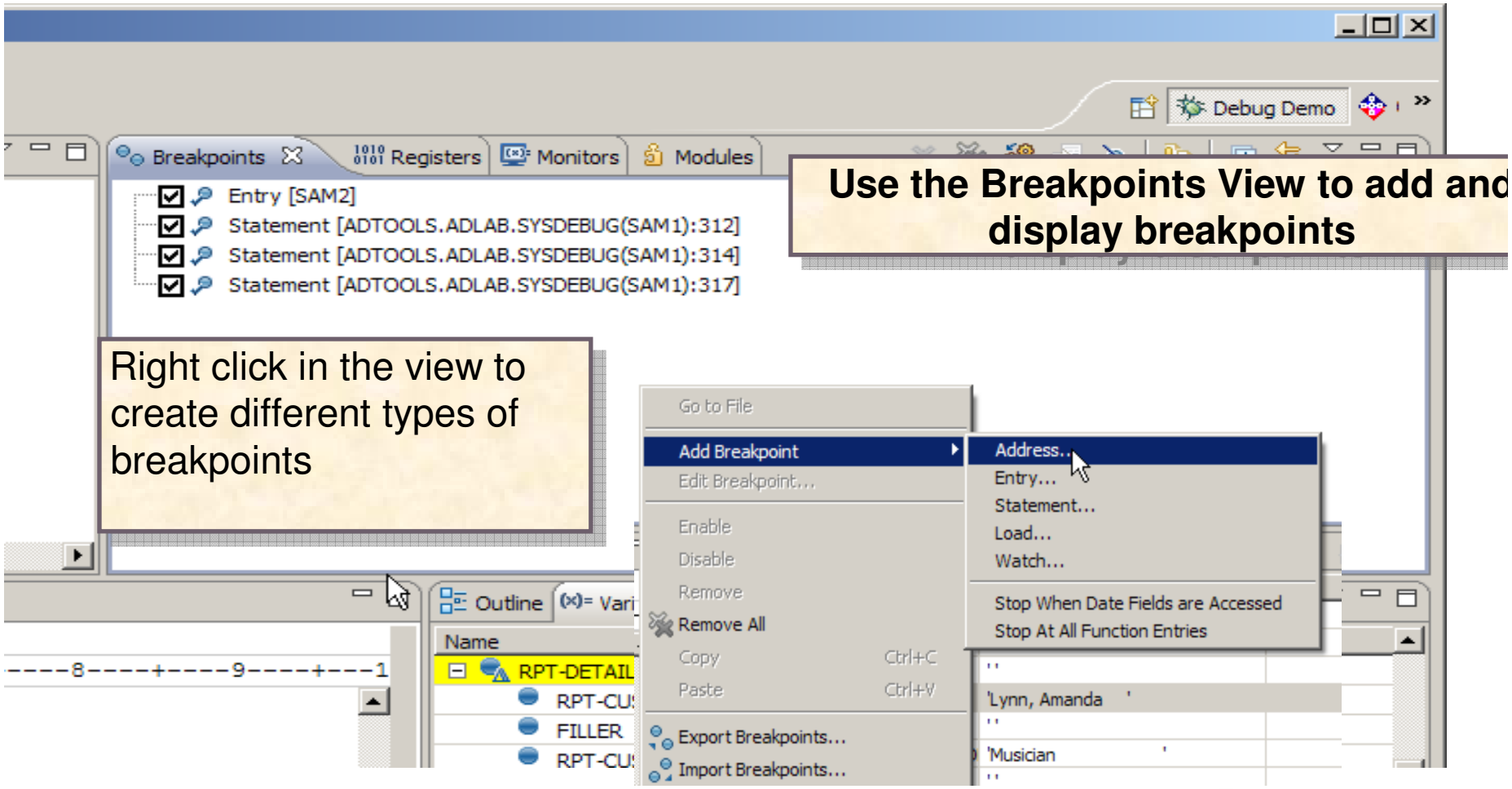
Click on a variable to display it in the expanded area

Look at all working storage (show, change screen)

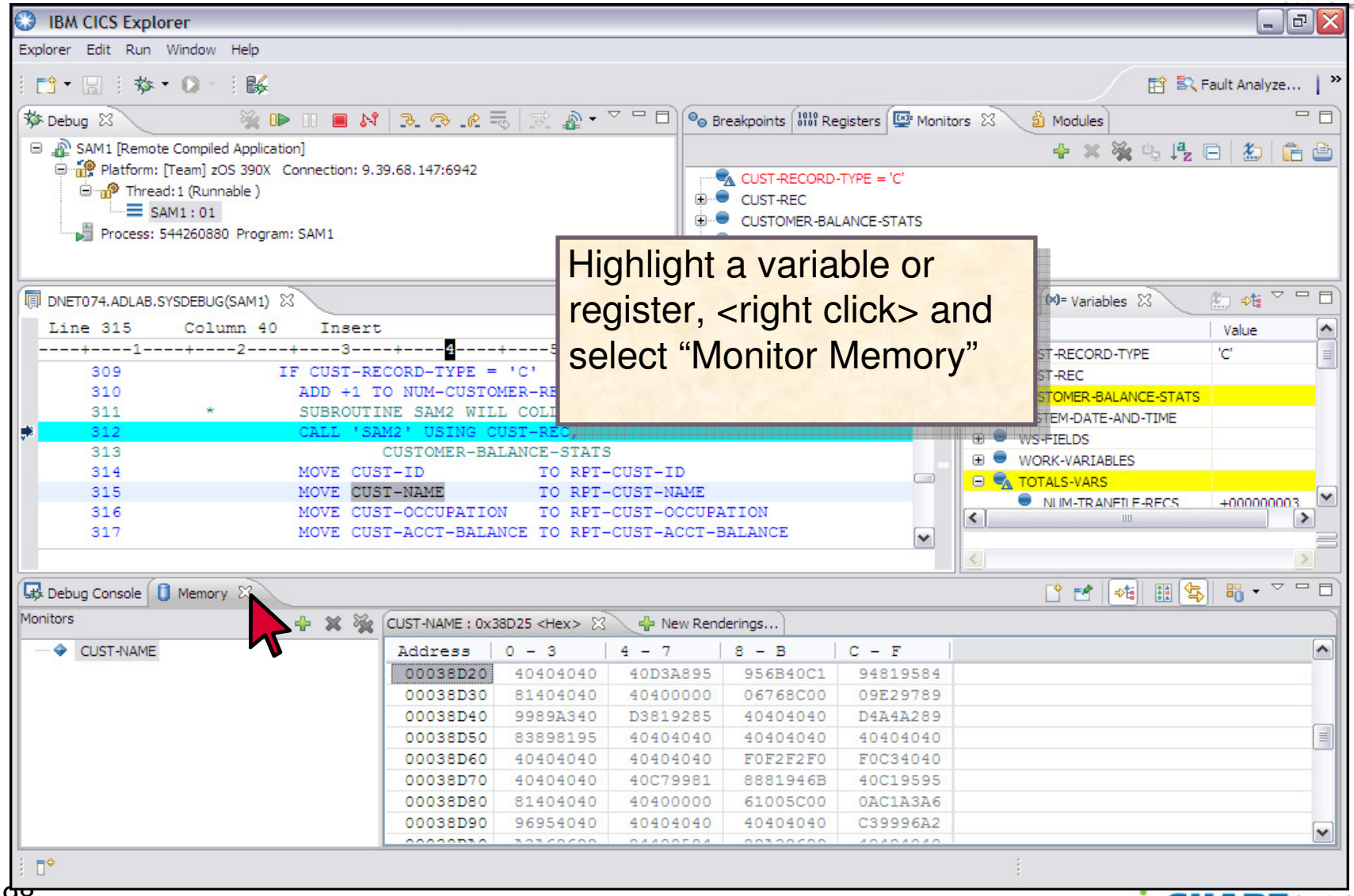
The Monitors View



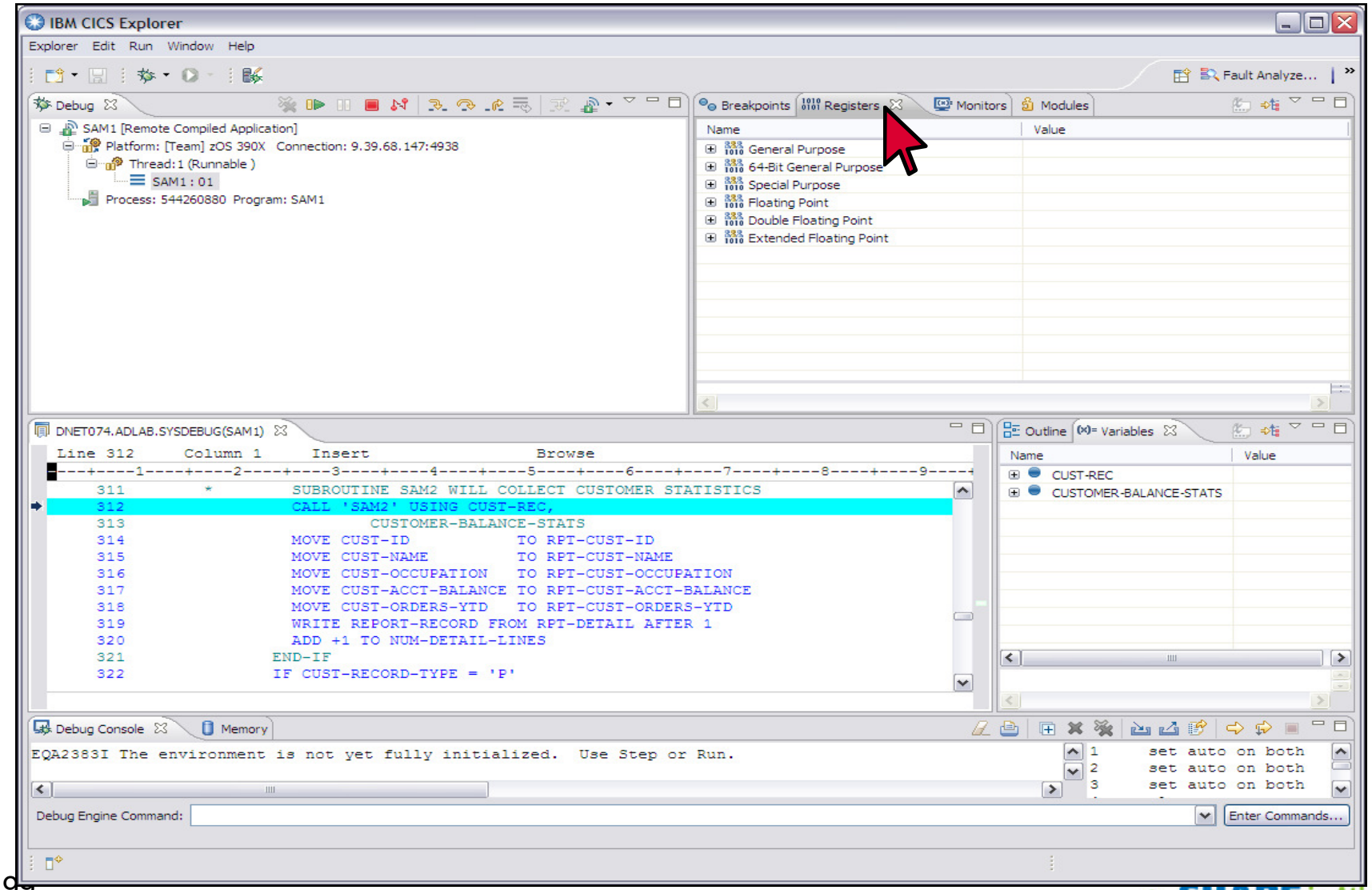
The Breakpoints View



The memory view



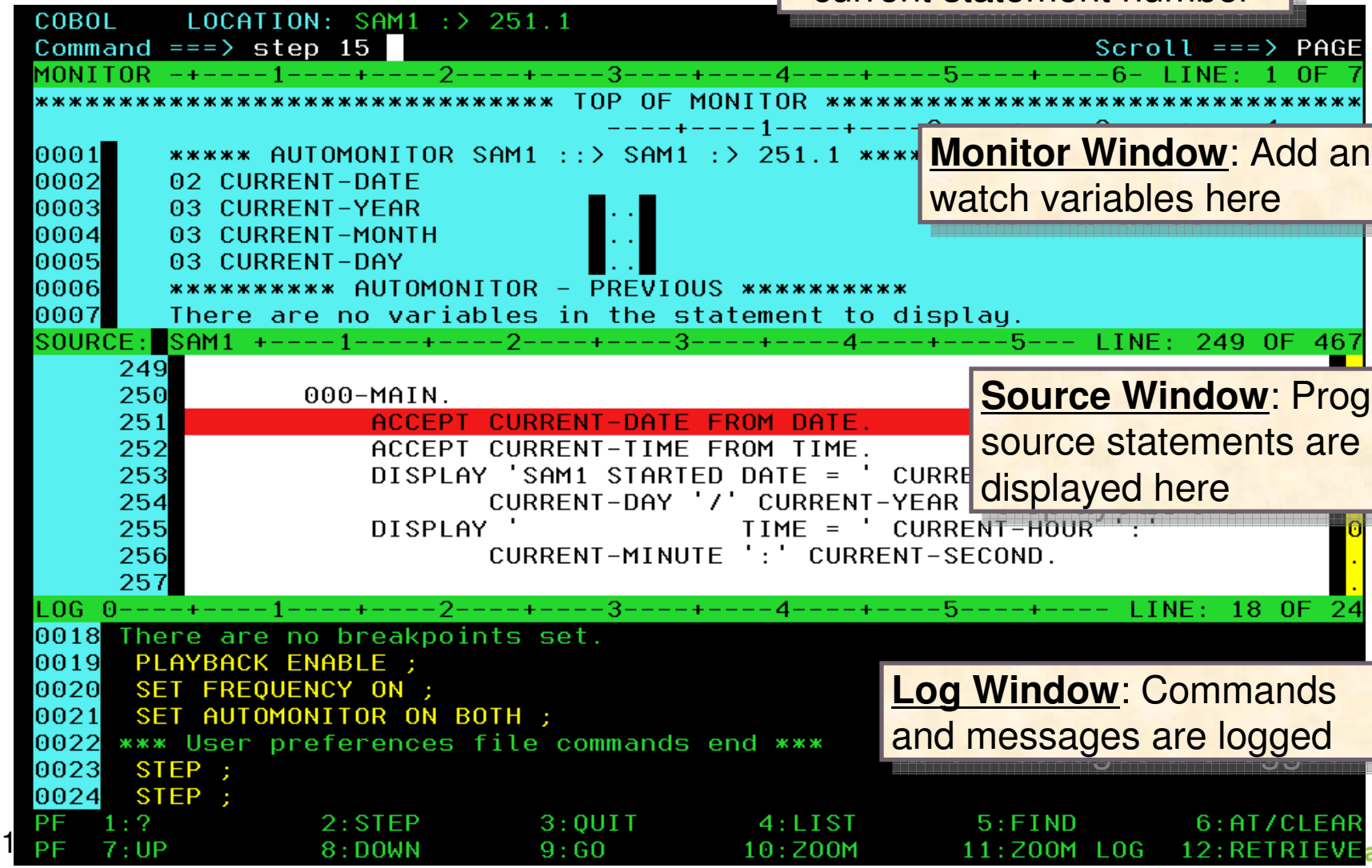
The registers view



The Debug Tool MFI Interface

Three windows in initial dis

Header: Shows:
- the name of program
- current statement number

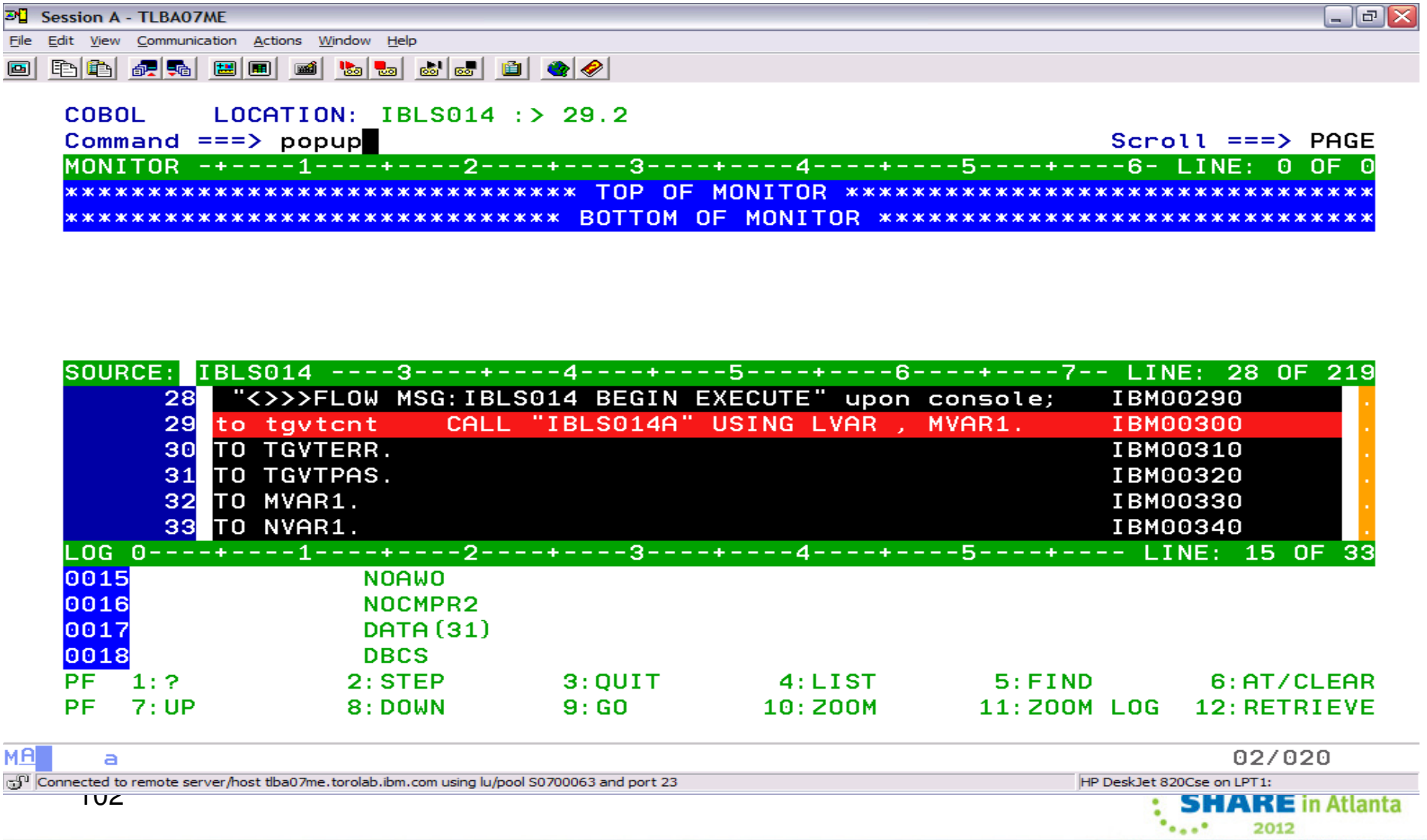


Monitor Window: Add and watch variables here

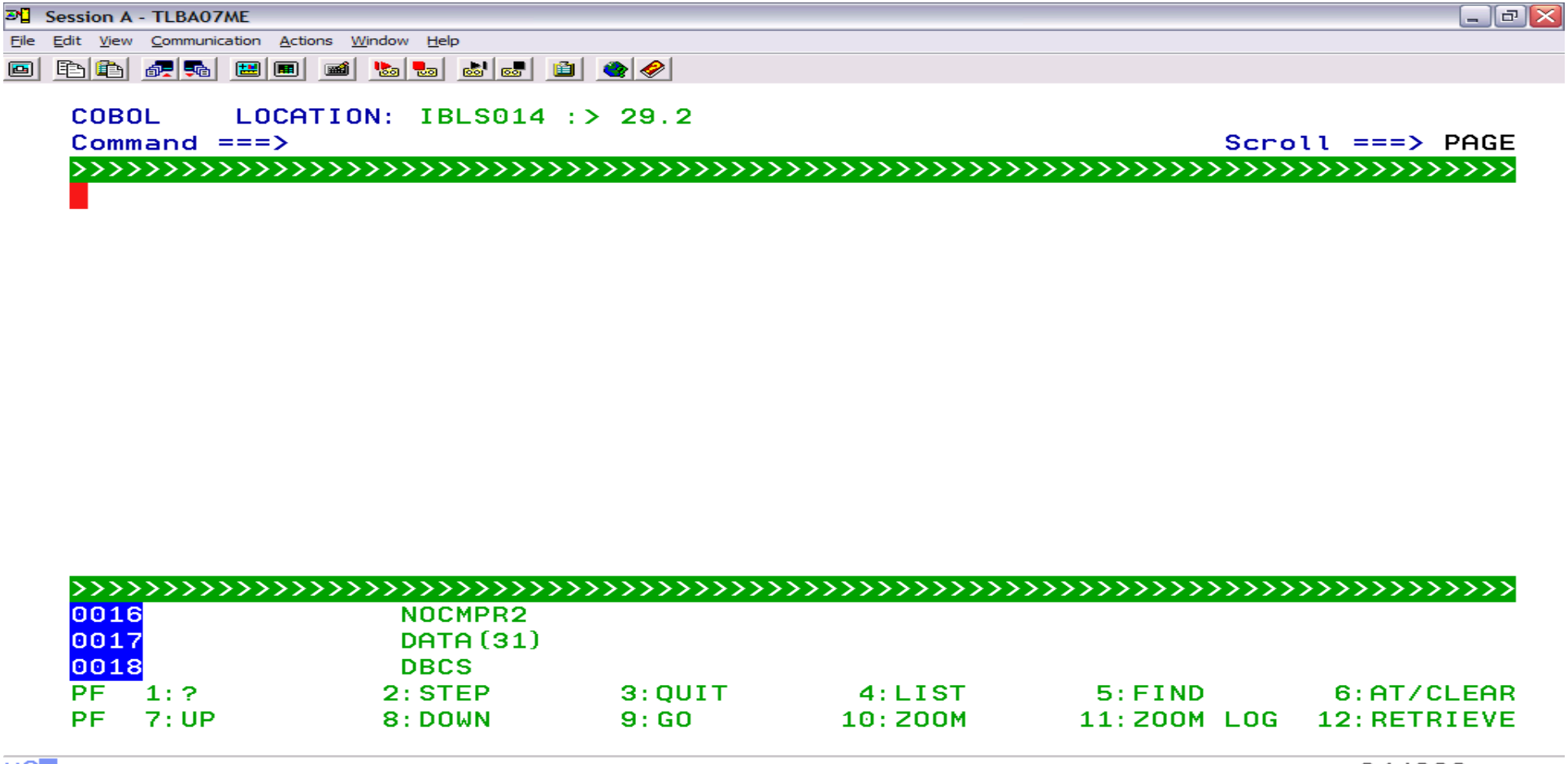
Source Window: Program source statements are displayed here

