What's New In the IBM Problem Determination Tools

Francisco M Anaya
IBM Problem Determination Tools Architect

Session 15458: What's New In the IBM Problem Determination Tools
August 8, 2014
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
- Application Performance Analyzer Enhancements
- File Manager Enhancement
- Fault Analyzer Enhancements
- Debug Tool Enhancements
- New Fault Analyzer COBOL Explorer
- Debug Tool IMS Private Message Region Creation and Dynamic rerouting of transactions.
IBM Problem Determination Tools Products

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 set: 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.
Comprehensive z development environment

- **Choice of development environments:** PD Tools Eclipse GUIs are provided all inclusive and can be used in conjunction with other IBM development tools.

- **Rational Developer for System z:** Full Integrated Development Environment for mainframe application developers. When used in conjunction with PD Tools, developers can expect productivity improvement for writing programs but also faster problem diagnosis and resolution.

- **IBM Worklight:** provides a comprehensive solution for developing, deploying and the server infrastructure for enabling access to existing assets on z (System of Records).
  - PD Tools continue to be important for problem diagnosis while accessing existing assets on z.
  - PD Tools Eclipse GUIs can be installed into Worklight Studio.
  - New browser-based interface for Fault Analyzer so mobile/web developers can access abend information.

- **Continued support for 3270 based users:** many application developers continue to rely on 3270-based interface for developing and maintaining mainframe applications today. PD Tools continue to support those users as well as offering additional productivity gain through products like Data Set Commander..
Exploitation of GUIs made Simple!

Simplicity
- All PD Tools GUIs talks to PD Tools Common Server (a quick to start and extensible!).
- Additional capabilities added to the server by a simple configuration script (APA V13.1 exploits this feature).

Flexibility
- Select capabilities needed to create your own workbench.
- Select capabilities you need from a single update site.

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Java on z/OS? It’s real – Mobile, SOA, batch, modernization…

Debugging legacy and Java workload together – stepping through both programs smoothly.

Understand WAS resource consumption and WAS initiated workload together – reports call count, response time and more for CICS DPL calls and DB2 SQL calls.

Diagnosing application failures involving legacy and Java workload.

<