Log Window: Commands and messages are logged

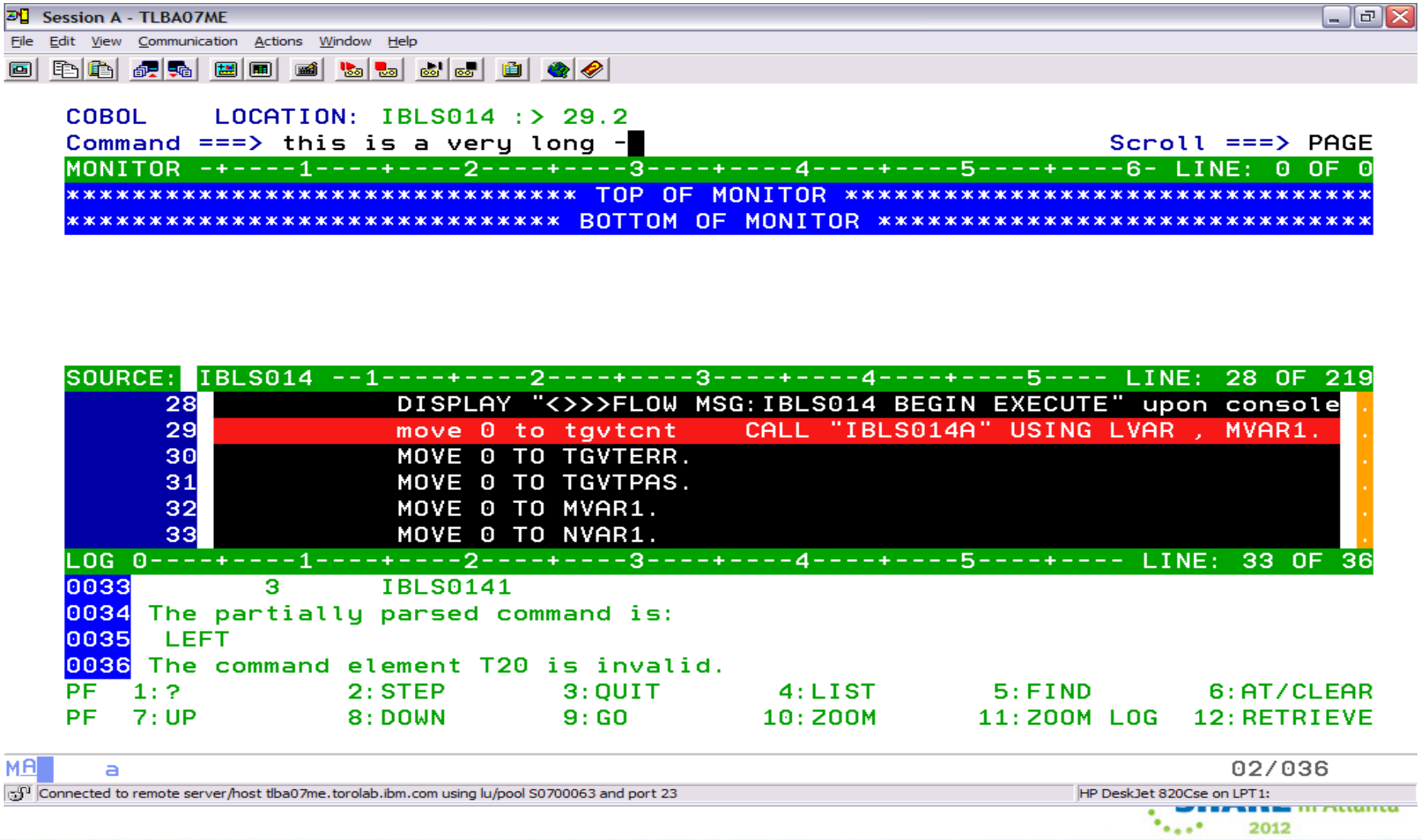
POPUP Command



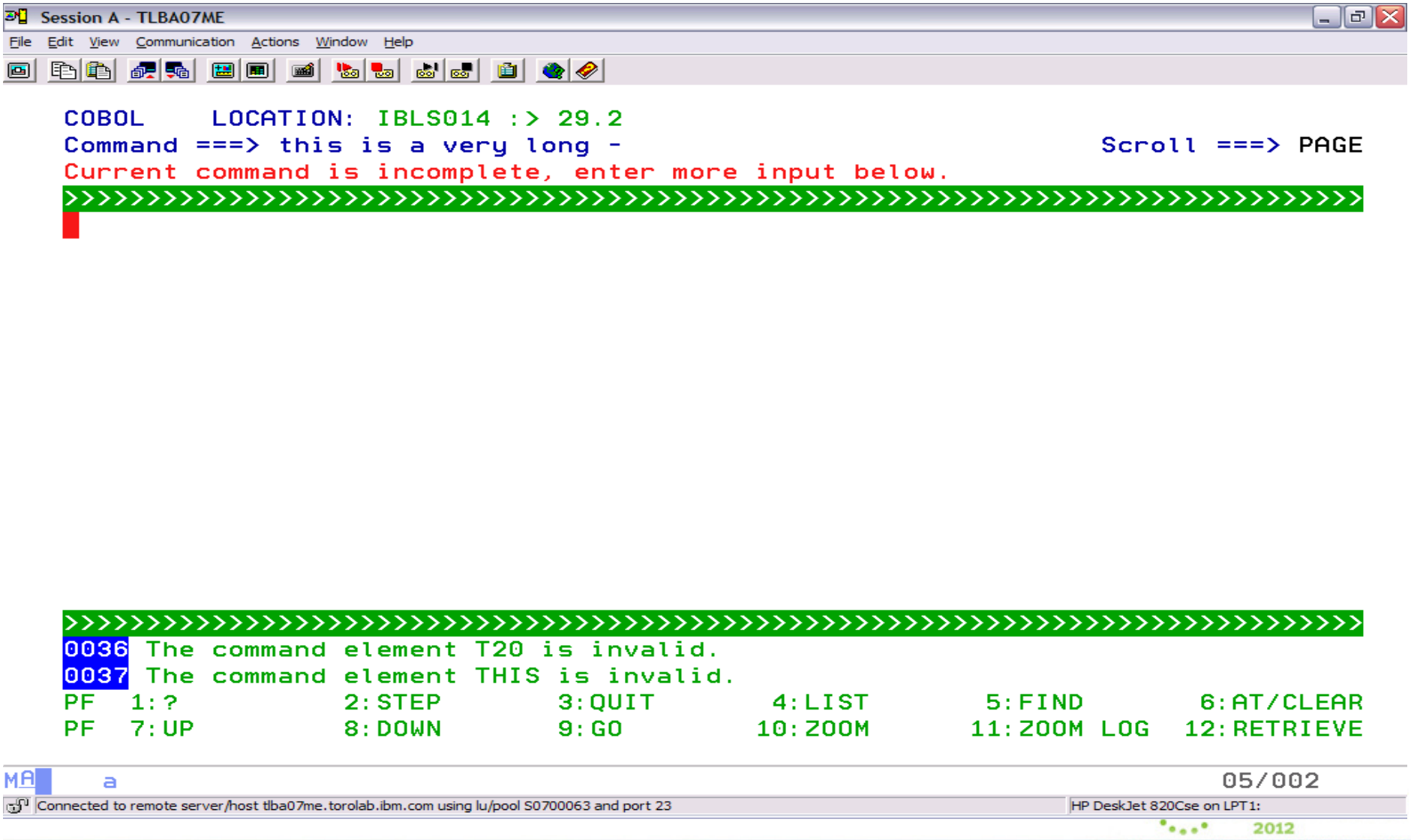
Popup window



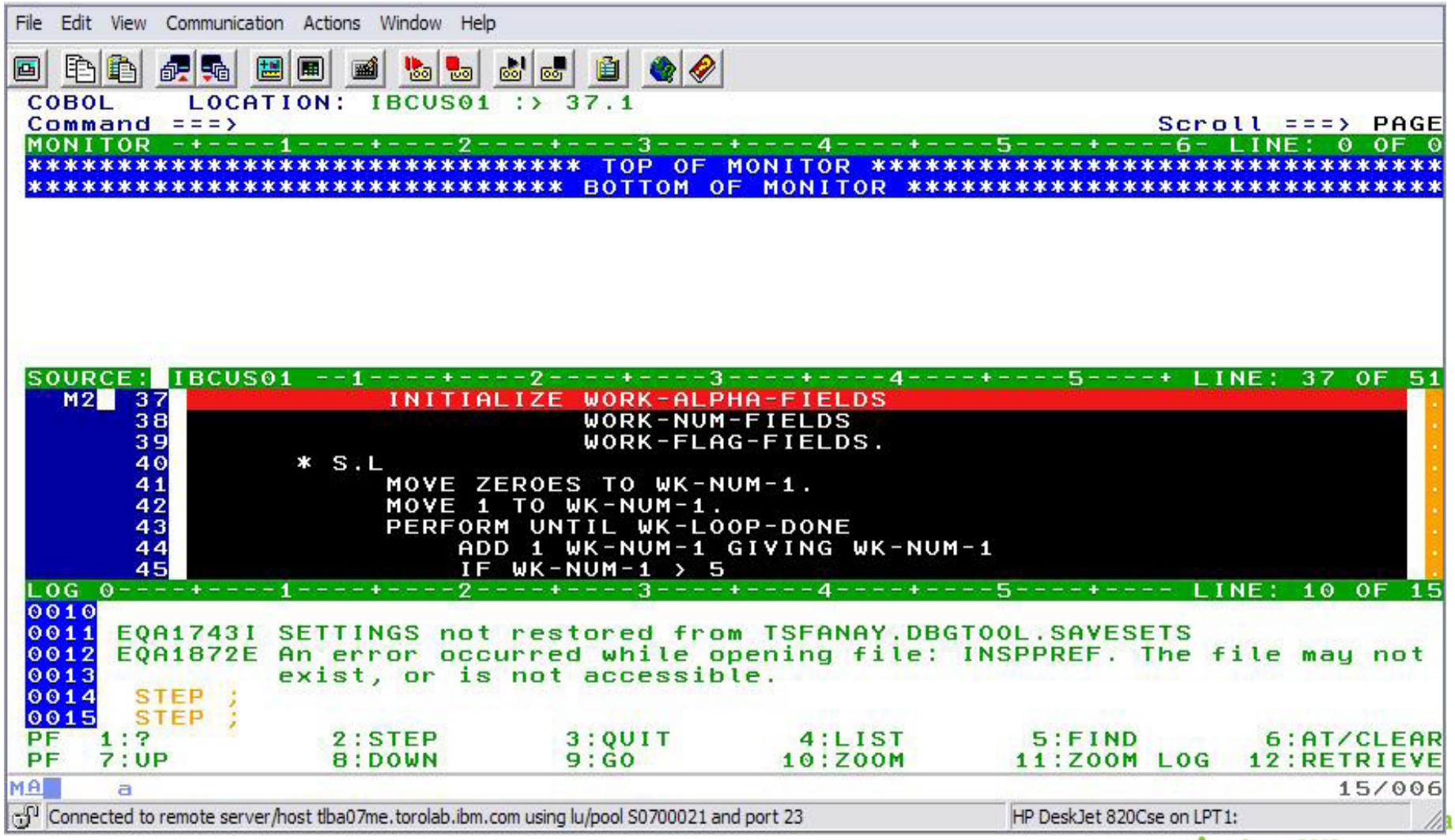
Enter after continuation character..



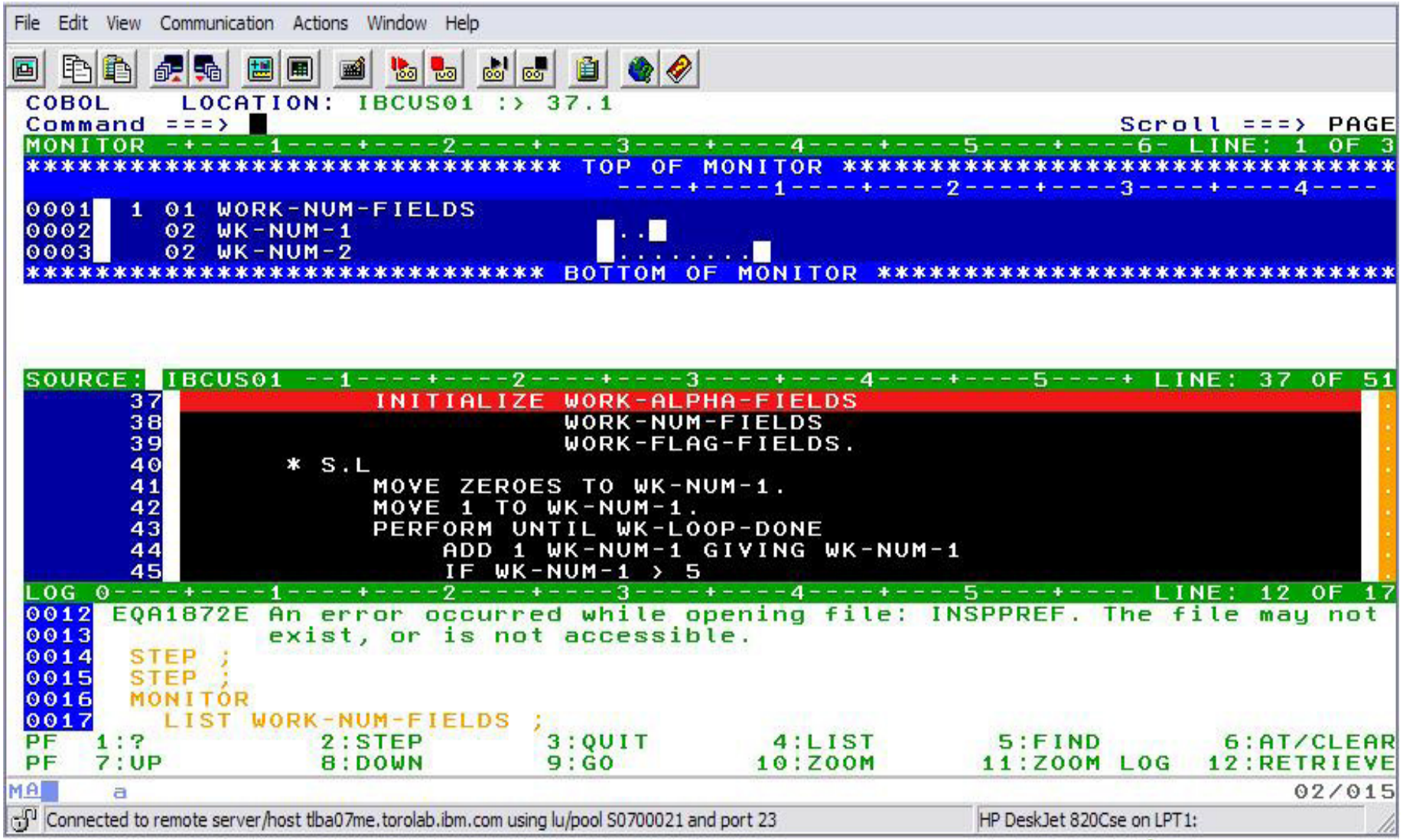
You get the popup window.



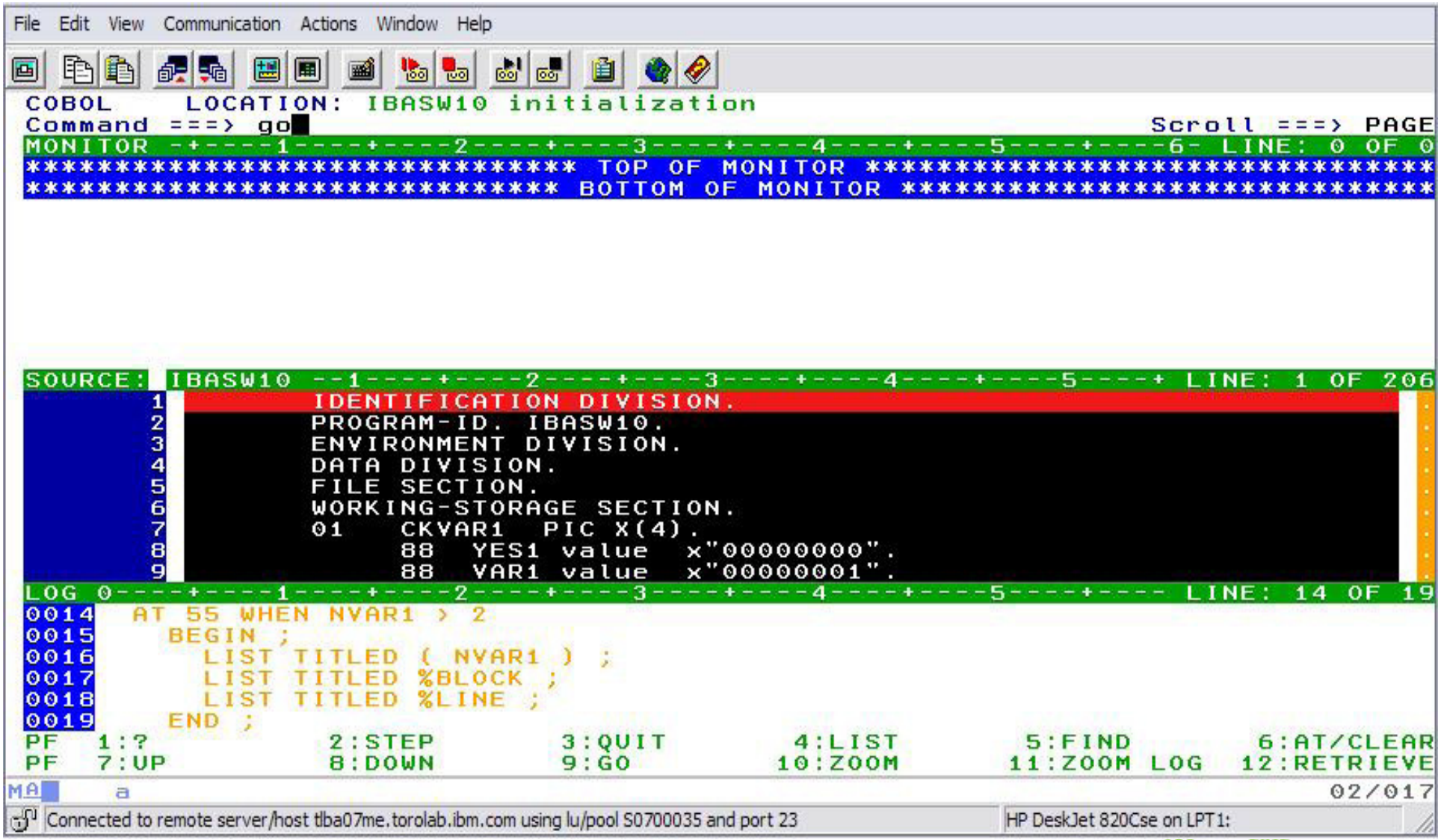
Adding a new monitor using prefix command Mn



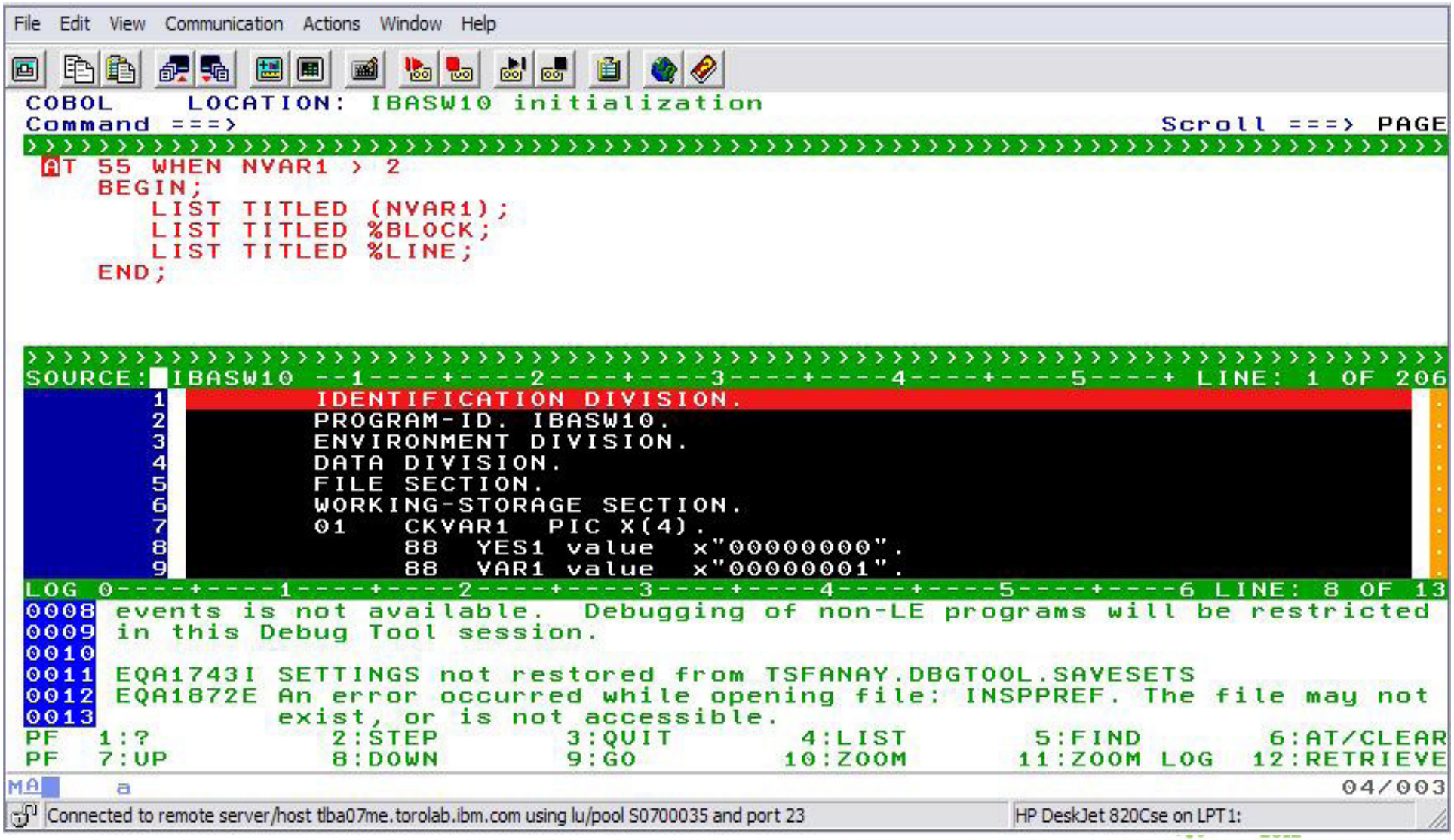
Second variable is now being monitored



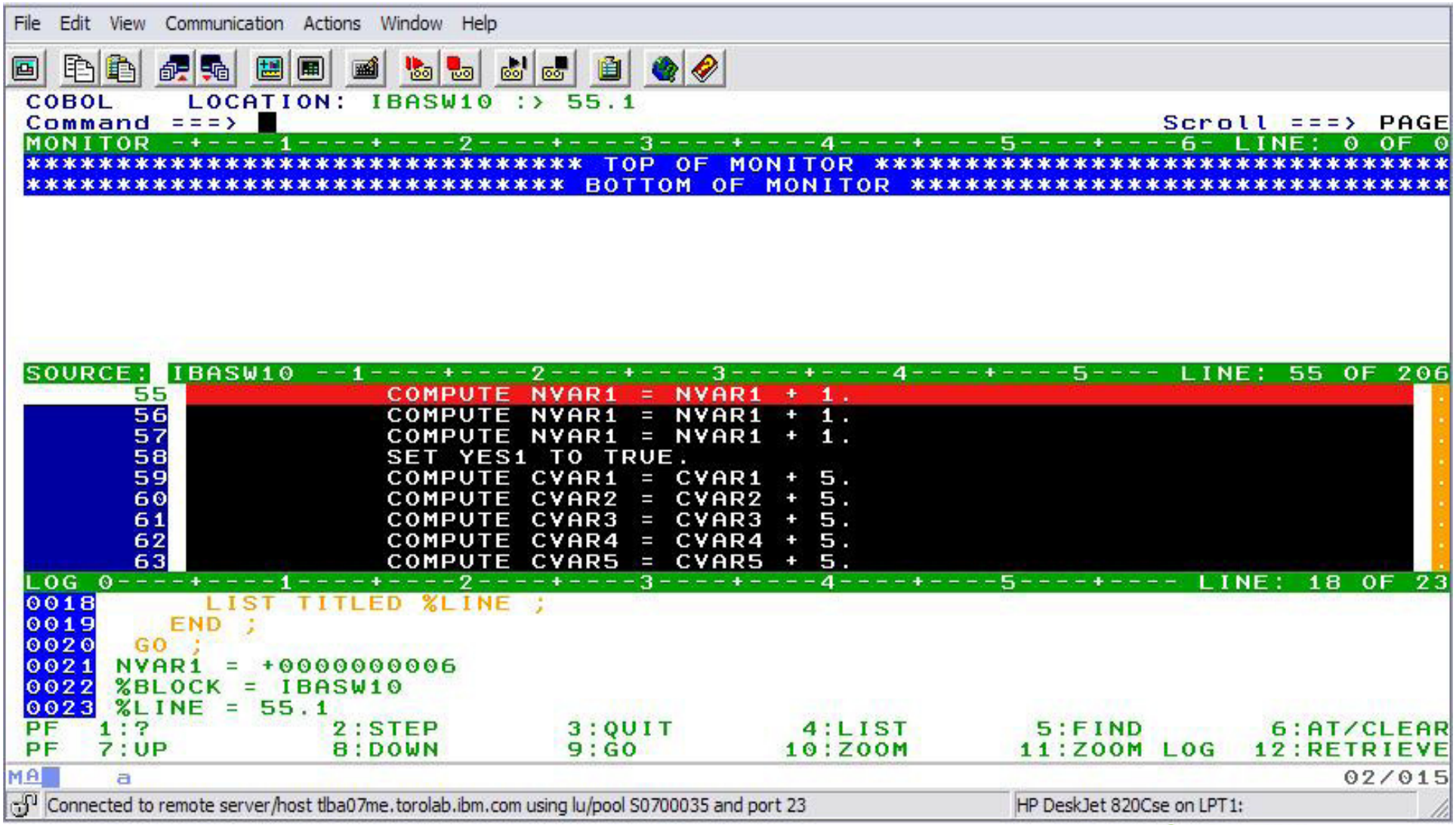
AT Statement……



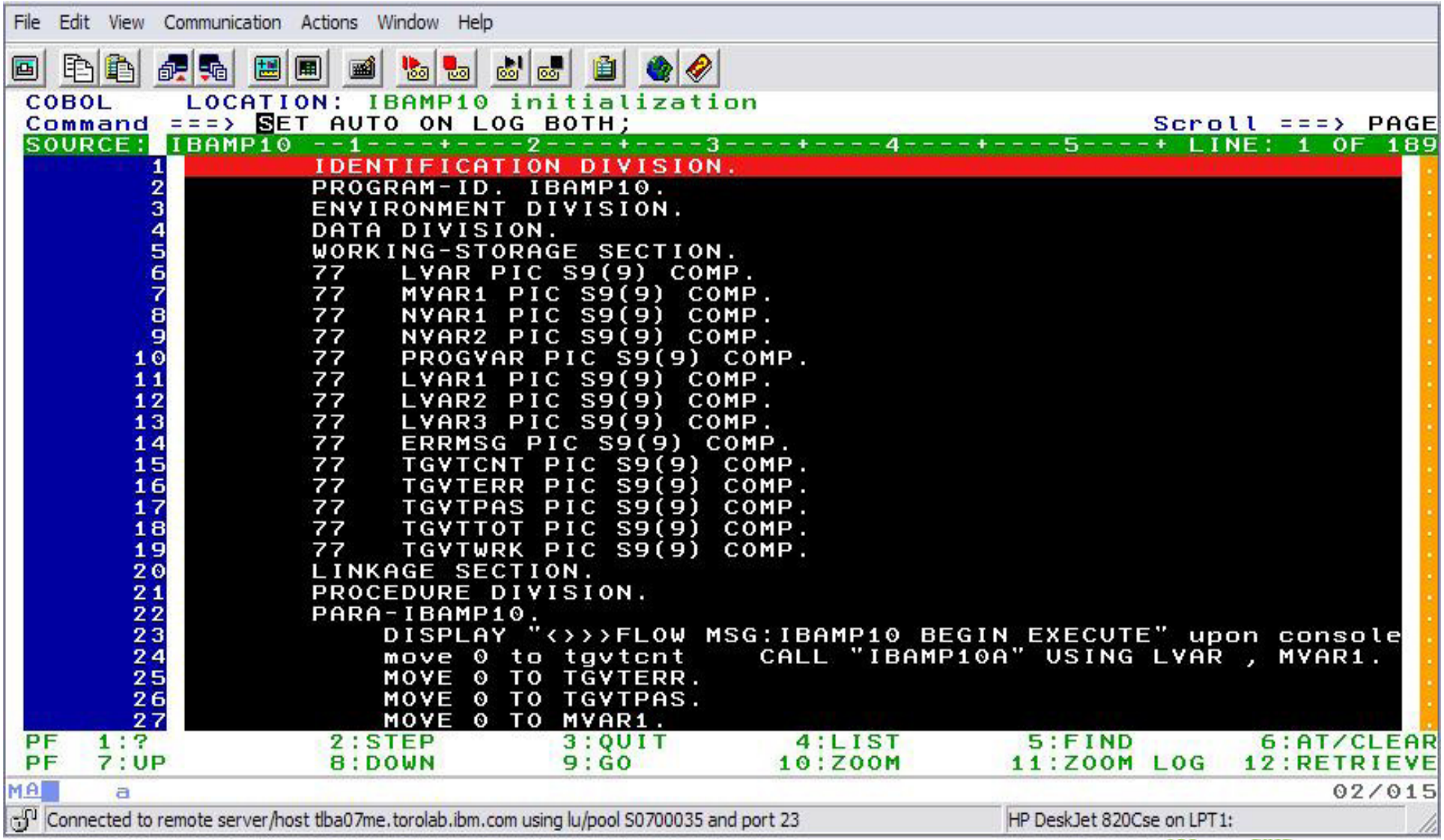
AT Statement WHEN and Grouping commands using BEGIN/END.



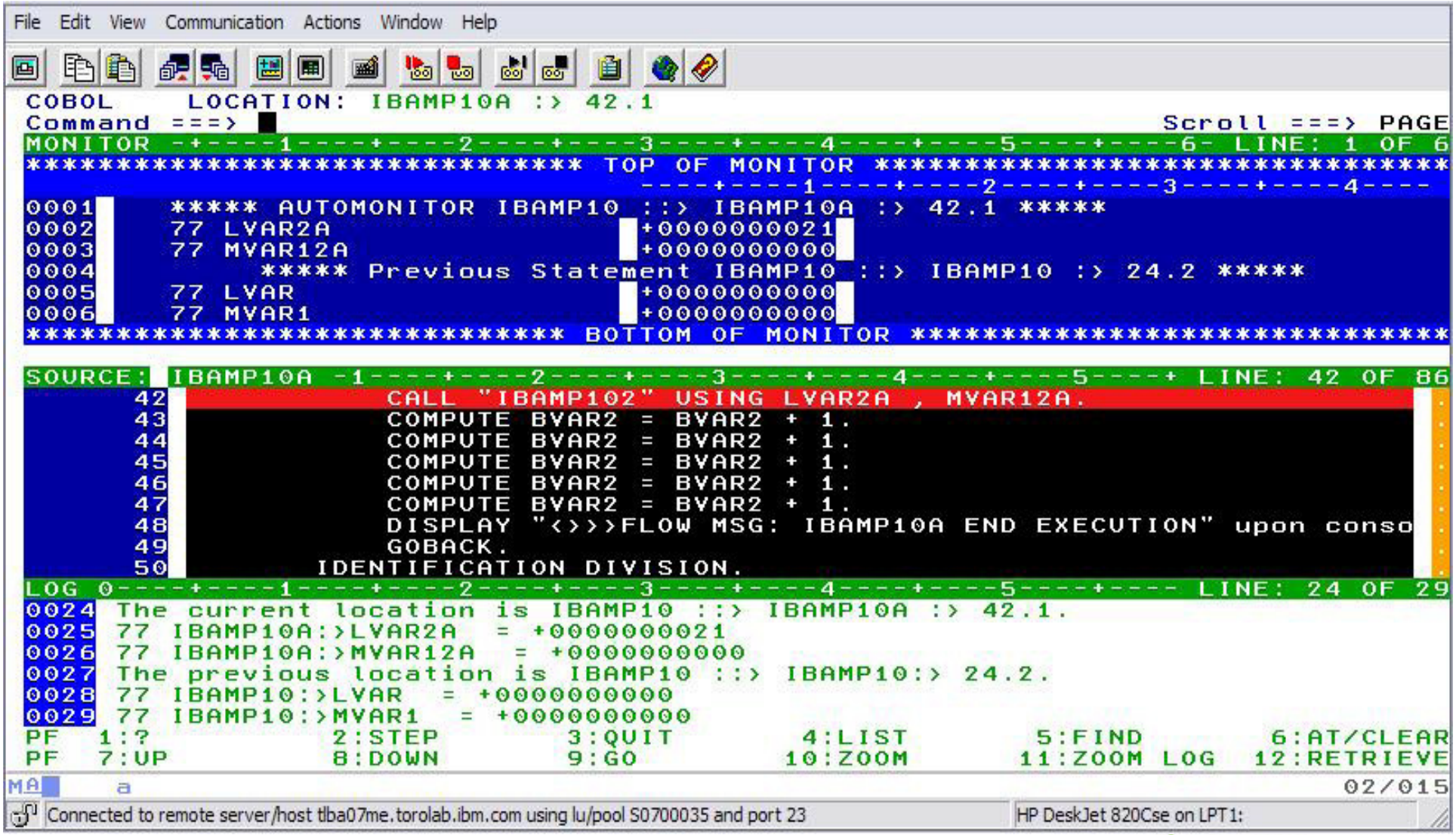
AT Statement……



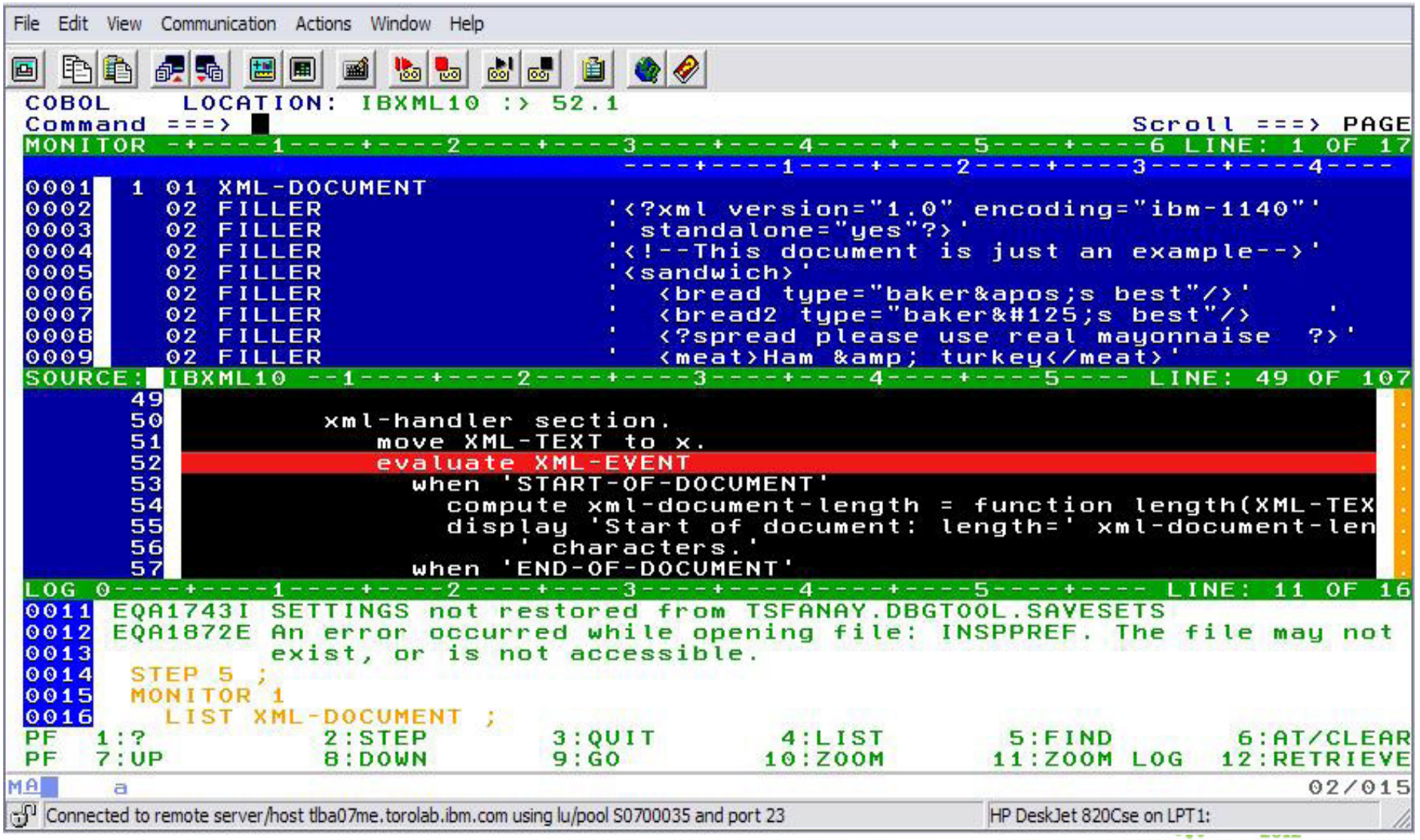
SET AUTO (BOTH, PREVIOUS)



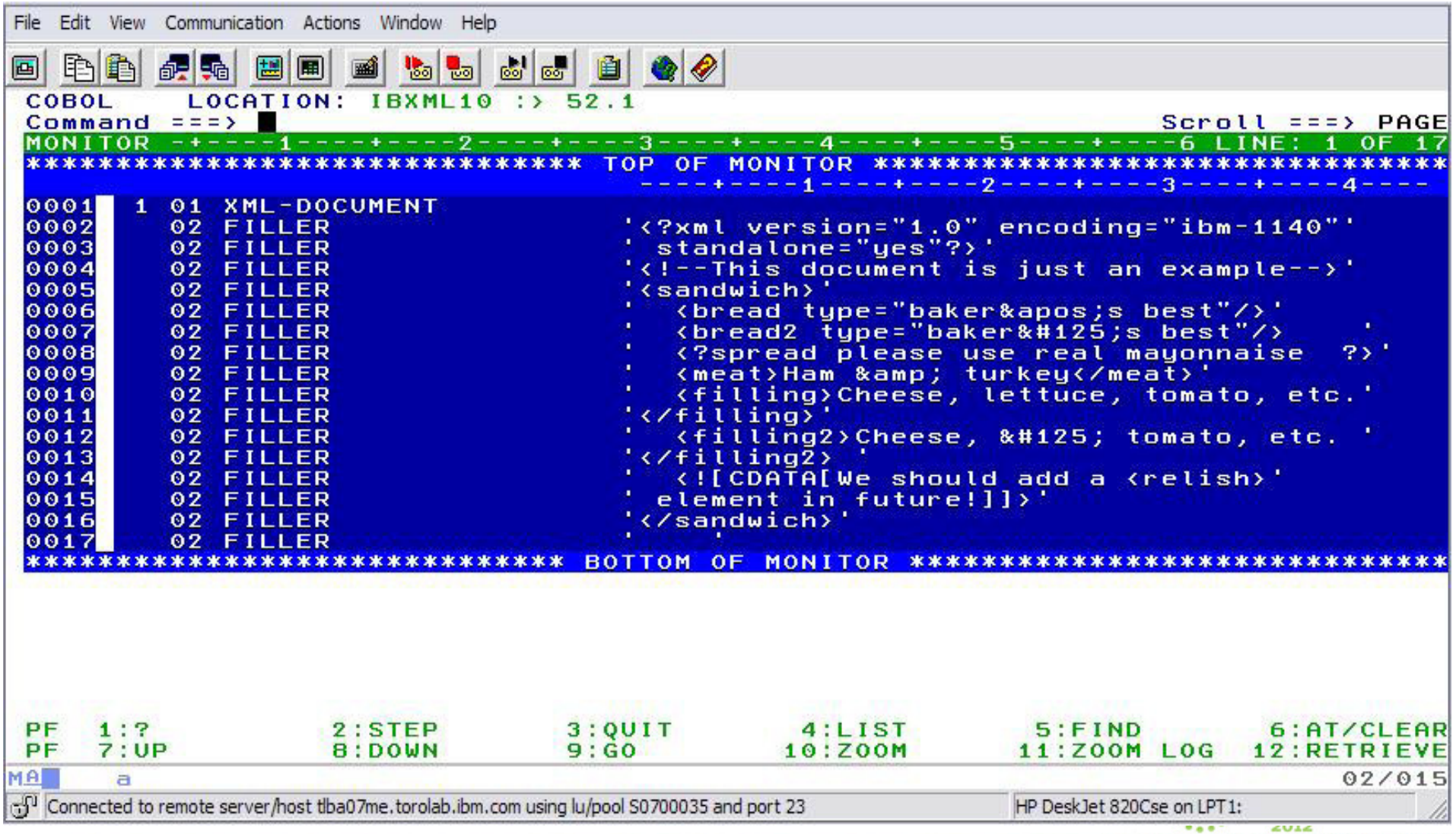
SET AUTO…..


File Edit View Communication Actions Window Help
COBOL LOCATION: IBAMP10A :> 42.1
Command ===>
Scroll ===> PAGE
MONITOR -+----1----+----2----+----3----+----4----+----5----+----6- LINE: 1 OF 6
********** TOP OF MONITOR **********
0001 ***** AUTOMONITOR IBAMP10 ::> IBAMP10A :> 42.1 *****
0002 77 LVAR2A +0000000021
0003 77 MVAR12A +0000000000
0004 ***** Previous Statement IBAMP10 ::> IBAMP10 :> 24.2 *****
0005 77 LVAR +0000000000
0006 77 MVAR1 +0000000000
********** BOTTOM OF MONITOR **********
SOURCE: IBAMP10A -1----+----2----+----3----+----4----+----5----+ LINE: 42 OF 86
42 CALL "IBAMP102" USING LVAR2A , MVAR12A.
43 COMPUTE BVAR2 = BVAR2 + 1.
44 COMPUTE BVAR2 = BVAR2 + 1.
45 COMPUTE BVAR2 = BVAR2 + 1.
46 COMPUTE BVAR2 = BVAR2 + 1.
47 COMPUTE BVAR2 = BVAR2 + 1.
48 DISPLAY "<>>>FLOW MSG: IBAMP10A END EXECUTION" upon conso
49 GOBACK.
50 IDENTIFICATION DIVISION.
LOG 0----+----1----+----2----+----3----+----4----+----5----+---- LINE: 24 OF 29
0024 The current location is IBAMP10 ::> IBAMP10A :> 42.1.
0025 77 IBAMP10A:>LVAR2A = +0000000021
0026 77 IBAMP10A:>MVAR12A = +0000000000
0027 The previous location is IBAMP10 ::> IBAMP10:> 24.2.
0028 77 IBAMP10:>LVAR = +0000000000
0029 77 IBAMP10:>MVAR1 = +0000000000
PF 1:? 2:STEP 3:QUIT 4:LIST 5:FIND 6:AT/CLEAR
PF 7:UP 8:DOWN 9:GO 10:ZOOM 11:ZOOM LOG 12:RETRIEVE
MA a 02/015
Connected to remote server/host tlba07me.torolab.ibm.com using lu/pool S0700035 and port 23
HP DeskJet 820Cse on LPT1:

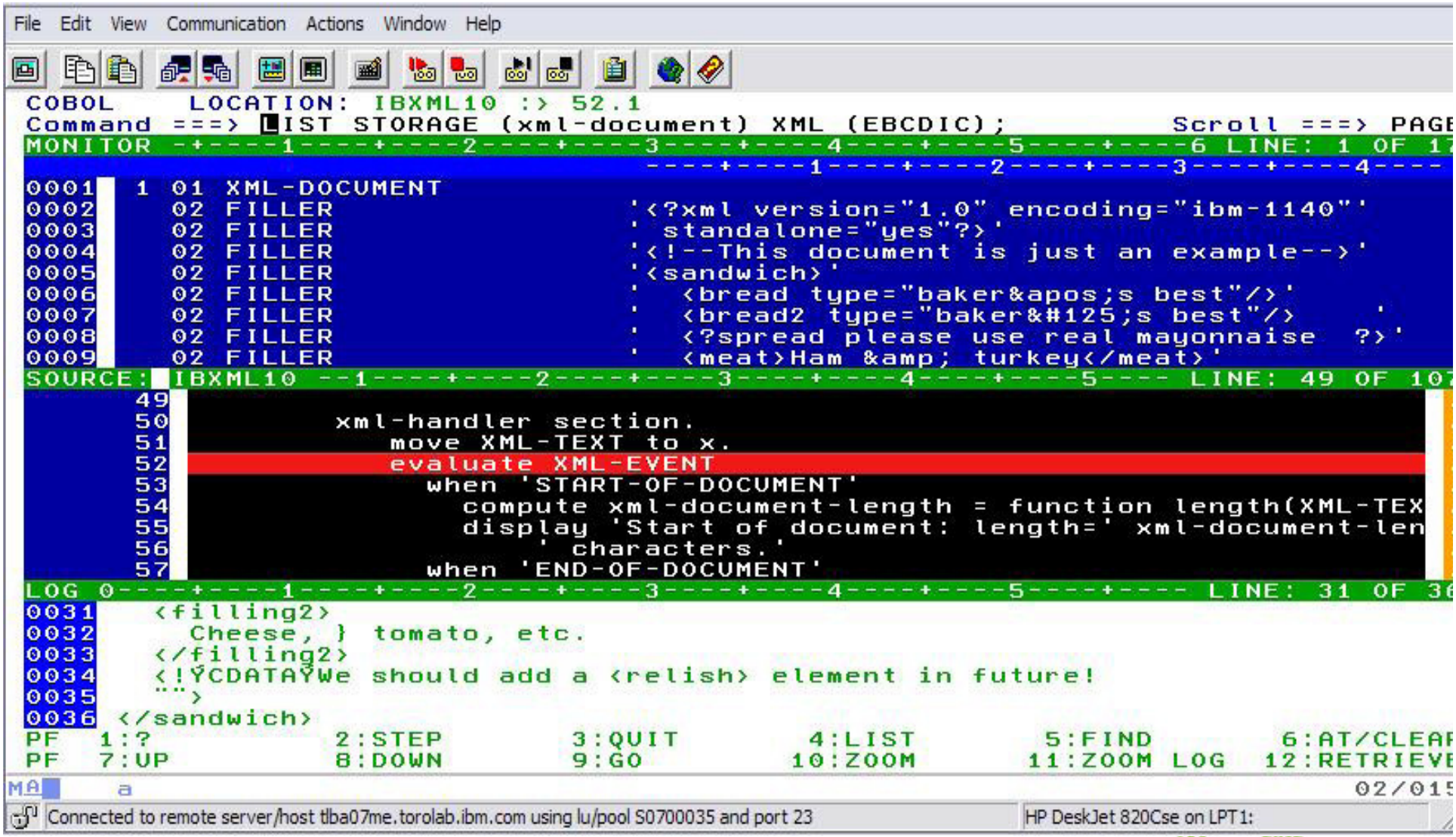
XML formatting



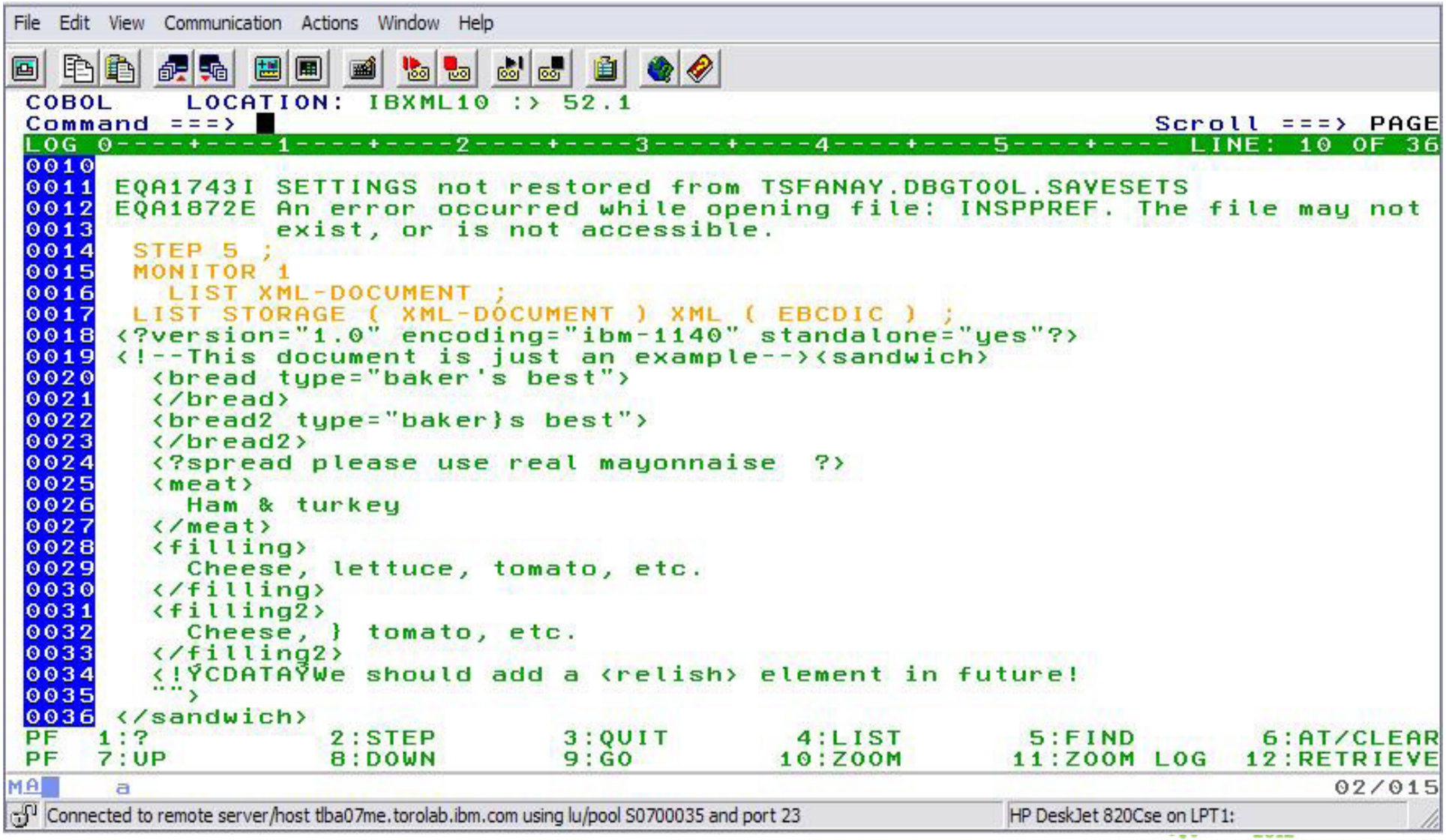
XML…. (zooming monitor area)



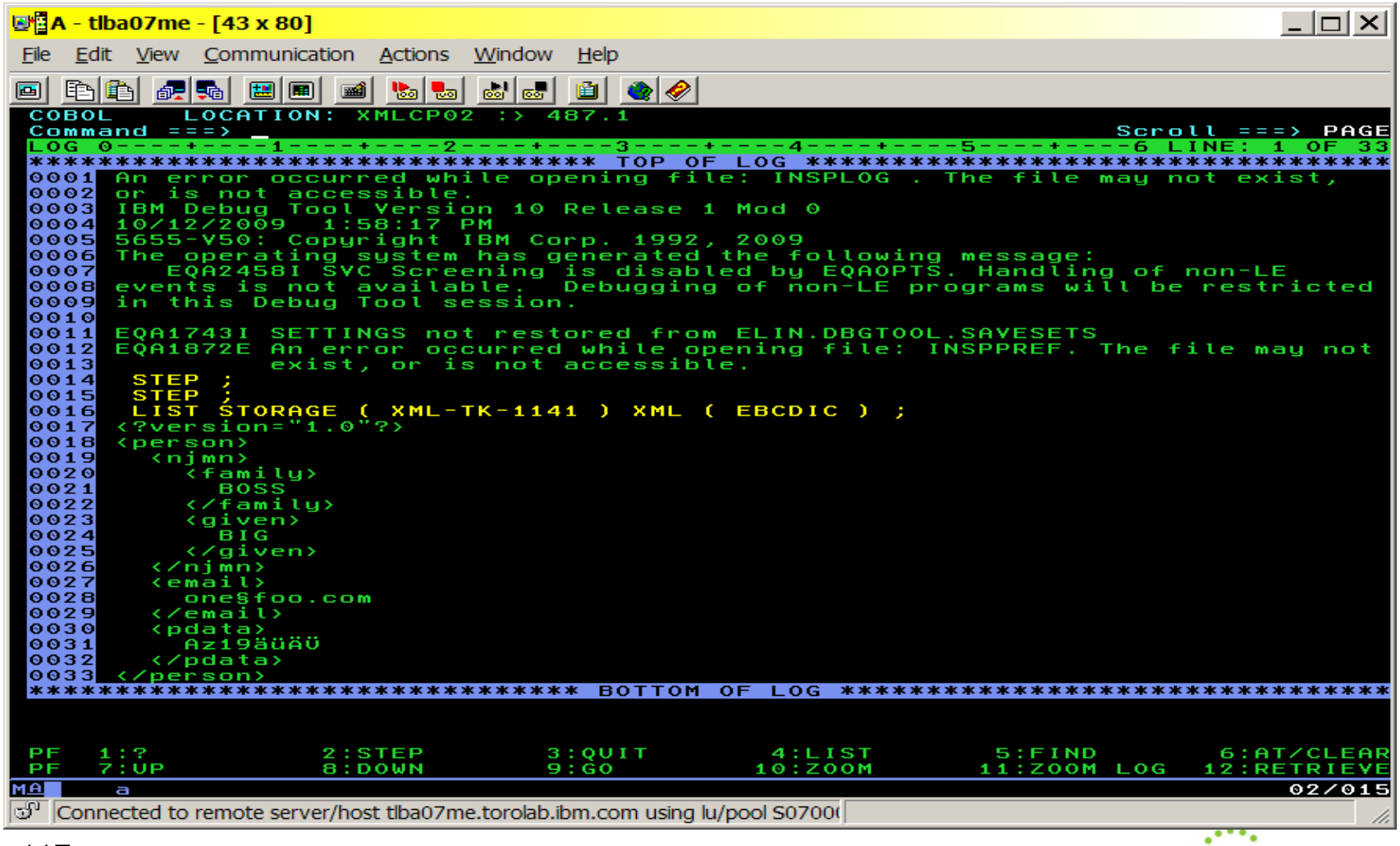
XML…(before entering LIST STORAGE)



XML… formatted document



XML doc in code page 1141 with XML(EBCDIC)



Debug Tool: What's new in version 11?

- A new mode of operation, explicit debug mode, is now supported. In this mode, the user identifies the compile units to debug, then Debug Tool loads debug data only for those compile units. This mode can significantly improve debugger performance when it is debugging very large and complex programs. This new mode is an alternative to the standard Debug Tool mode of operation where debug data is automatically loaded for all compile units. It is intended to be used only when debugging large, complex applications that don't perform as well in the standard Debug Tool mode.
- A new user interface is added to the Terminal Interface Manager (TIM) that helps you create and manage the TEST runtime options data set.
- The TIM has been enhanced to remove the need for a site to set up a separate TN3270E port or to customize a set of terminal LUs.
- A GUI interface is added that helps you create and manage the TEST runtime options data set from the workstation.
- The Debug Tool Language Environment user exit for DB2 (EQADDCXT) now supports debugging of DB2 stored procedures of type SUB invoked using the call_sub function.
- EQAOPTS commands can now be specified at runtime in addition to the use of a user-generated EQAOPTS load module. This allows individual users to enter EQAOPTS commands at runtime by supplying a data set containing EQAOPTS commands.
- Support is added for debugging of assembler programs that exploit the latest IBM zEnterprise 196 architecture.
- Enhanced performance while debugging C and C++ applications.

Debug Tool: What's new in version 11?

- A Popup window, which displays the result of the LIST expression command when the Log window, is not visible.
- Support for the Enterprise PL/I ADDRDATA built-in function.
- Support for the Enterprise PL/I V4.1 compiler and its new GONUMBER(SEPARATE) option.
- For programs compiled with any level of Enterprise PL/I, you can now list a single element of an array of structures. For programs compiled with Enterprise PL/I V4.1, you can list a single element of an array of structures in automonitor or use the L prefix command in the Source window to list a single element of an array of structures.
- For programs compiled with any level of Enterprise PL/I, you can now change the format in which Debug Tool displays an array. By using the SET LIST BY SUBSCRIPT ON command, you can have Debug Tool display the array as it is stored in memory.
- A new keyword LABELS is added to the LIST NAMES command where you can list the names of all section and paragraph names in a COBOL program, and the names of all instruction labels in an assembler program.

Debug Tool: What's new in version 11?

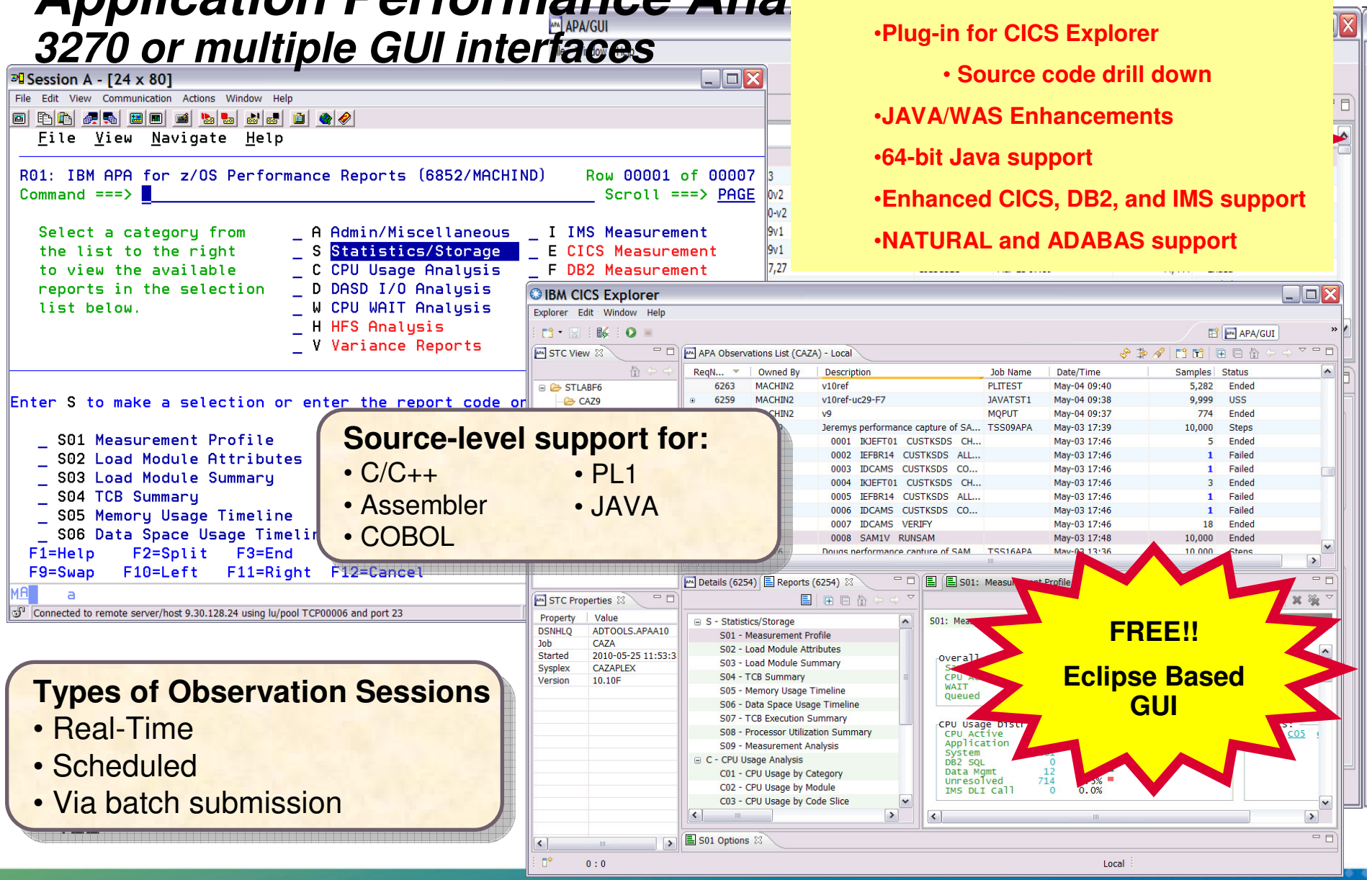
- The following breakpoints are enhanced:
  - AT CHANGE and AT LABEL breakpoints are enhanced to allow a user to limit the scope of the breakpoint to a specific compile unit.
  - AT GLOBAL is enhanced to provide an OCCURRENCE option or wild card (*) to stop for any condition raised in the application.
  - The QUERY LOCATION command is enhanced to provide more information when Debug Tool stops for an AT CHANGE breakpoint.
- Automonitor enhancements:
  - You can change the subscripts of an array directly in the Monitor window.
  - You can delete multiple items from Monitor window at one time.
  - You can use the cursor (in combination with the CLEAR MONITOR command) to indicate which variable to remove from the Monitor window.
- Automated allocation of the commands, log, preferences, save settings and save breakpoints and monitor specifications files.
- New functions are included in Debug Tool Utilities to help an application programmer more easily start debugging IMS applications running in BTS.
- A CICS transaction, DTNP, is provided which issues NEWCOPY or PHASEIN of application programs.
- Documentation is provided to assist debugging of Language Environment C/C++, COBOL, and PLI programs in the Java JNI environment in z/OS.

Agenda

- Fault Analyzer
- Debug Tool
- Application Performance Analyzer
- File Manager

Application Performance Ana

3270 or multiple GUI interfaces



New in APA V11.1

- Plug-in for CICS Explorer
  - Source code drill down
- JAVA/WAS Enhancements
- 64-bit Java support
- Enhanced CICS, DB2, and IMS support
- NATURAL and ADABAS support

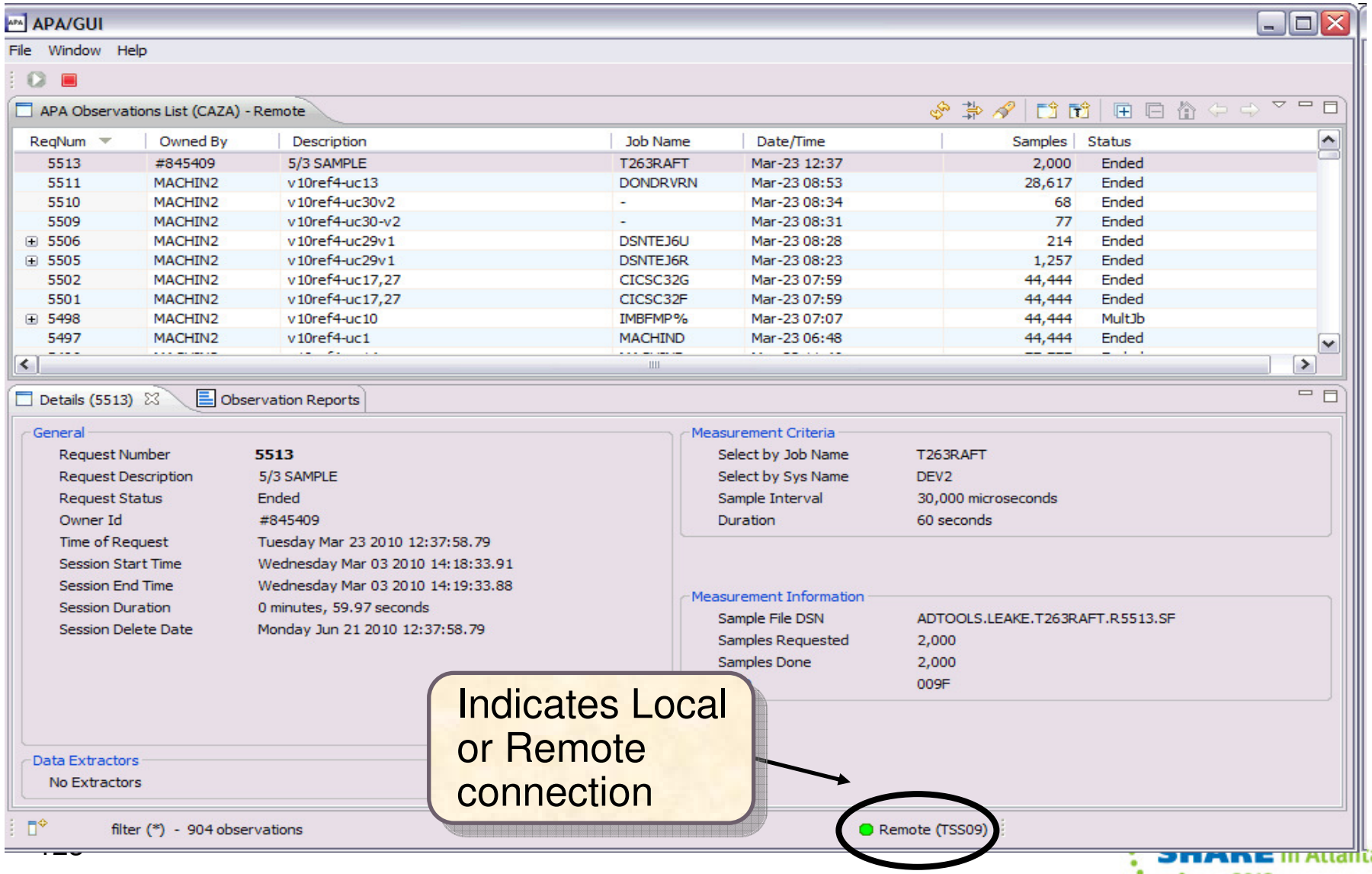
Source-level support for:

- C/C++
- Assembler
- COBOL
- PL1
- JAVA

Types of Observation Sessions

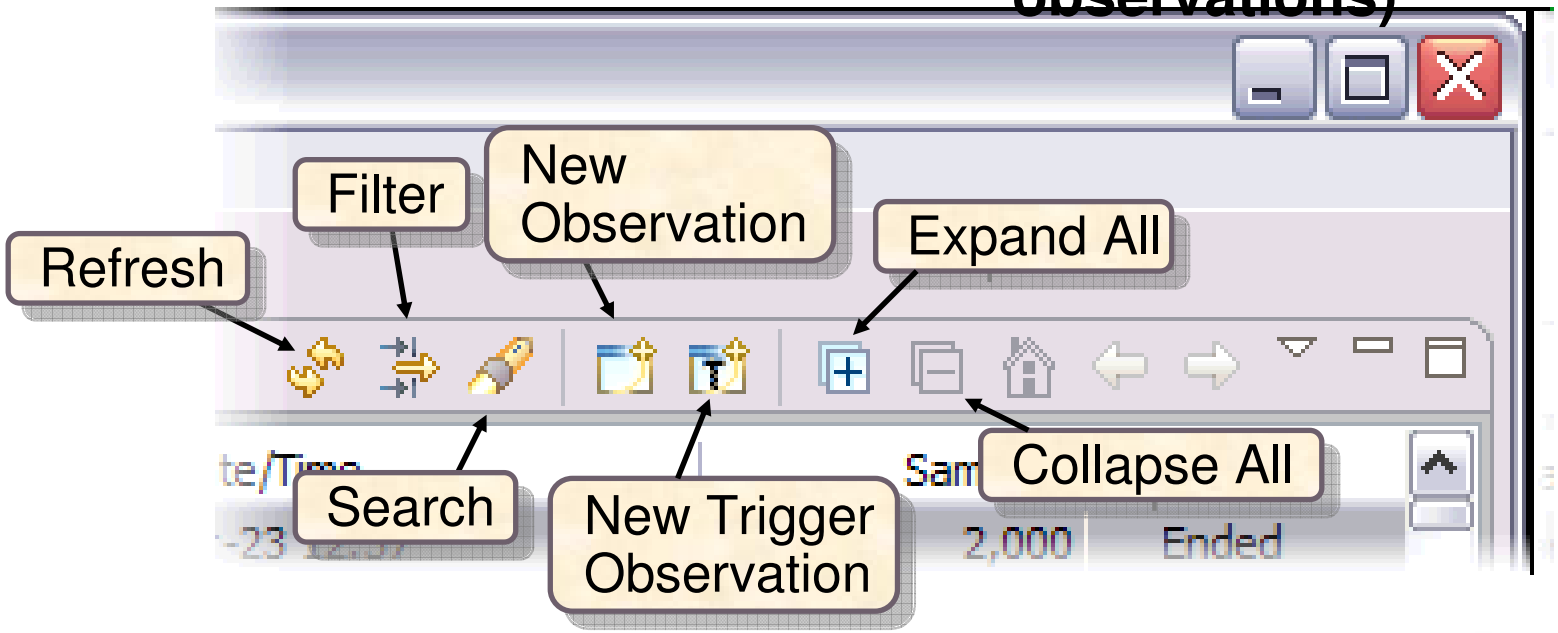
- Real-Time
- Scheduled
- Via batch submission

The Observations List View

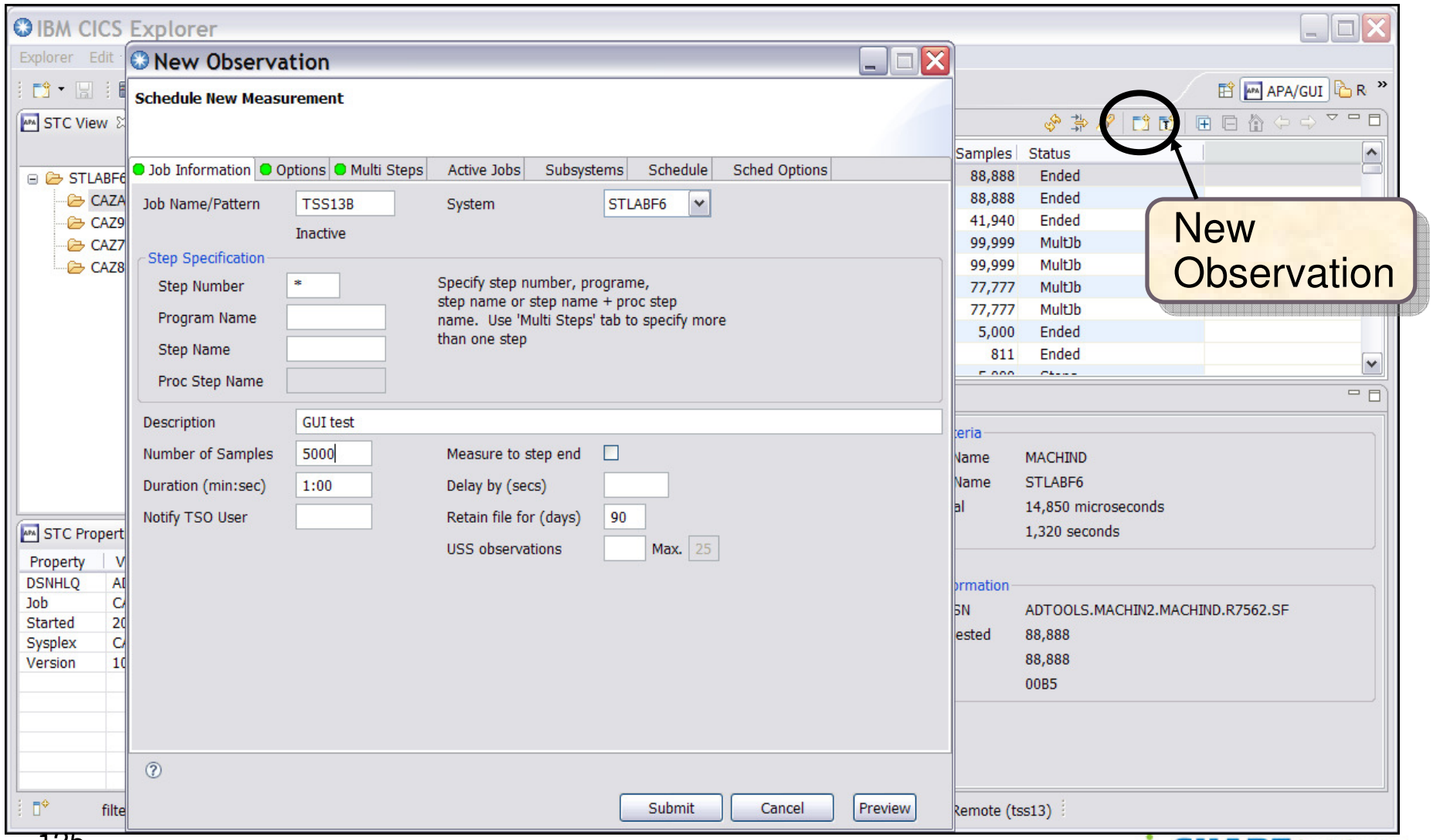


Observations List View Toolbar

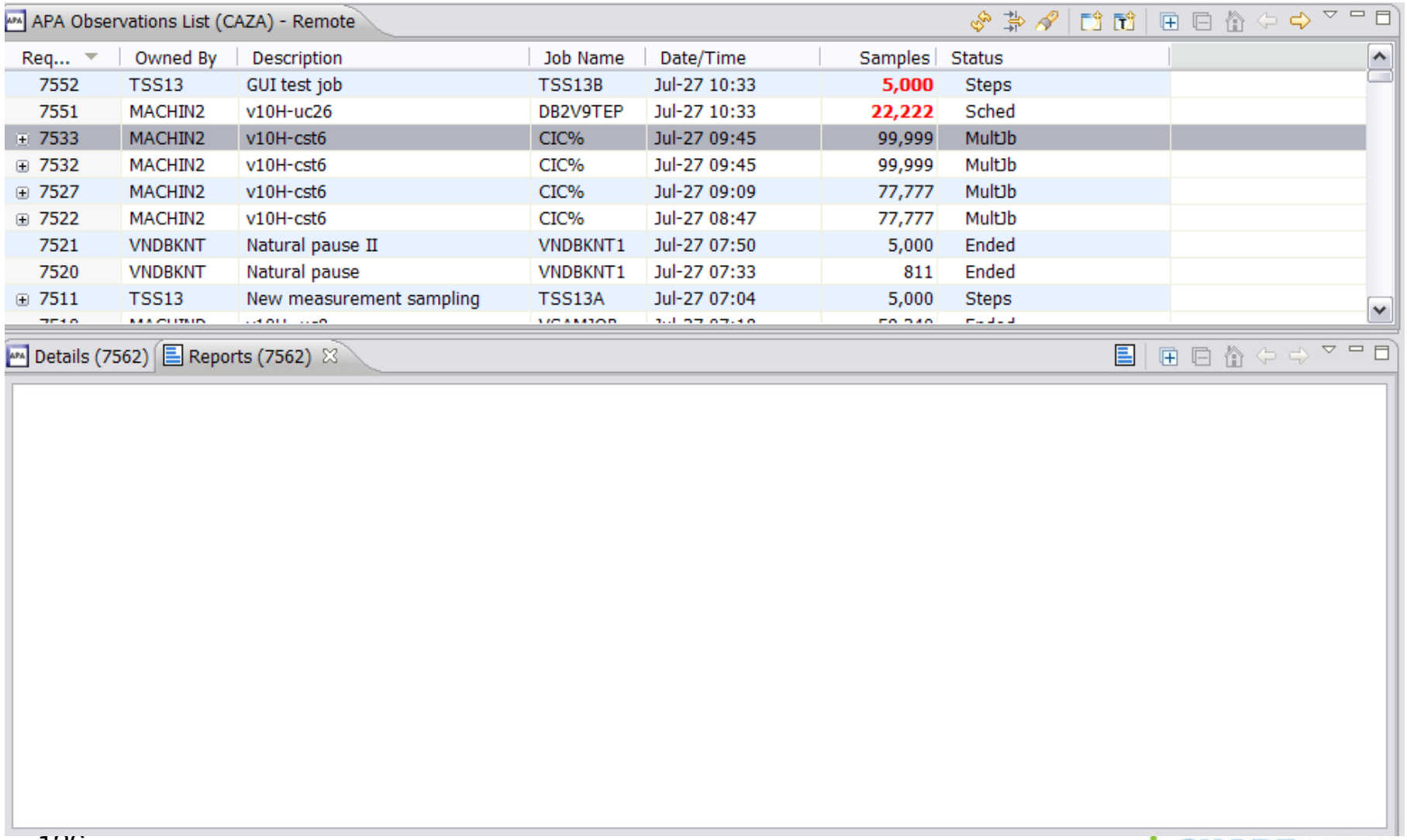
- **Observations List toolbar**
  - **Refresh Observations List**
  - **Filter Observations List**
  - **Search Observations**
  - **New Observation**
  - **New Trigger Observation**
  - **Expand All (expand all observations)**
  - **Collapse All (collapse all observations)**



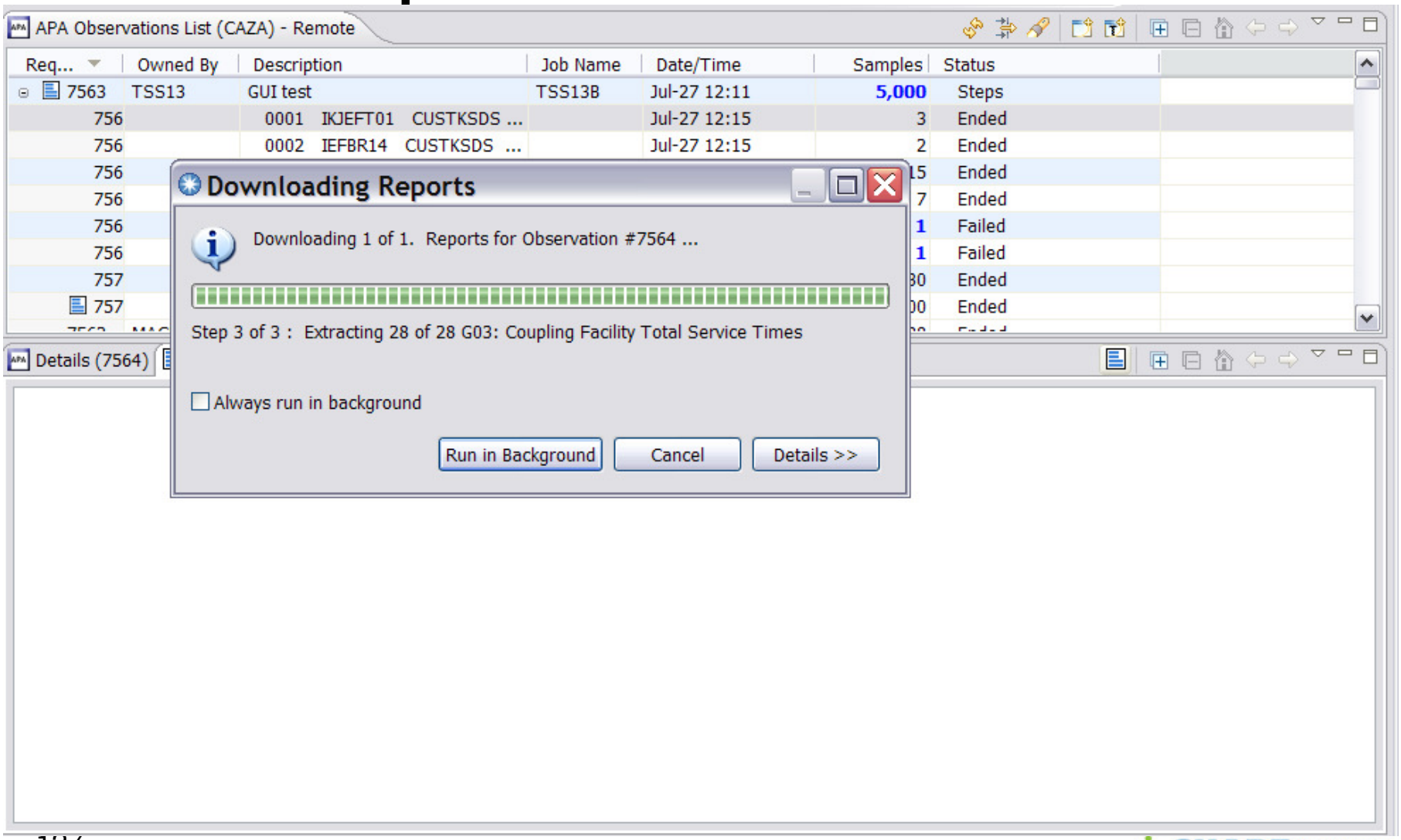
Click on new observation button
Create a new observation



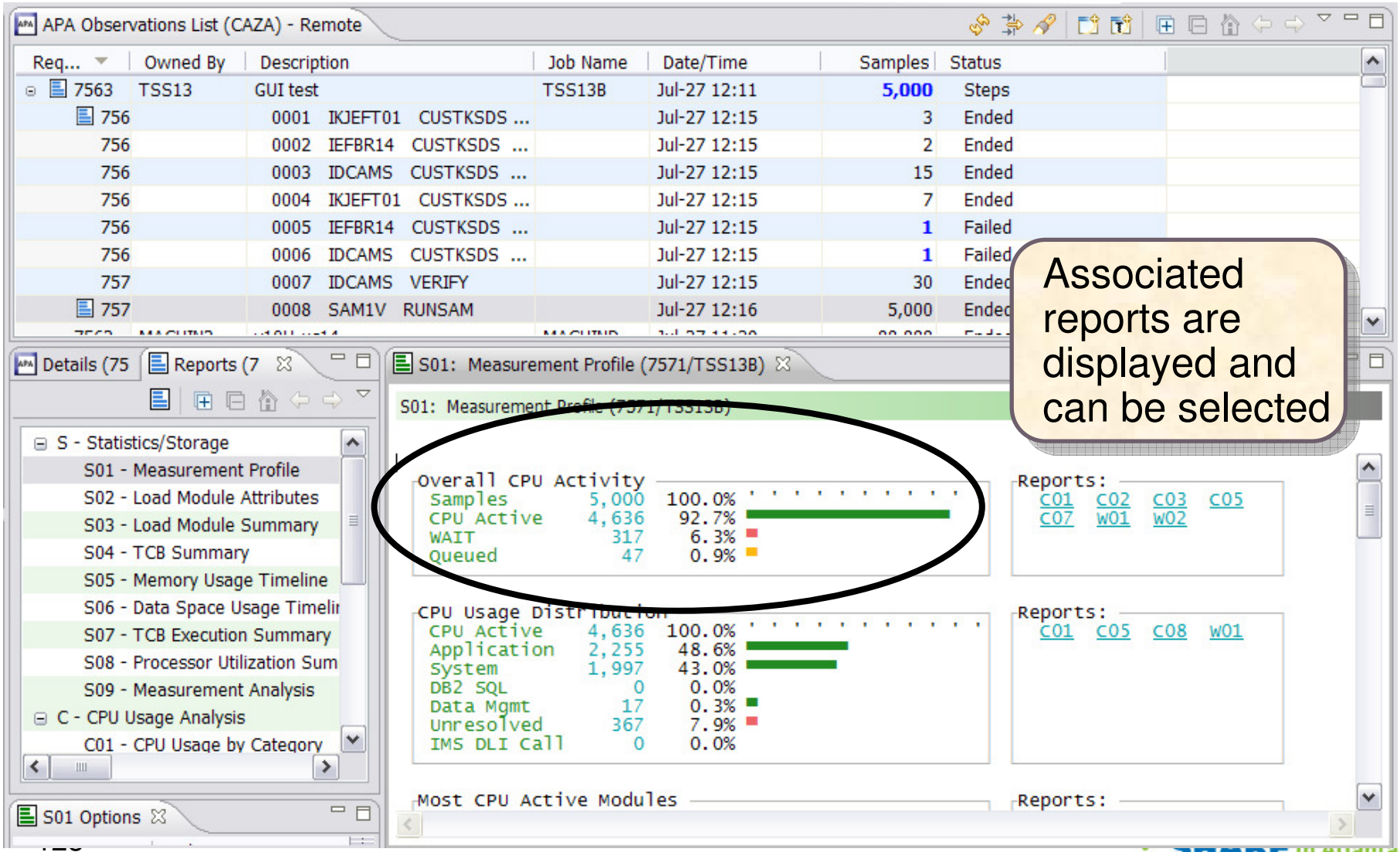
Observation is scheduled



Right click on report and select Download Reports

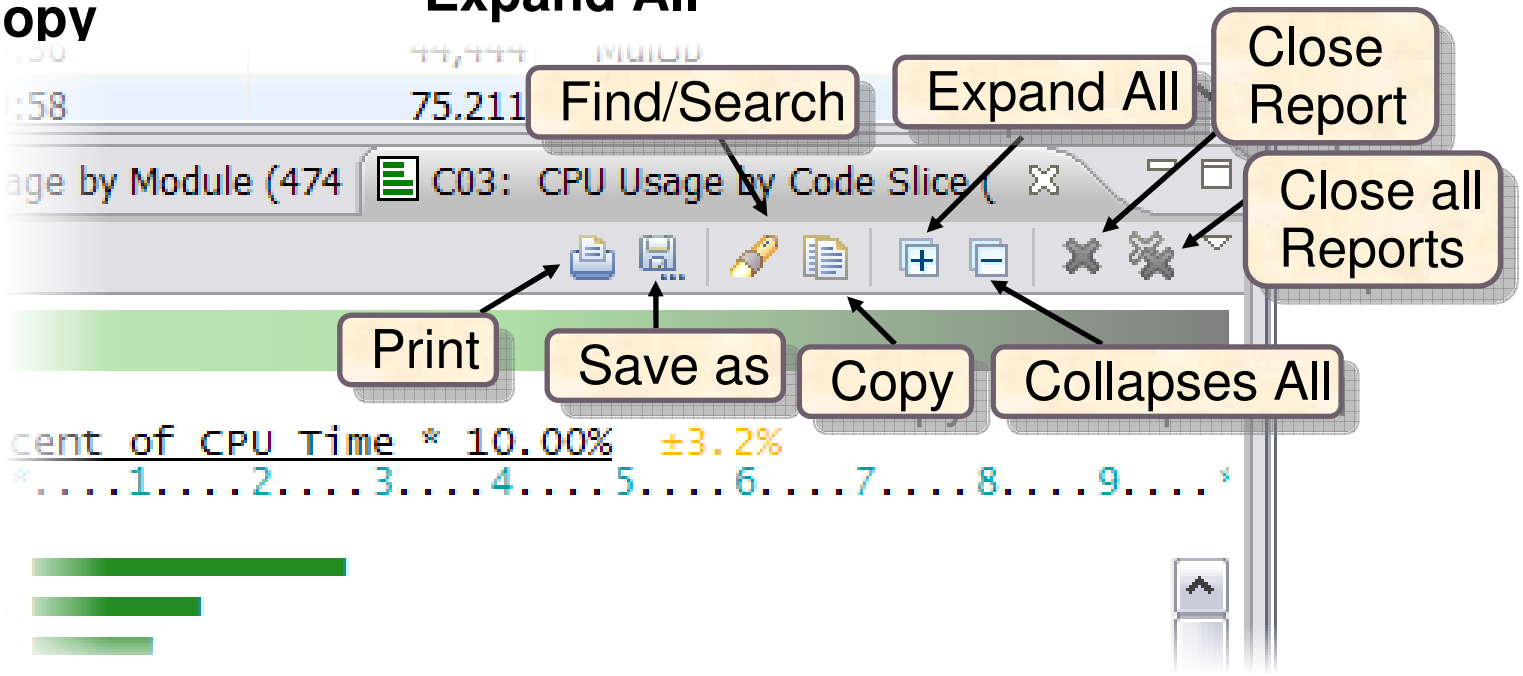


Report list is displayed
S01 Measurement Profile report shows high CPU activity

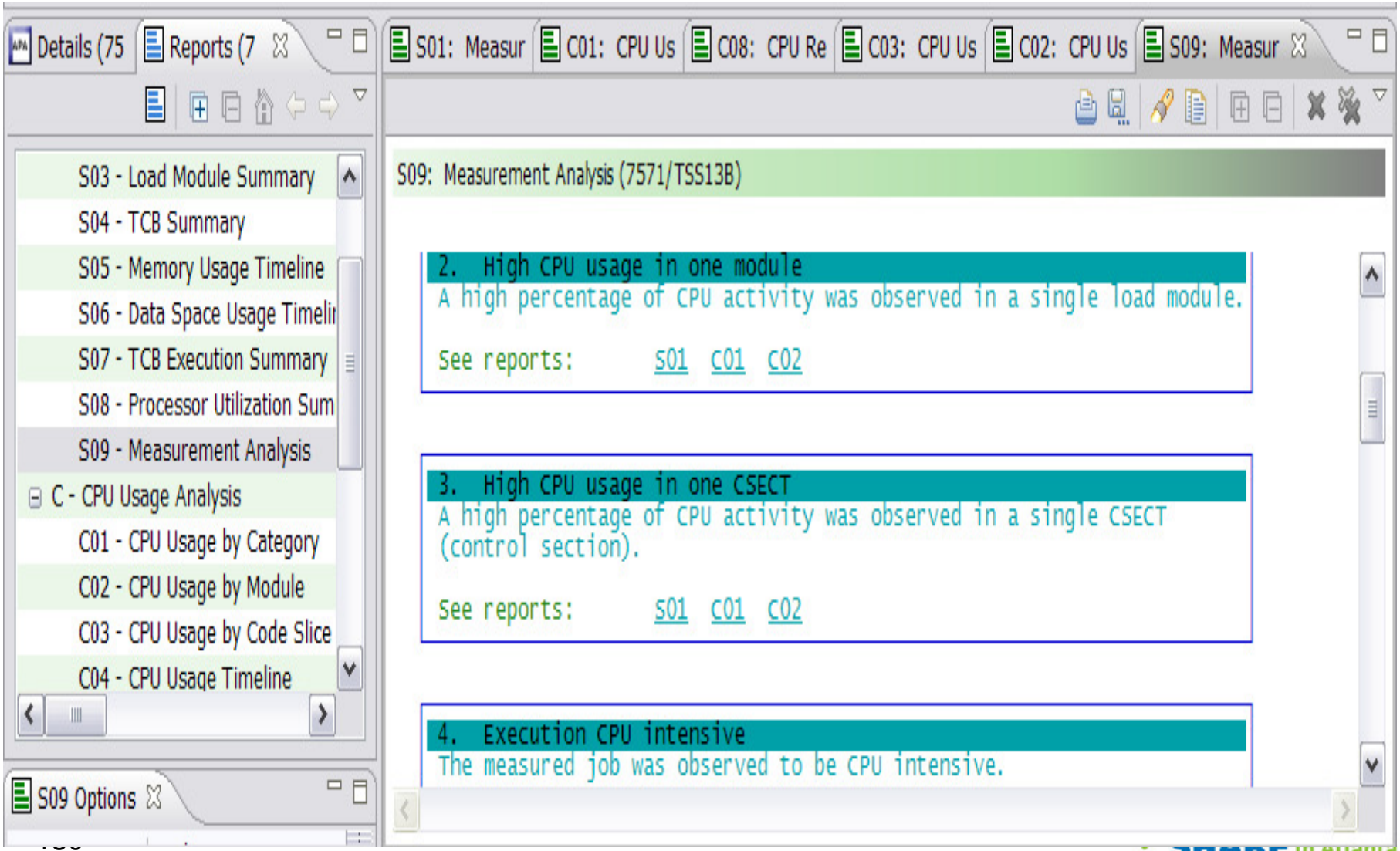


Report View

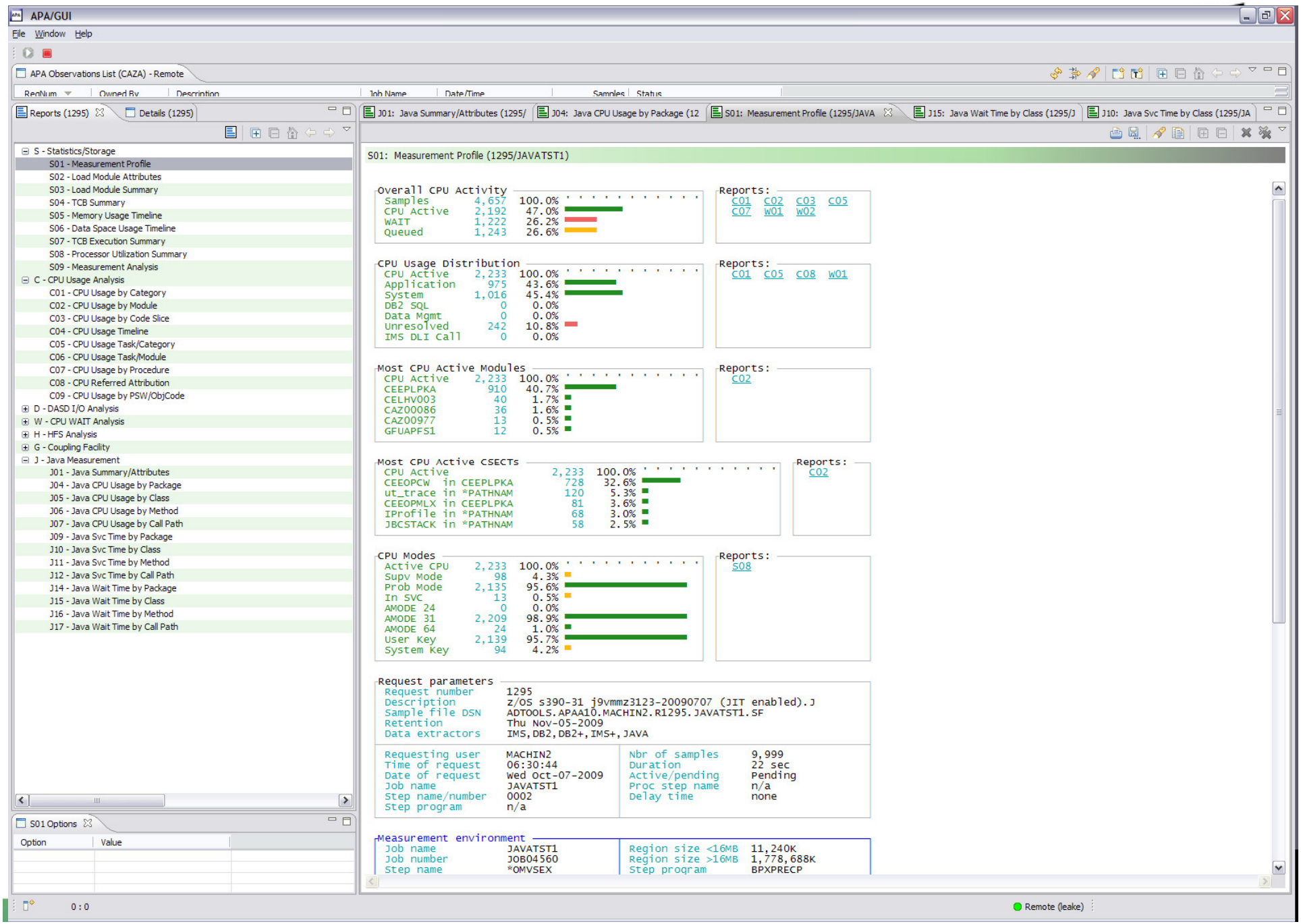
- **Toolbar**
  - **Provides buttons for report-level actions which include:**
    - **Print**
    - **Save As**
    - **Find**
    - **Copy**
    - **Close Report**
    - **Close All Reports**
    - **Expand All**



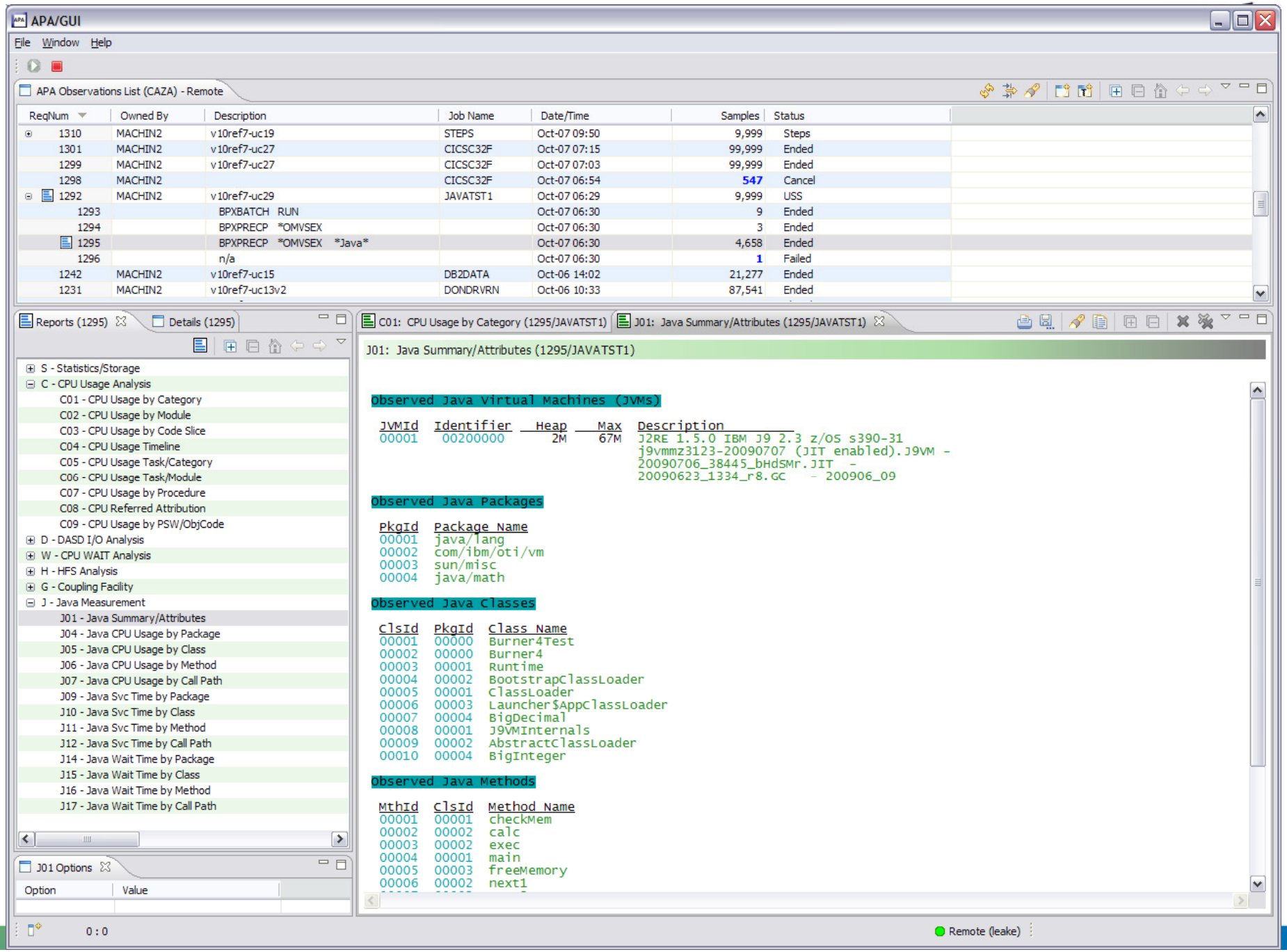
S09 Measurement Analysis report displays possible performance improvement



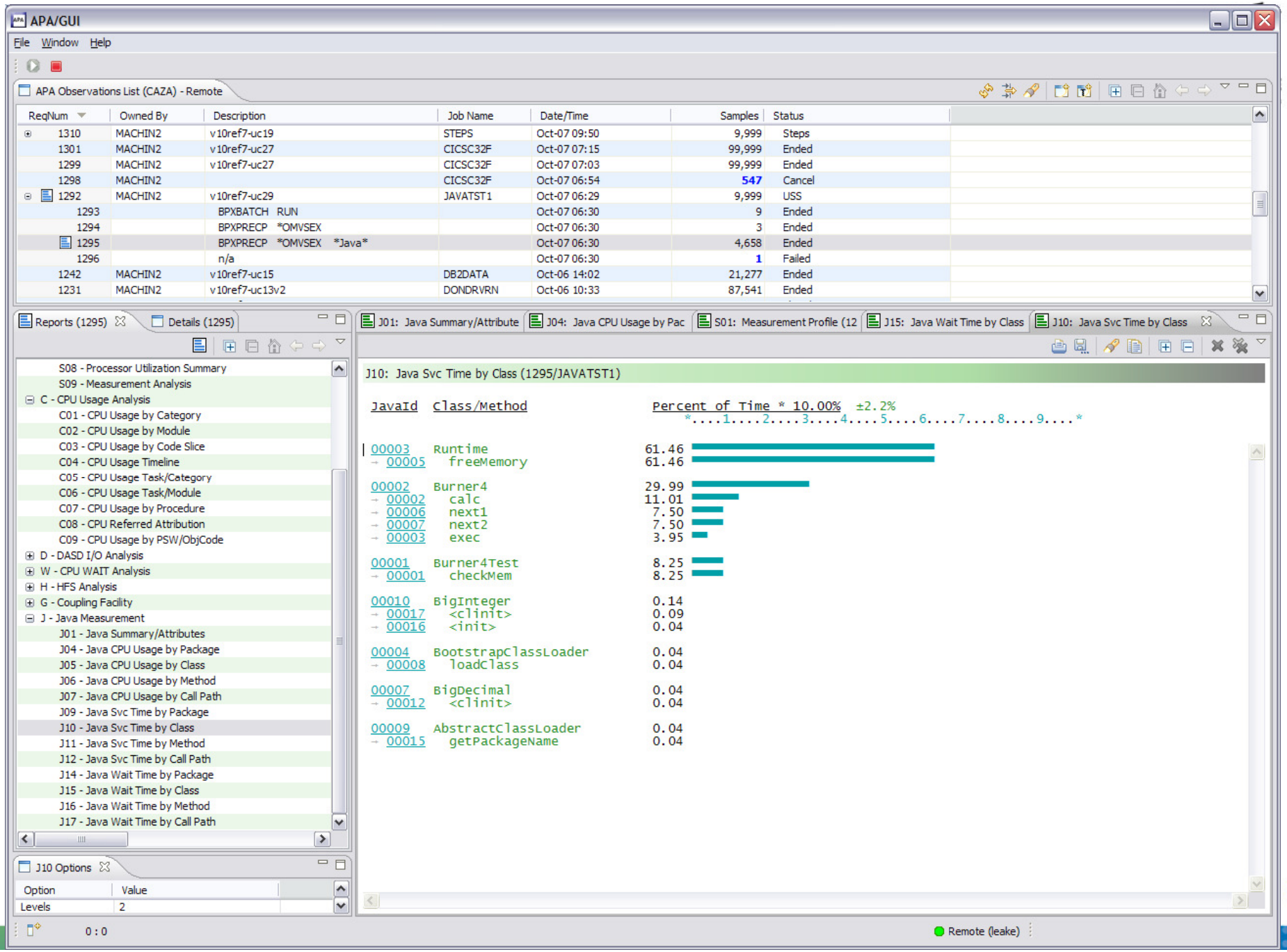
APA V11 Java Test Case (Java 5)



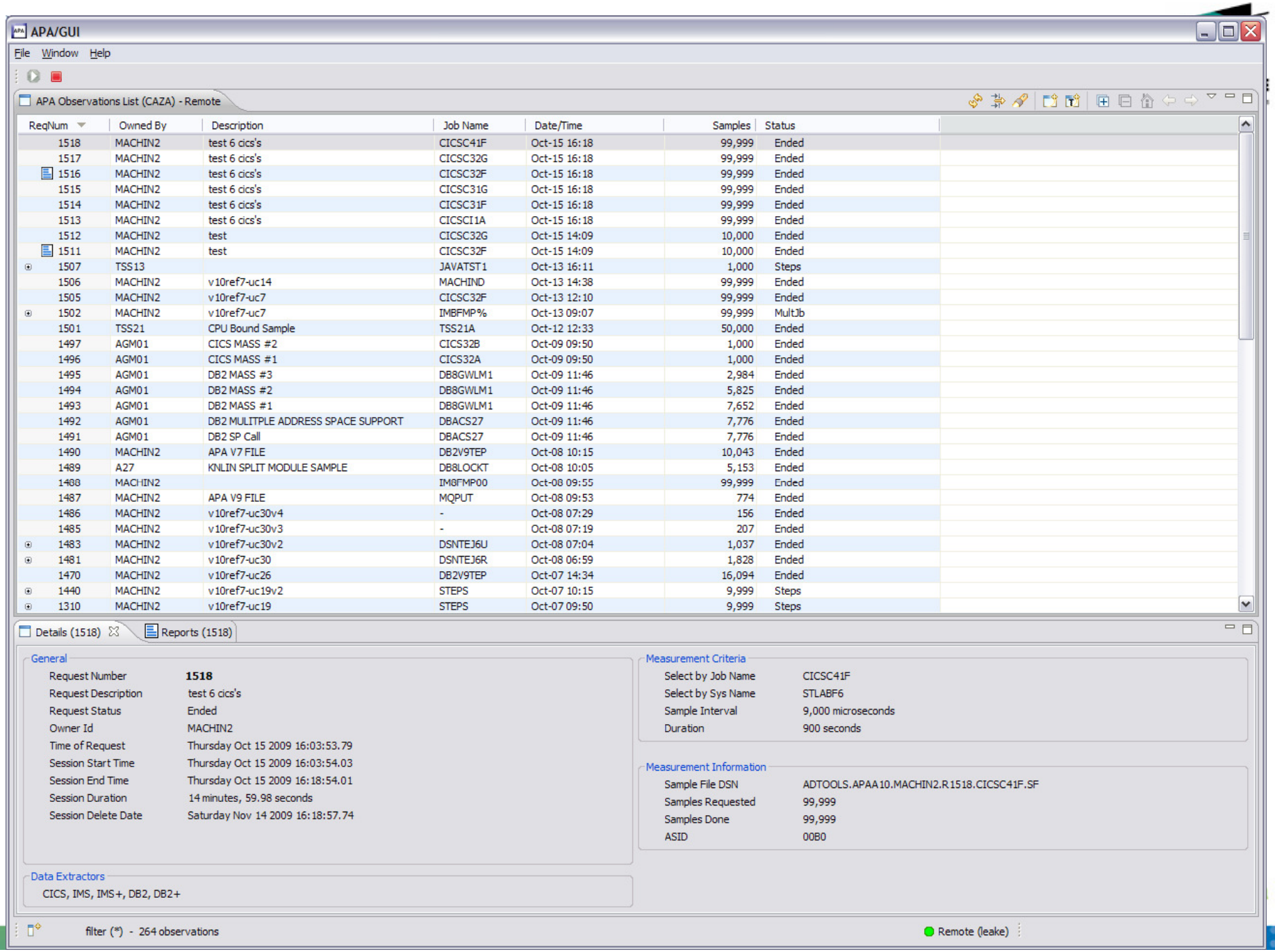
APA V11 Java Test Case (Java 5)



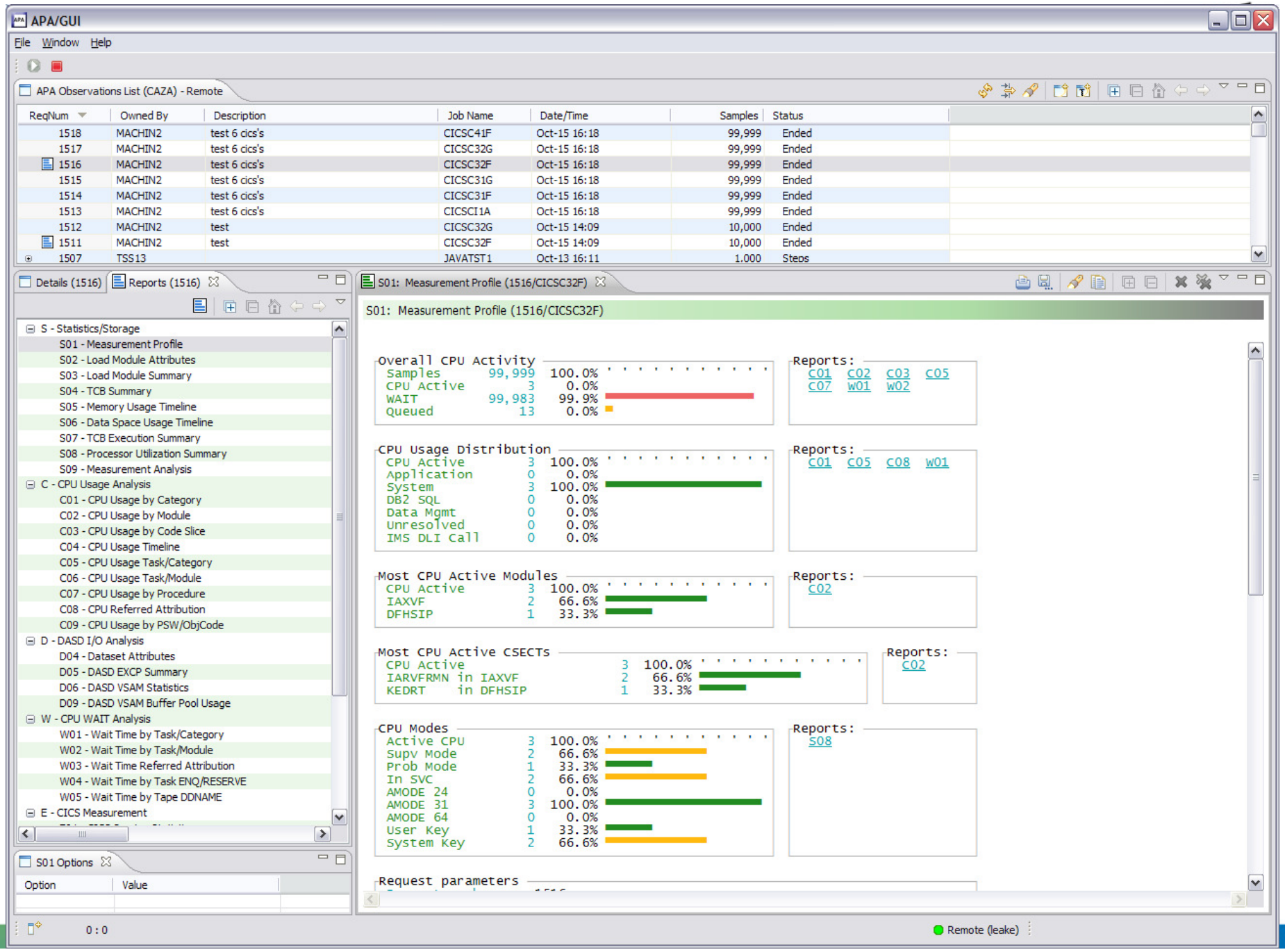
APA V11 Java Test Case (Java 5)



APA V11 Base User Interface

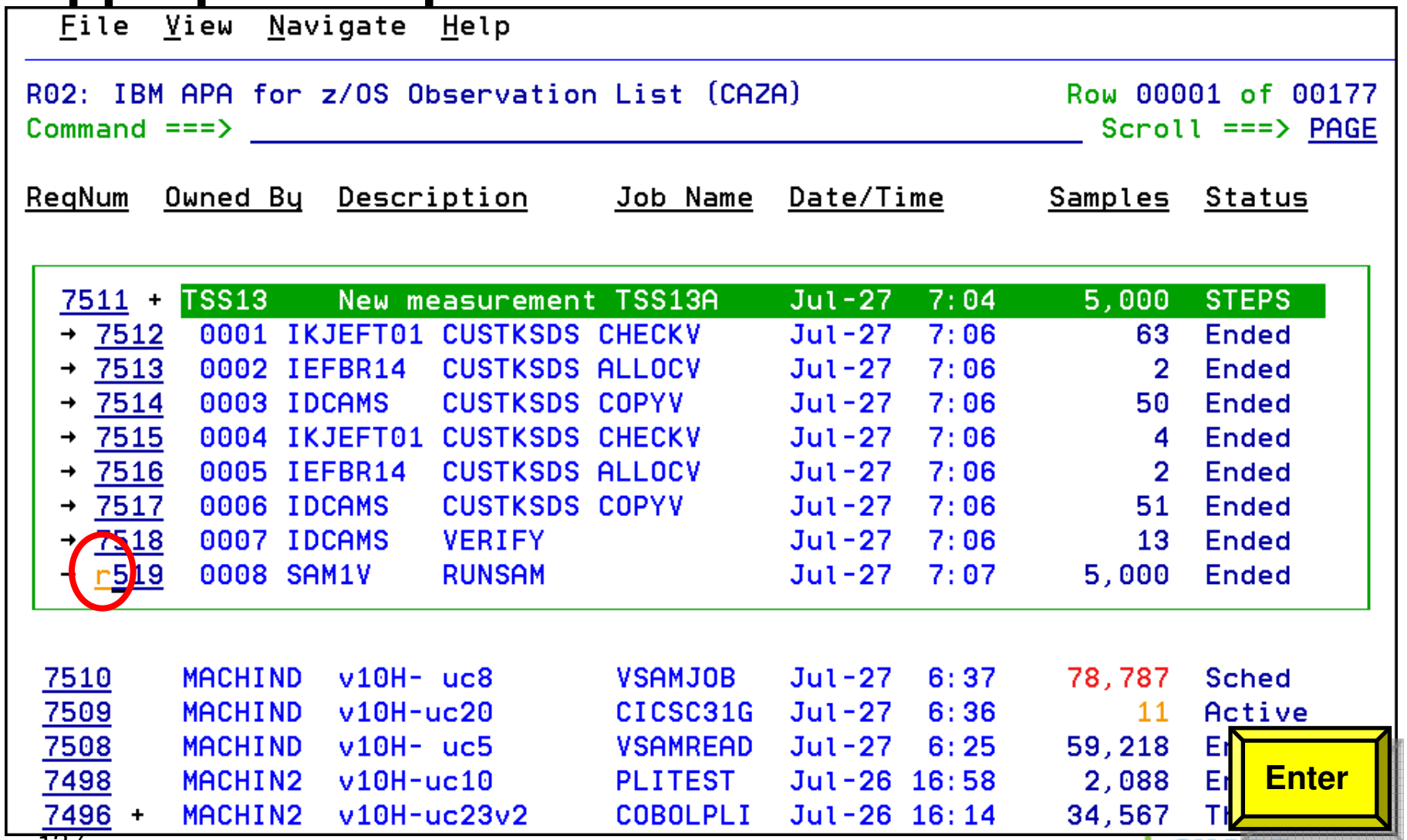


APA V11 CICS Test Case Sample



The Application Performance Analyzer ISPF Interface

Each step generates reports, select the appropriate report



File View Navigate Help

R02: IBM APA for z/OS Observation List (CAZA) Row 00001 of 00177
Command ===> Scroll ===> PAGE

ReqNum	Owned By	Description	Job Name	Date/Time	Samples	Status
7511 +	TSS13	New measurement TSS13A		Jul-27 7:04	5,000	STEPS
→ 7512	0001	IKJEFT01 CUSTKSDS CHECKV		Jul-27 7:06	63	Ended
→ 7513	0002	IEFBR14 CUSTKSDS ALLOCV		Jul-27 7:06	2	Ended
→ 7514	0003	IDCAMS CUSTKSDS COPYV		Jul-27 7:06	50	Ended
→ 7515	0004	IKJEFT01 CUSTKSDS CHECKV		Jul-27 7:06	4	Ended
→ 7516	0005	IEFBR14 CUSTKSDS ALLOCV		Jul-27 7:06	2	Ended
→ 7517	0006	IDCAMS CUSTKSDS COPYV		Jul-27 7:06	51	Ended
→ 7518	0007	IDCAMS VERIFY		Jul-27 7:06	13	Ended
→ r519	0008	SAM1V RUNSAM		Jul-27 7:07	5,000	Ended
7510	MACHIND	v10H- uc8	VSAMJOB	Jul-27 6:37	78,787	Sched
7509	MACHIND	v10H-uc20	CICSC31G	Jul-27 6:36	11	Active
7508	MACHIND	v10H- uc5	VSAMREAD	Jul-27 6:25	59,218	En
7498	MACHIN2	v10H-uc10	PLITEST	Jul-26 16:58	2,088	En
7496 +	MACHIN2	v10H-uc23v2	COBOLPLI	Jul-26 16:14	34,567	Th

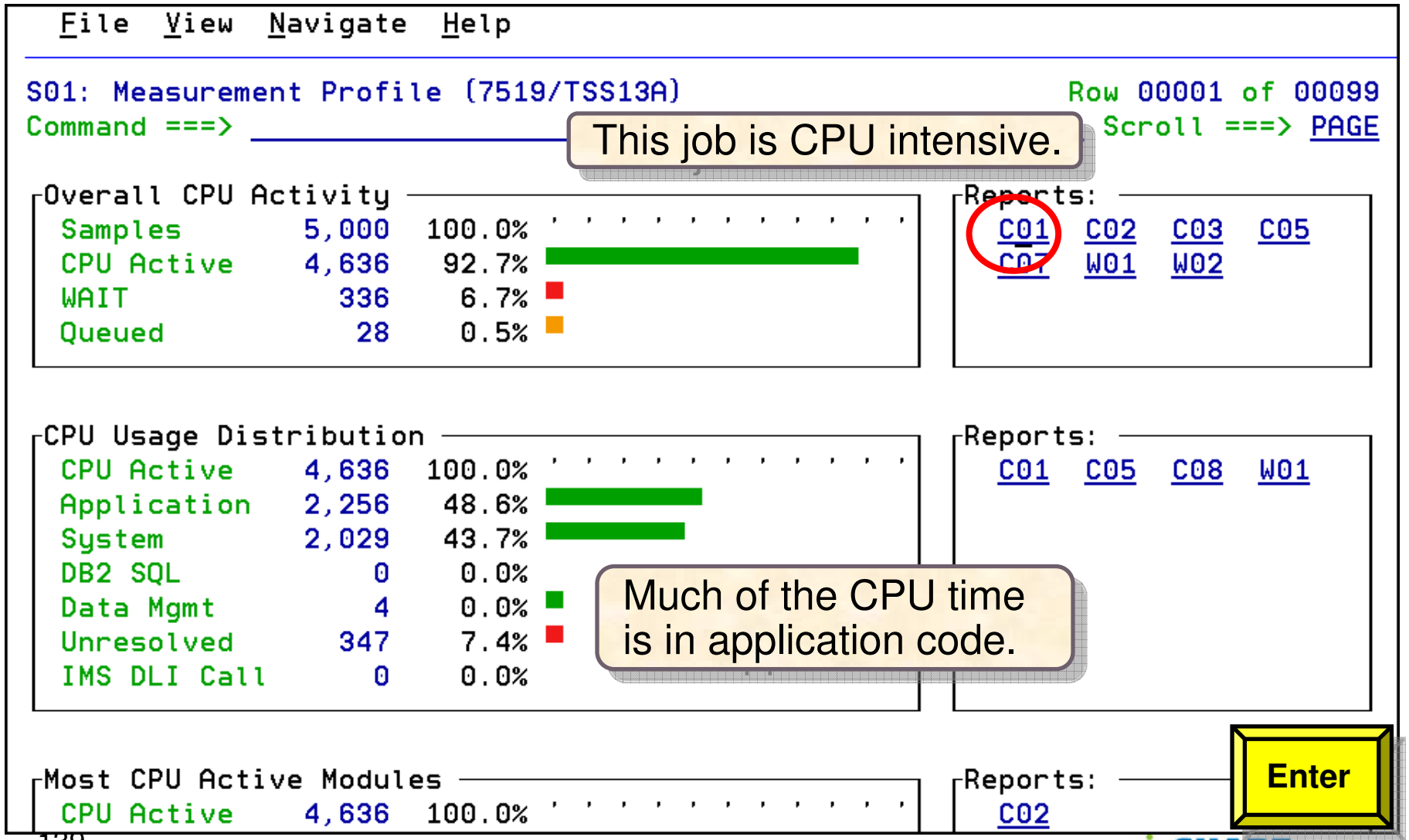
Enter

R01 Performance Report index is displayed Select S01 Measurement Report

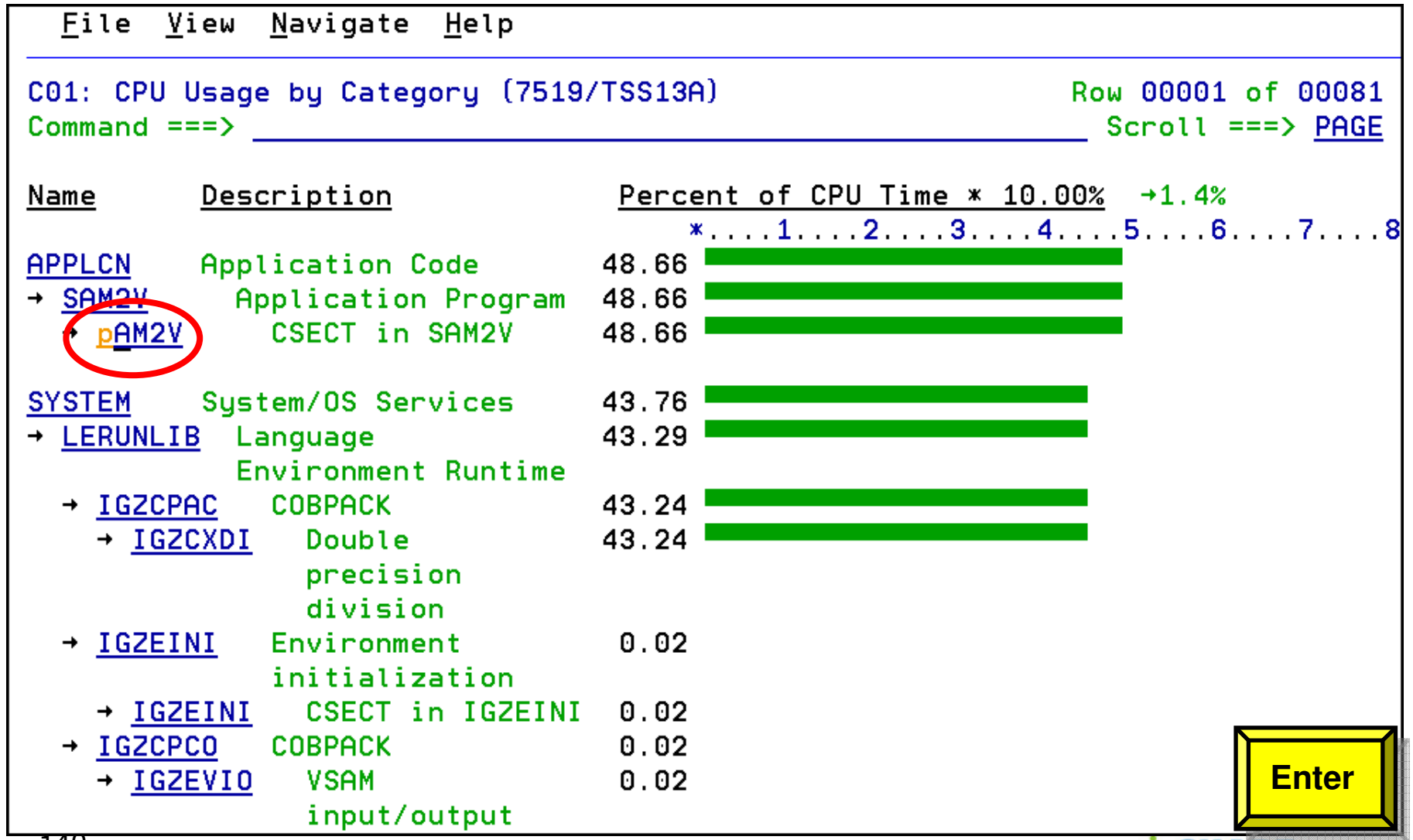

```
  File  View  Navigate  Help

R01: IBM APA for z/OS Performance Reports (7519/TSS13A)      Row 00001 of 00007
Command ===> _____________________________________________ Scroll ===> PAGE

  Select a category from      _ A Admin/Miscellaneous    _ I IMS Measurement
  the list to the right       _ S Statistics/Storage     _ E CICS Measurement
  to view the available       _ C CPU Usage Analysis     _ F DB2 Measurement
  reports in the selection    _ D DASD I/O Analysis      _ Q MQ Measurement
  list below.                 _ W CPU WAIT Analysis      _ G Coupling Facility
                              _ H HFS Analysis           _ J Java Measurement
                              _ V Variance Reports       _ X Multi Address Space

                                                                More:     +
Enter S to make a selection or enter the report code on the command line

  s S01 Measurement Profile                 _ S07 TCB Execution Summary
  _ S02 Load Module Attributes              _ S08 Processor Utilization Summary
  _ S03 Load Module Summary                 _ S09 Measurement Analysis
  _ S04 TCB Summary
  _ S05 Memory Usage Timeline
  _ S06 Data Space Usage Timeline
```

Enter

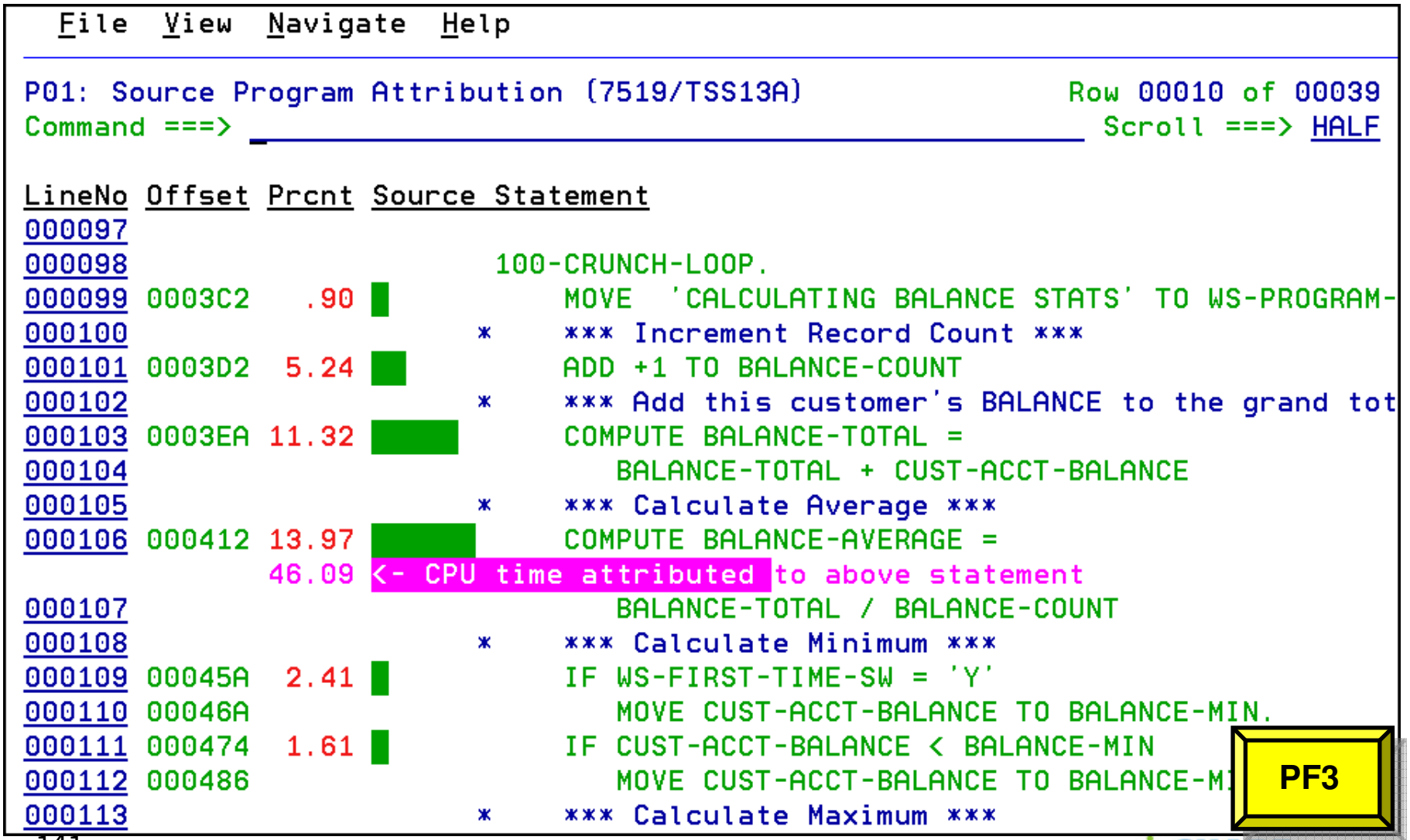
Measurement Profile report
Cursor select the C01 report



Use the ‘p’ line command to display program source for the SAM2V csect



The program statements that used the most CPU time are displayed Bar charts indicate statements using the most resources



Application Performance Analyzer: What's new in version 11?

- The IBM Application Performance Analyzer Plug-in for Eclipse, when integrated with Application Performance Analyzer for z/OS and CICS Explorer, encompasses both the Observation Request and Reporting functions, including the R02 screens list, detail views, edit functions, and reports for the observation. The plug-in GUI can be used for submitting new observation requests and for navigating the performance analysis reports generated from observation requests. The plugin GUI can display and provide functions to multiple components of Application Performance Analyzer at the same time. The major views include:
  - STC View, which lists all active started tasks
  - Observations List View, which lists all observations
  - Observation Detail View, which provides details of an observation
  - Reports List View, which lists all reports for an observation
  - Report View, which displays an individual report
- The Application Performance Analyzer Plug-in is enhanced to support IMS Multiple Address Space measurements, source program mapping, and to display windows for details of the sample file and the module information, to make it consistent with the ISPF panels.
- 64-bit support for xplink.
- 64-bit Java support

Application Performance Analyzer: What's new in version 11?

- CICS+ new CICS intercept extractor.
- New CICS reports are added to report mean and total service times by user, and to report CPU/service time by CICS transaction.
- New Java reports are added to report the Java heap usage timeline and Java CPU usage by thread.
- The "Variance Report" feature is improved by providing three new variance reports highlighting the main difference between the CICS, DB2, and IMS "summary reports." Up to 20 measurements can be selected for variance reporting, providing the ability to evaluate the performance of specific jobs over an extended time period.
- The DB2 CPU/service time reports are enhanced to allow the developer to display the percentage used in place of the mean fields.
- Support for the natural language and ADABASE from Software AG.
- Recommendation to the developer that a VSAM reorganization be done when CI and CA splits are present. This is shown in the Application Performance Analyzer S09 report.
- Support for large block size (greater than 262K).

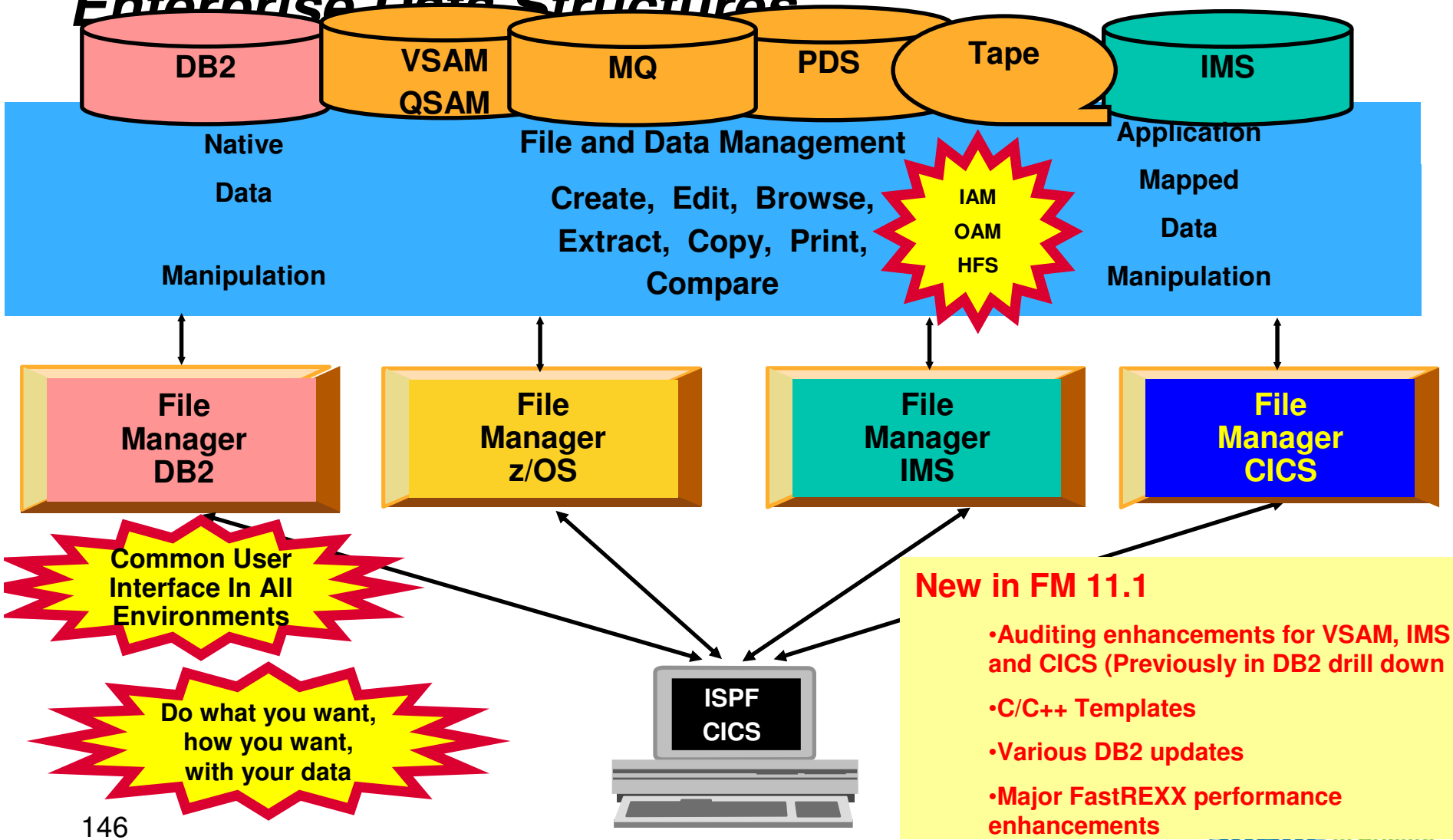
Application Performance Analyzer: What's new in version 11?

- A memory tracking exit that can include data about modules managed via directed load.
- Permit use of system symbols in SampleHLQ, and allow complete control of sample file naming convention via SampleDSN and DuplicateDSN configuration settings.
- C/C++ mapping time stamp interval is reduced.
- An option to load source and then map it to the module (the inverse of usual C/C ++).
- IMS Multiple Address Space Support that groups all IMS MPP region observations under a single parent observation in R02.
- IMS Multiple Address Space Support for IMSPlex.
- Support a shared source program mapping dataset list, called the Common Data Store.
- Support measuring Java jobs running in WebSphere V7.
- Currency support for DB2 V10.

Agenda

- Fault Analyzer
- Debug Tool
- Application Performance Analyzer
- File Manager

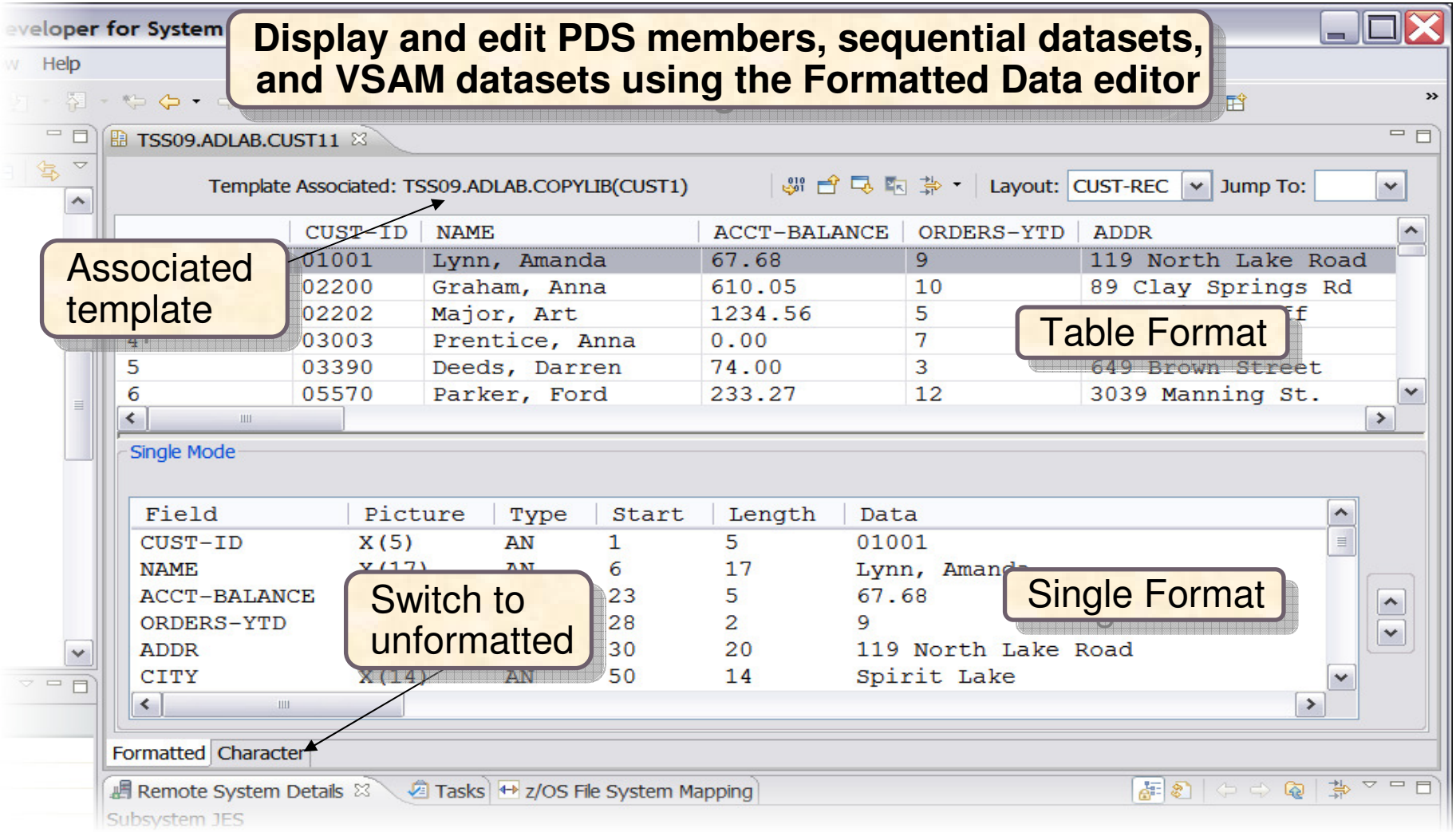
File Manager for z/OS (FM) – Manage a variety of Enterprise Data Structures



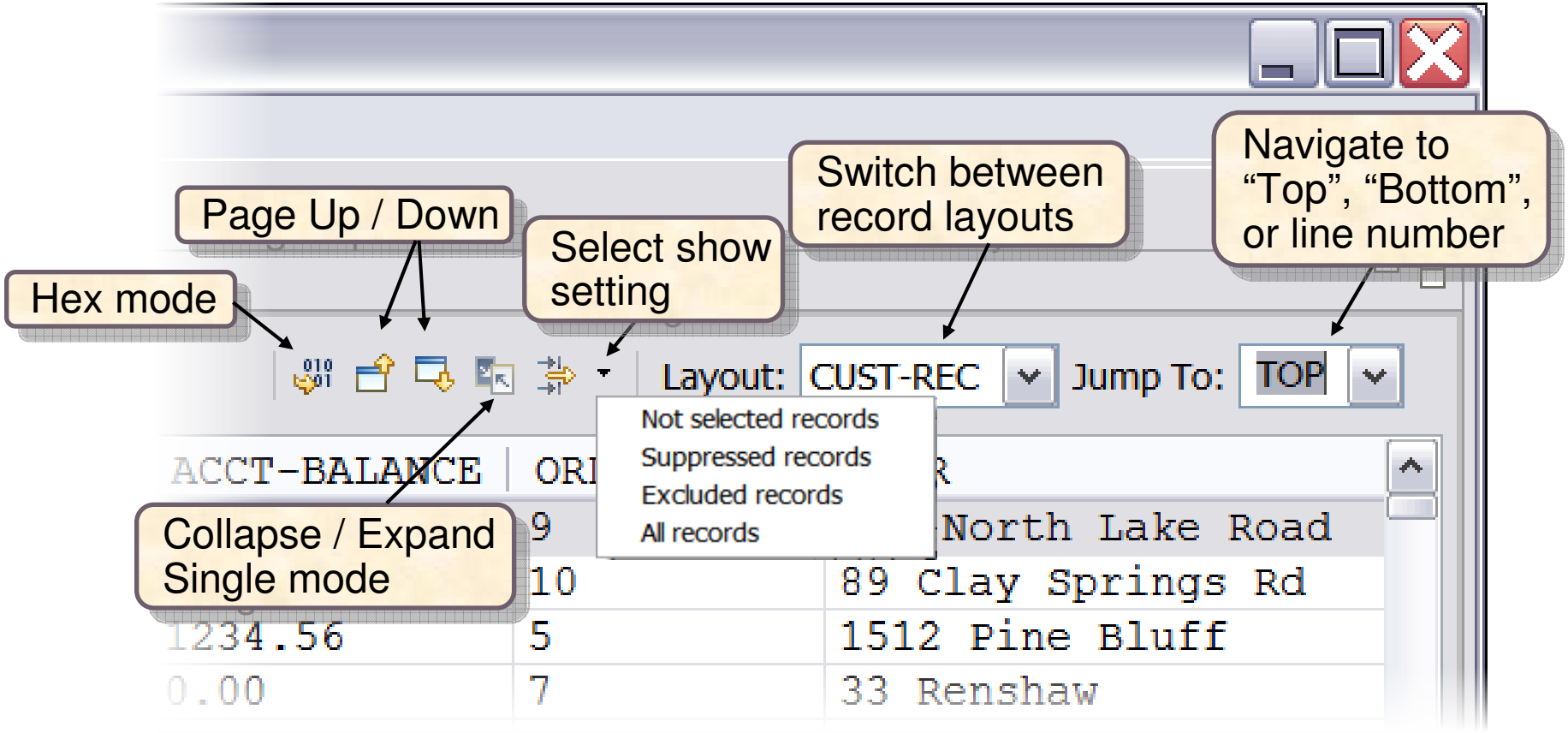
New in FM 11.1

- Auditing enhancements for VSAM, IMS and CICS (Previously in DB2 drill down
- C/C++ Templates
- Various DB2 updates
- Major FastREXX performance enhancements

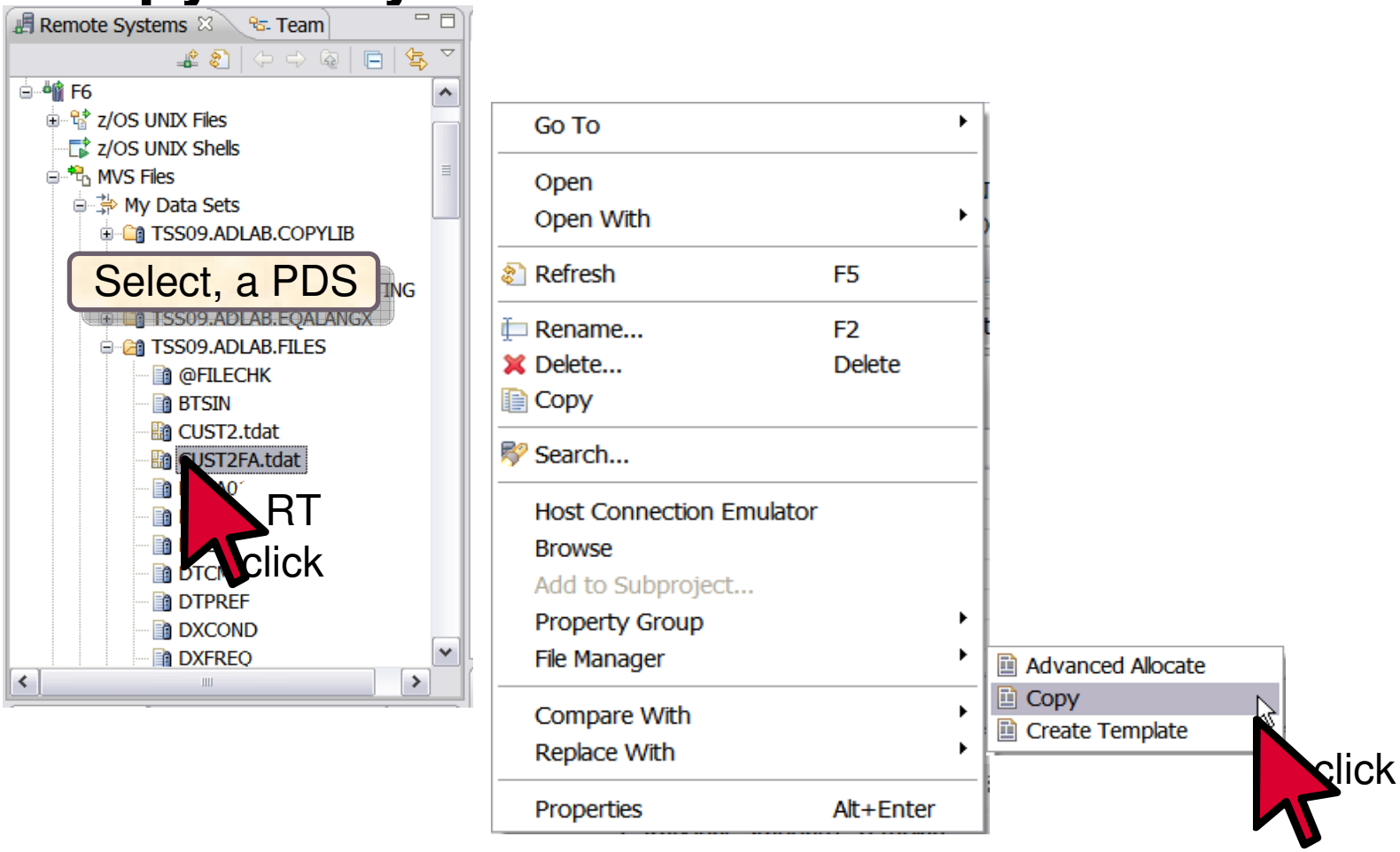
Formatted Data Editor



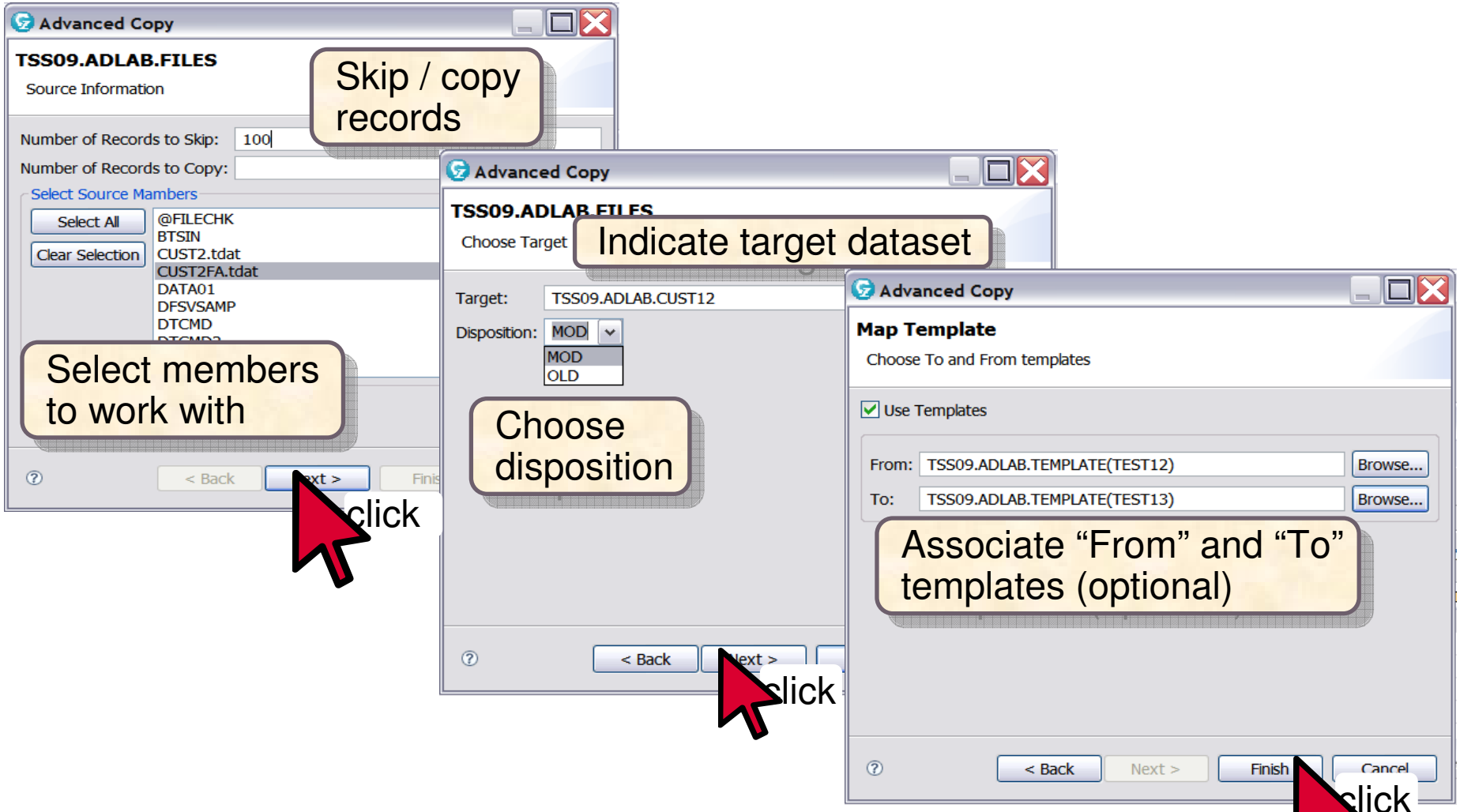
Formatted Data Editor Actions



Copy Utility

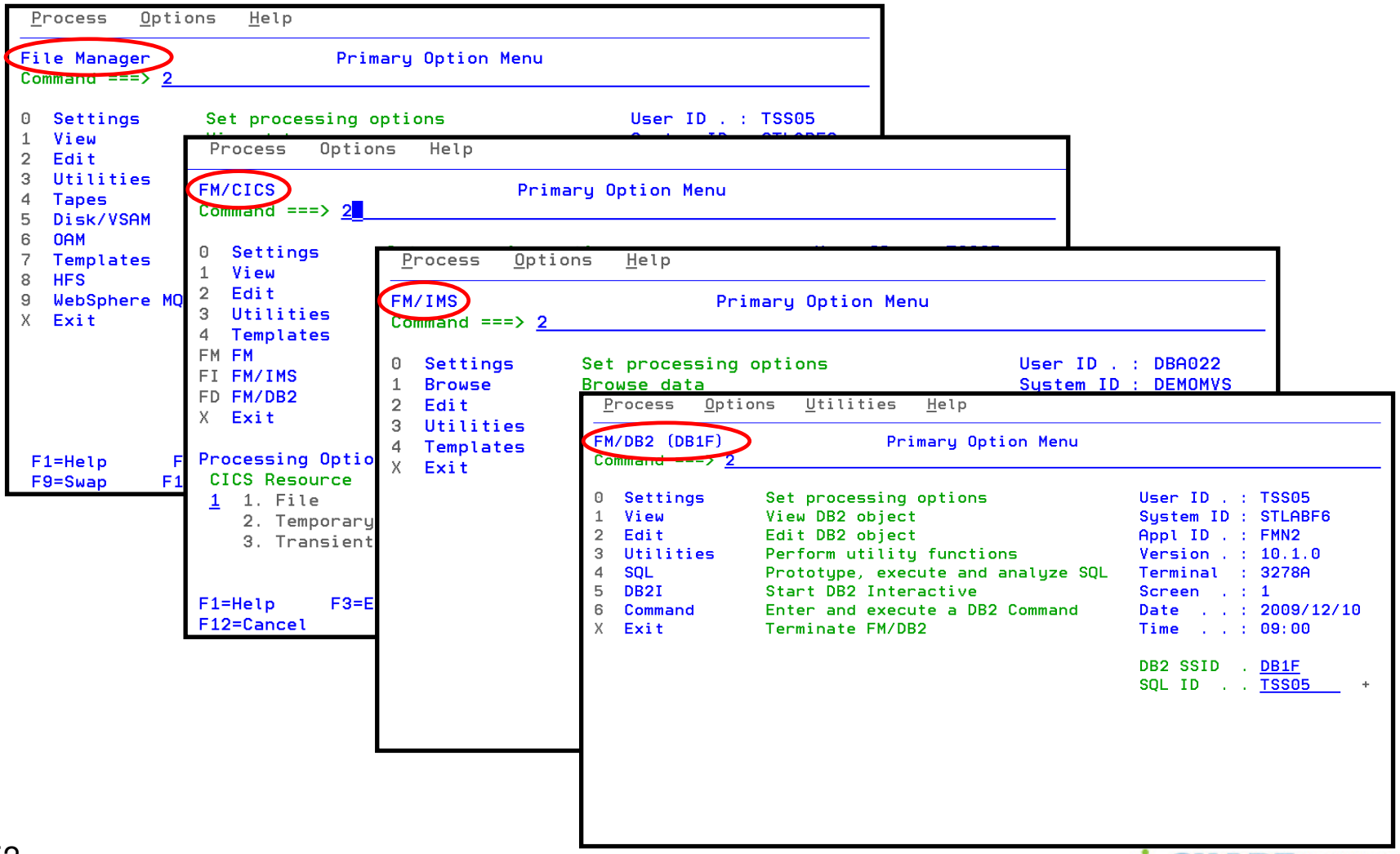


Copy Utility (continued)



The File Manager ISPF Interface

File Manager
Inclusive of all environments



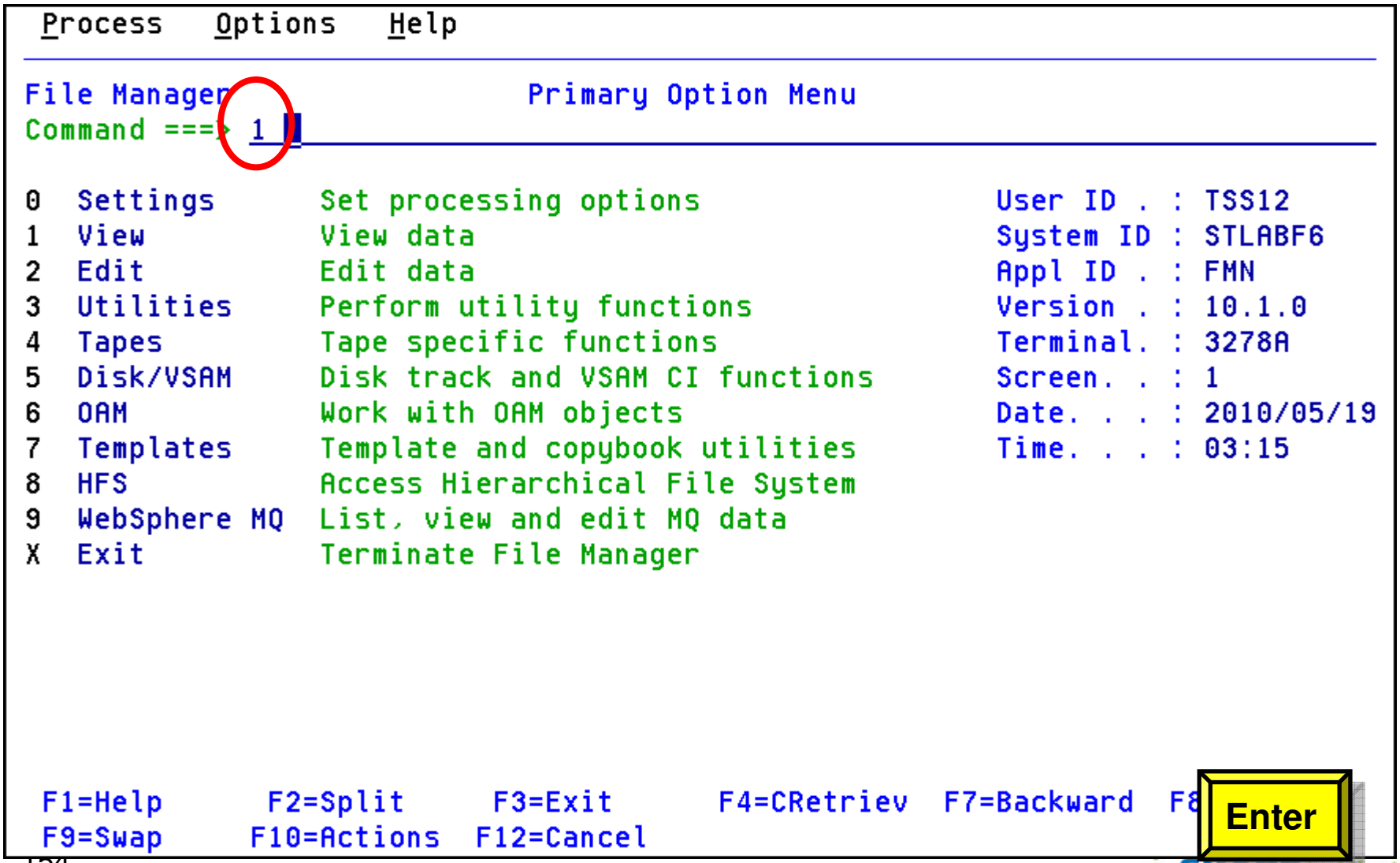
- Use a copybook or template to format the display
- A COBOL or PL/I copybook, or an assembler DSECT can be used

```
EDIT       TSS12.ADLAB.COPYLIB(CUST1) - 01.00
Command ===>
=COLS> ---1----+----2----+----3----+----4----+----5----+---
****** ***************************** Top of Data **********
000100 *   ---------------------------------------------
000200 *   Sample COBOL Copybook for IBM PD Tools Workshops
000300 *   Describes file <userid>.ADLAB.CUST1
000400 *   ---------------------------------------------
000500  01  CUST-REC.
000600      05  CUSTOMER-KEY.
000700          10  CUST-ID                PIC X(5).
000800      05  NAME                   PIC X(17).
000900      05  ACCT-BALANCE           PIC S9(7)V99  COMP-3.
001000      05  ORDERS-YTD             PIC S9(4)     COMP.
```

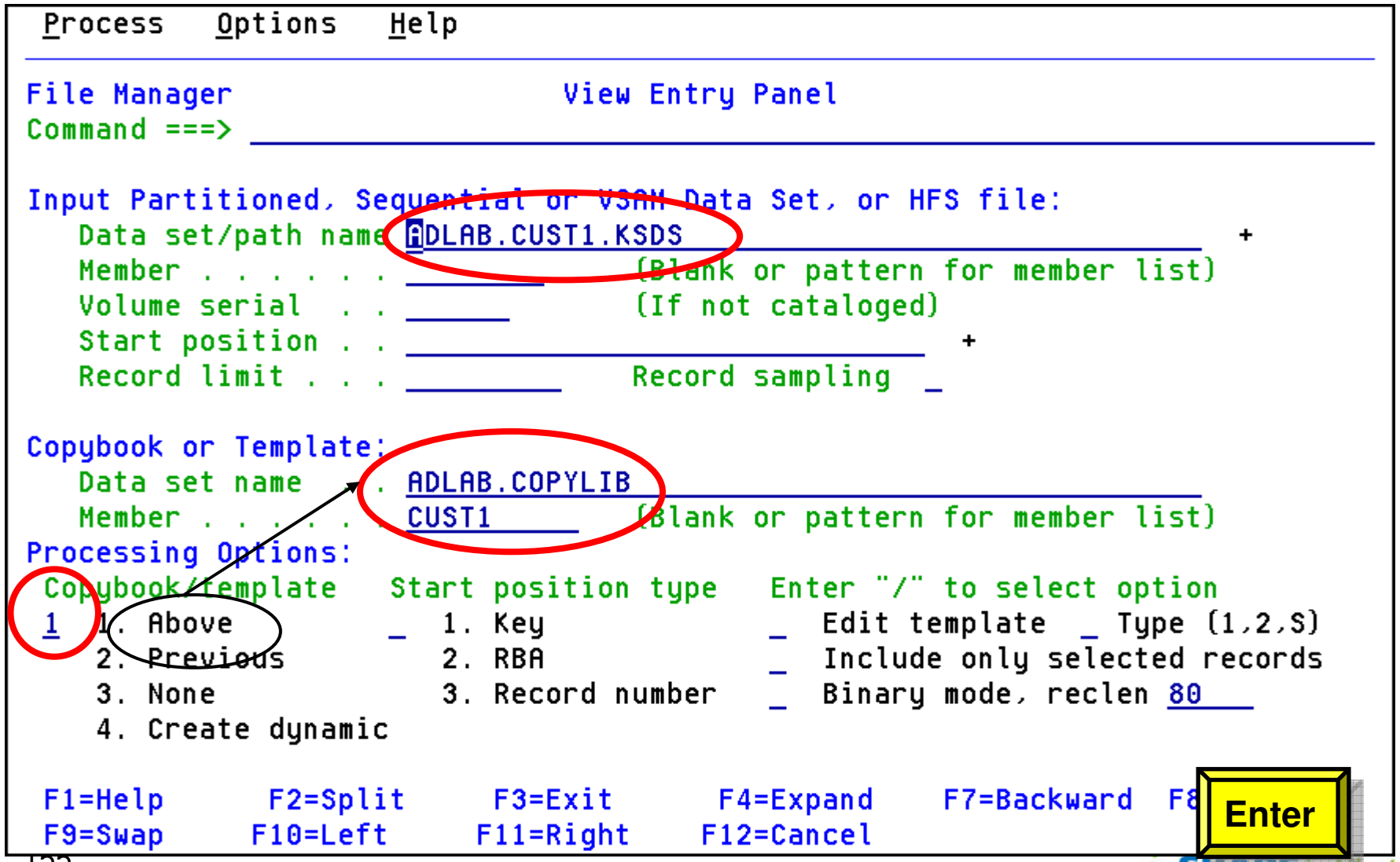
```
EDIT       TSS12.ADLAB.COPYLIB(PLCU1) - 01.00              Columns 000
Command ===>                                                Scroll =
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6---
****** ***************************** Top of Data **********************
000001  /*   ---------------------------------------------   */
000002  /*   SAMPLE PLI COPYBOOK FOR IBM PD TOOLS WORKSHOPS   */
000003  /*   DESCRIBES FILE <USERID>.ADLAB.CUST1              */
000004  /*   ---------------------------------------------   */
000005  DCL 1 CUST_REC,
000006        2 CUSTOMER_KEY,
000007          3 CUST_ID          CHAR(5),
000008        2 NAME               CHAR(17),
000009        2 ACCT_BALANCE       FIXED DEC (9,2) UNALIGNED,
000010        2 ORDERS_YTD         FIXED BINARY(15,0) SIGNED UNALIGNED,
```

```
EDIT       TSS12.ADLAB.COPYLIB(ASCU1)
Command ===>
=COLS> ----+----1----+----2----+----3
****** ******************************
000001 *   ---------------------------
000002 *   SAMPLE ASSEMBLER DSECT FOR
000003 *   DESCRIBES FILE <USERID>.AD
000004 *   ---------------------------
000005 CUST_REC   DSECT
000006 CUST_ID    DS CL5
000007 NAME       DS CL17
000008 ACCT_BAL   DS 5D
000009 ORDS_YTD   DS 4H
```

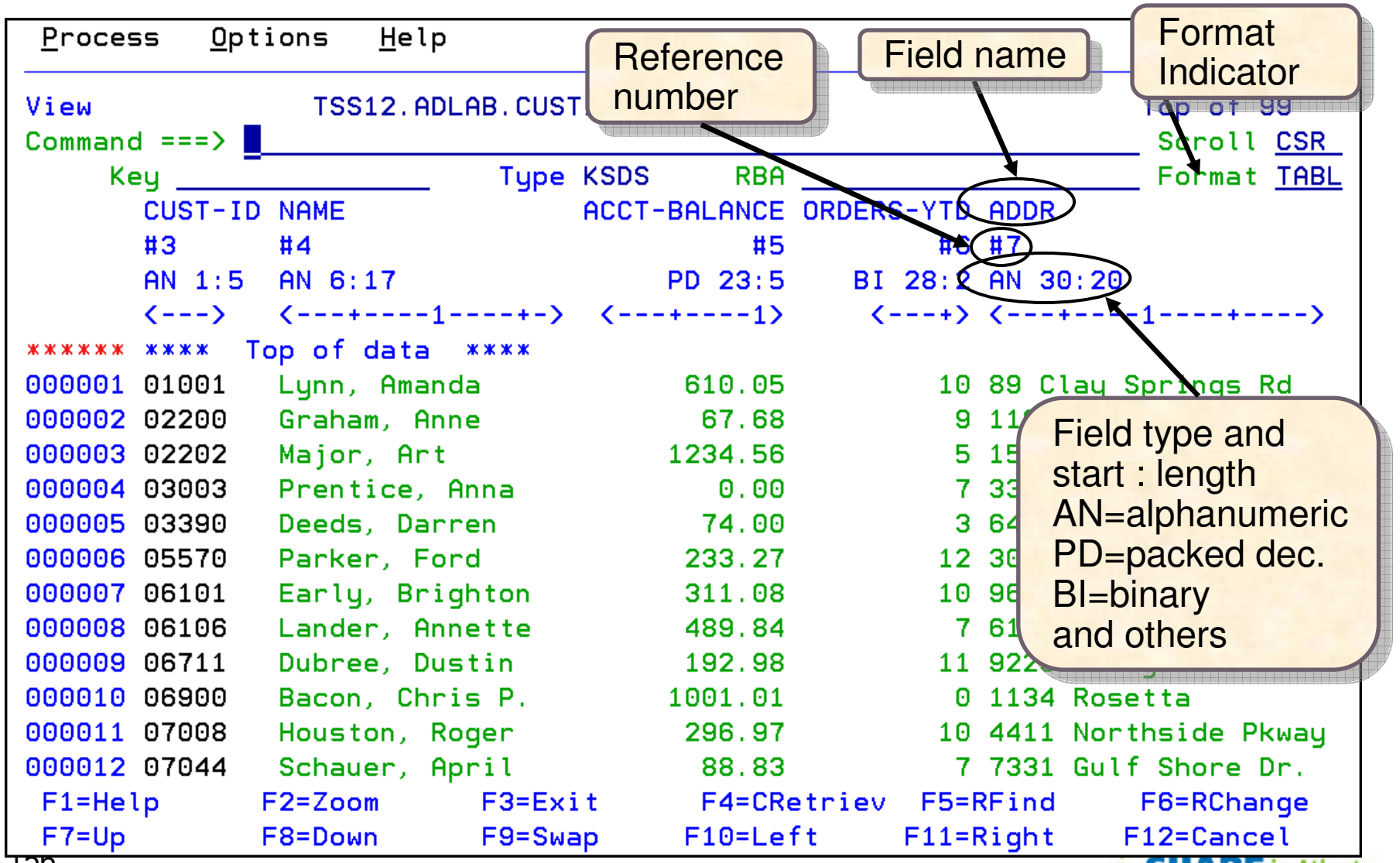
Option 1 displays the view entry panel



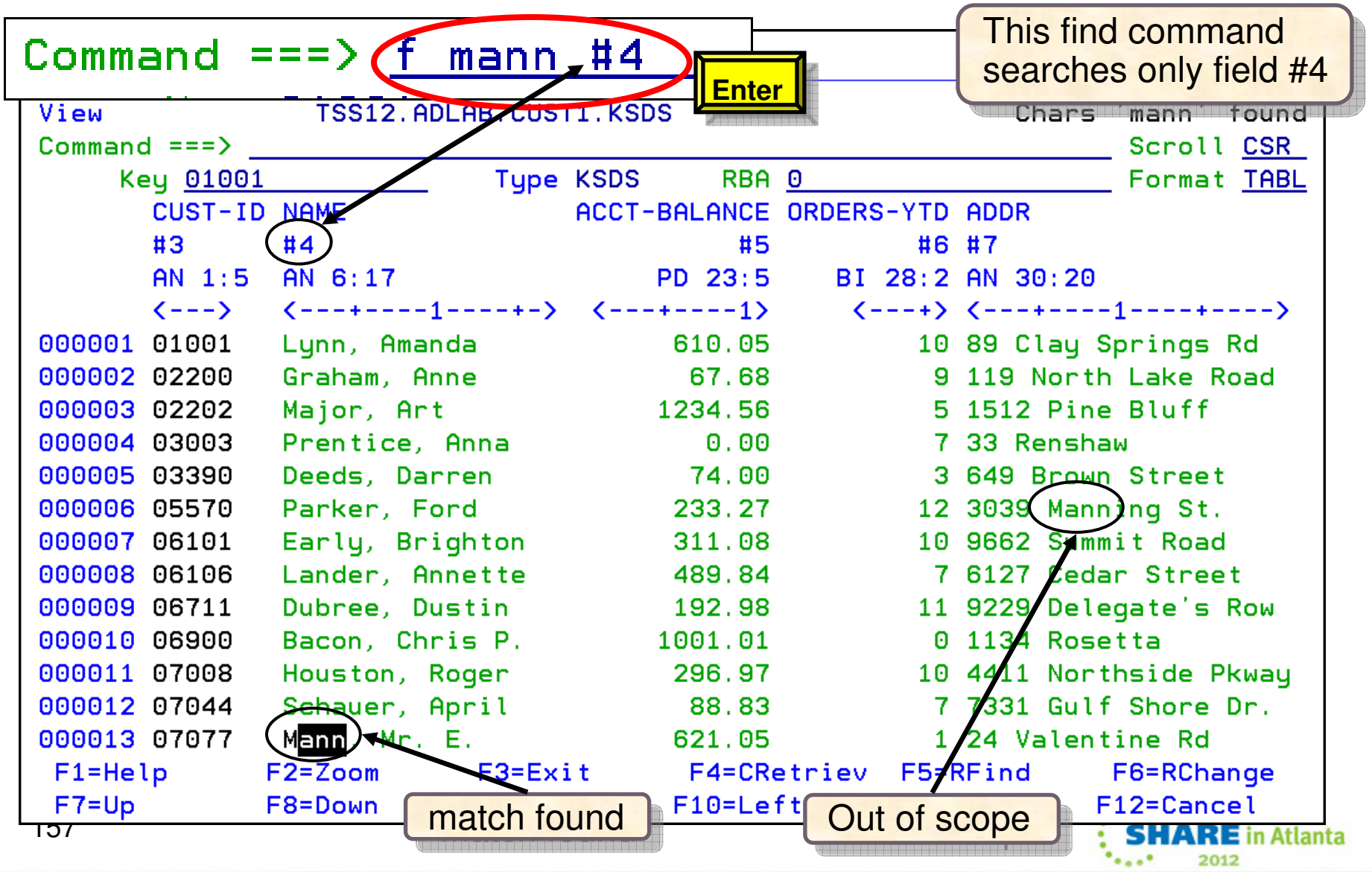
Enter the name of the copybook, and specify the 1 (above) option



Records are formatted showing fields defined in the copybook



Some commands can use the field reference number



Utilities

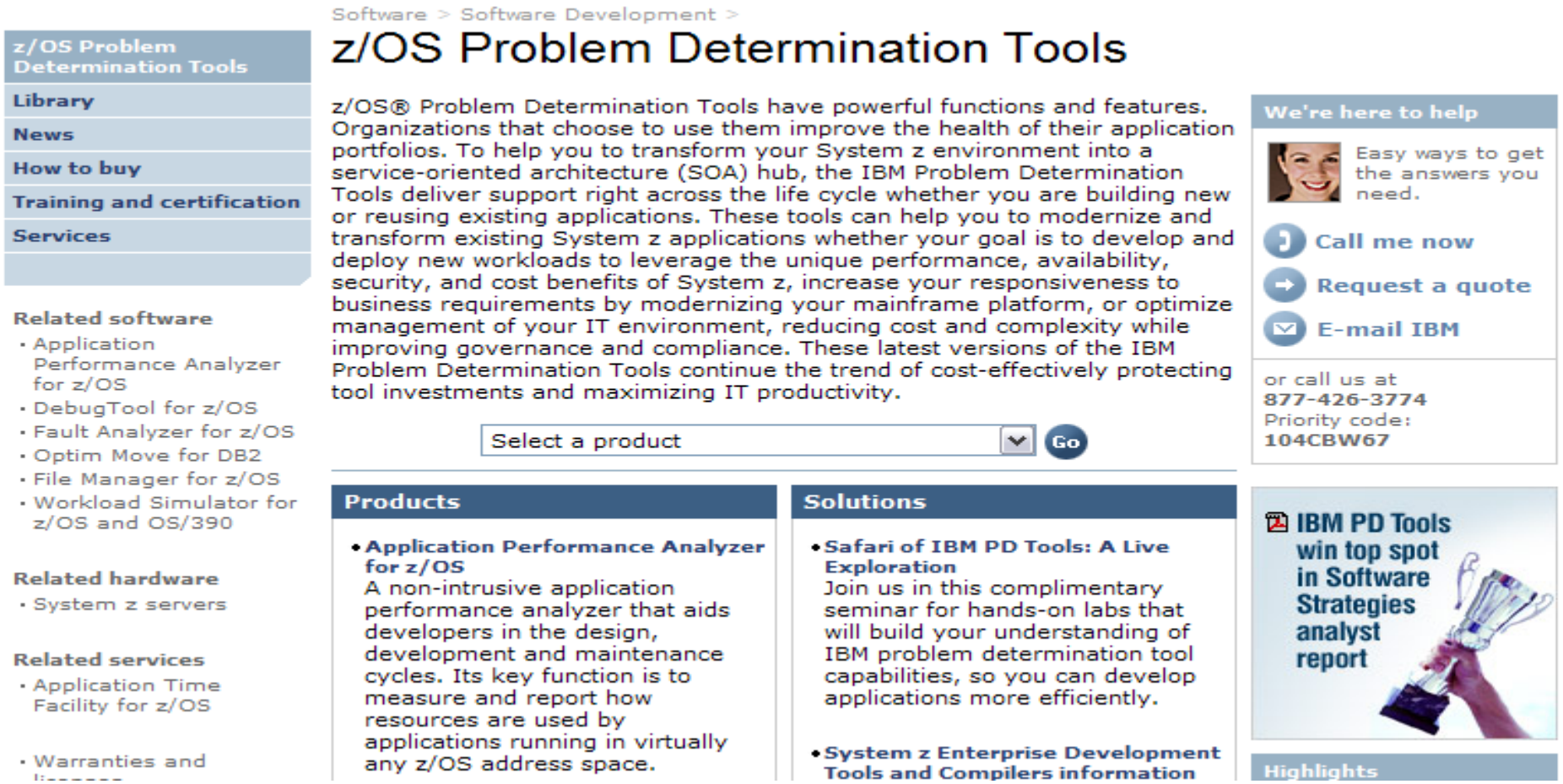
- Search for/change Test data generation
  - Generate new data based on existing copybooks
- Copy data
  - Reformat and generate data while copying
  - Field values can be "**scrambled**" to protect sensitive data
  - Copy data into XML format
- Global Find/Change data across members in a PDS(E)
  - Perform a new search based on the results of a previous search
- Compare
  - Compare records/fields between files
  - Use field level mapping for comparison criteria
  - Special options for load module comparisons
- Websphere MQ support

File Manager: What's new in version 11?

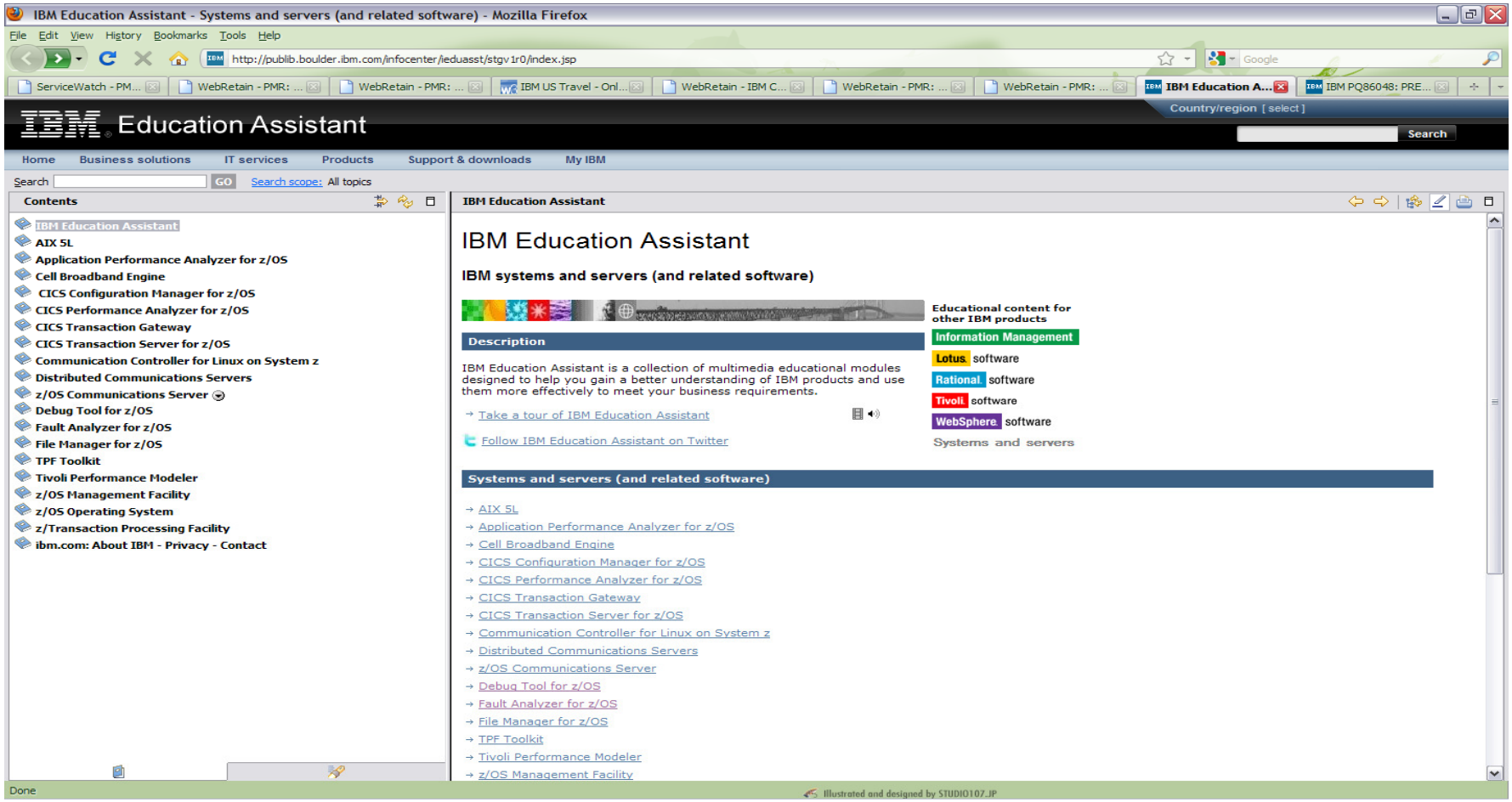
- File Manager Editor and Batch performance is improved with enhanced expression processing which includes:
  - Improved performance for the File Manager Base Editor with REXXTM template expressions
  - Improved performance for the File Manager Base Utility functions with REXX procedures
  - Additional programming constructs supported in FASTREXX
  - Ability to access files in read only mode for Data Set Edit Batch (DSEB) and to run DSEB procedures using FASTREXX
  - IImproved expression capability for segmented data
- The File Manager Base is enhanced to work with Rational® Developer for System z V7.6.2 to enable compiler option processing.
- Auditing enhancements for the File Manager Base, CICS, and IMS components. The enhancements are built on the facility provided in the File Manager DB2 V10 component.
- Support for DB2 export to produce column names in CSV format as well as supporting DBCS characters.
- Support for creating and updating the DB2 template in batch.
- Support for setting commit conditions for FM DB2 edit and import.
- Usability enhancements:
  - Member list panel enhancements to support COPY, DELETE, MOVE, PRINT, RESET, and SUBMIT commands
  - Allocation of template data sets when the specified data set does not exist

Get more information about these tools at:

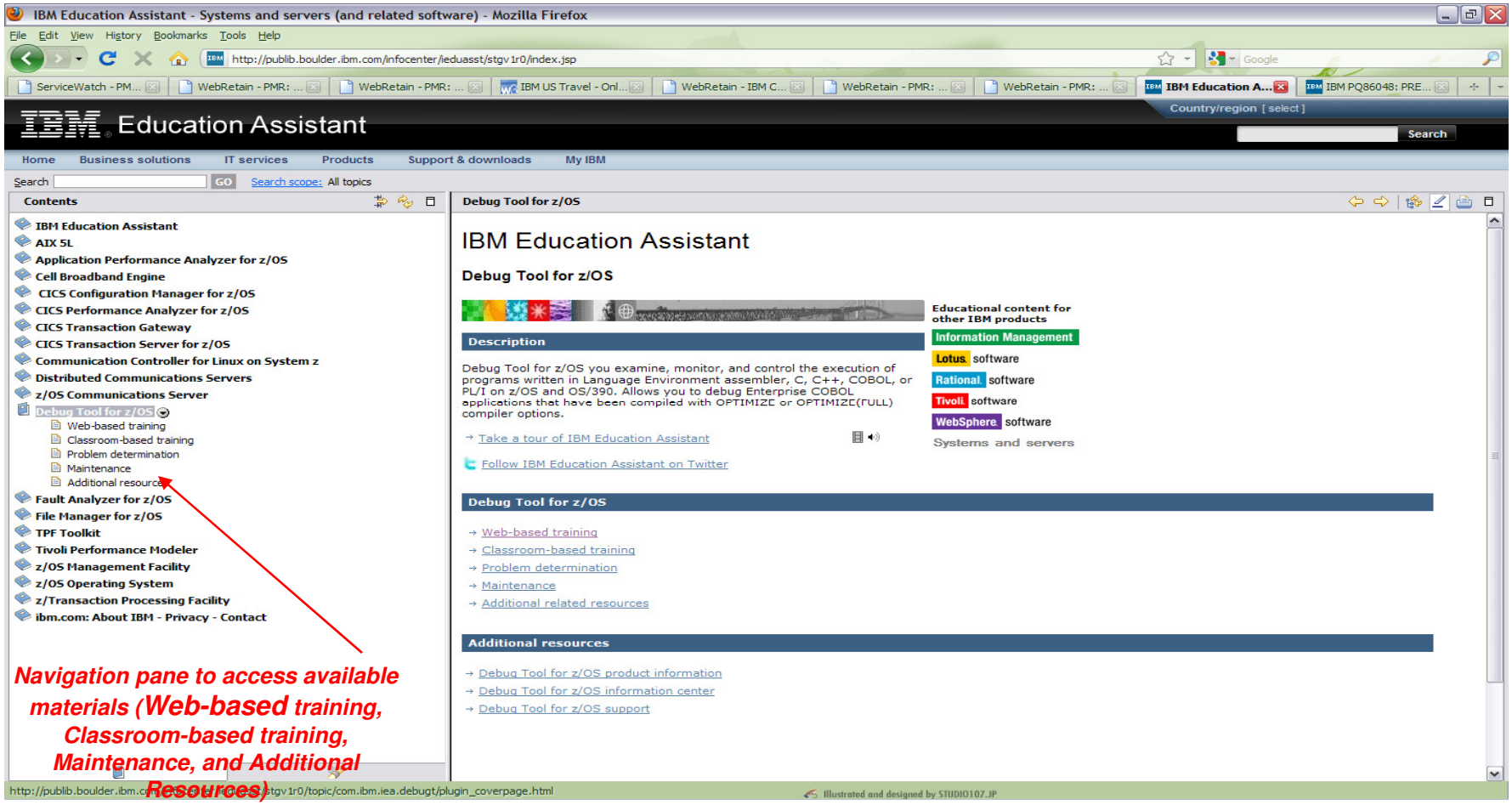
www.ibm.com/software/awdtools/deployment



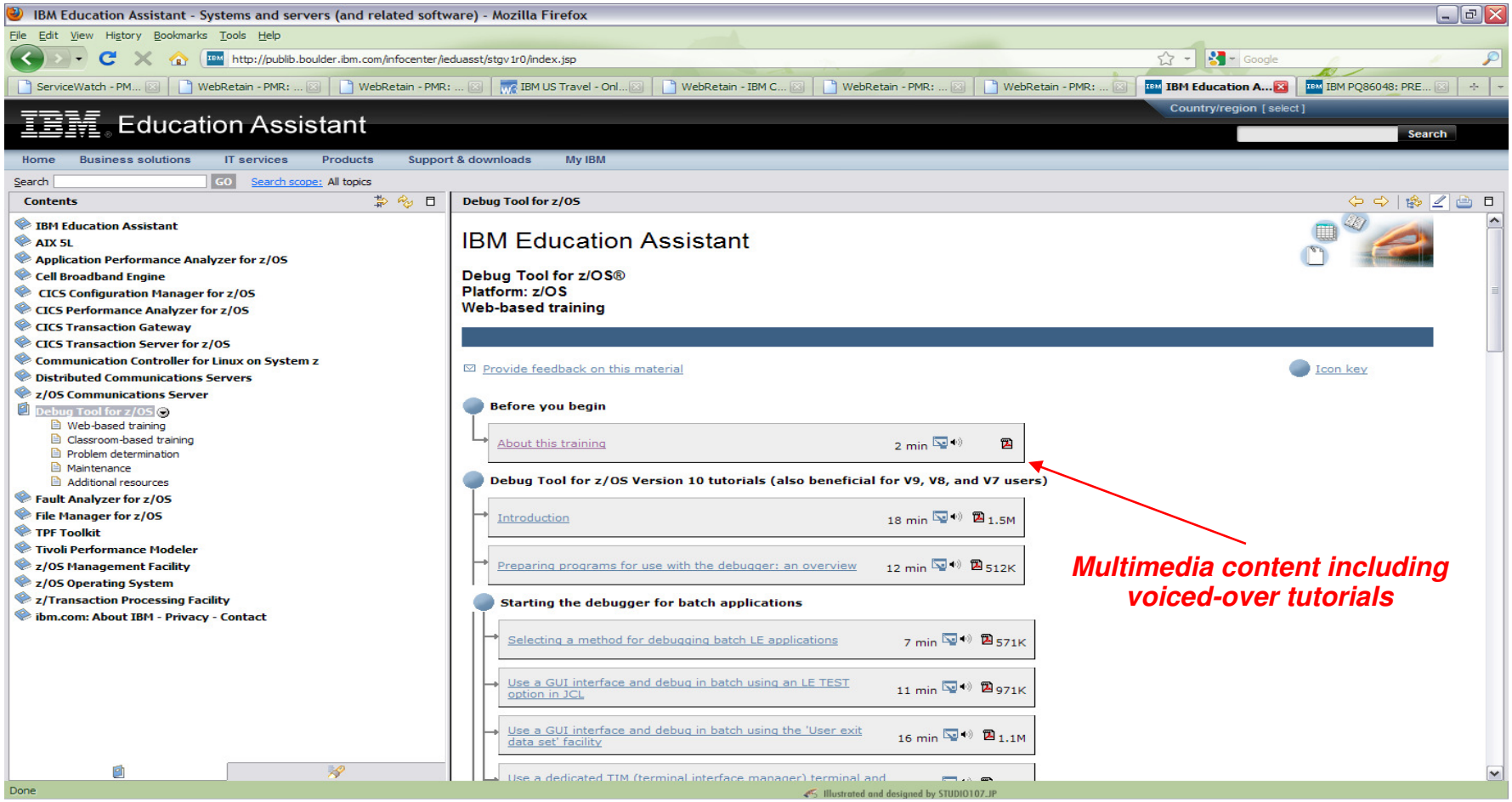
IBM Education Assistant
http://publib.boulder.ibm.com/infocenter/iduasst/stgv1r0/index.jsp



IBM Education Assistant
A Drill Down



The Education Assistant
More Information



We love your Feedback!

- Don’t forget to submit your SHARE Session and Speaker feedback! Your feedback is very important to us, we use it to improve our conference for you next year.


Thank
You

#SHAREorg

Copyright and Trademarks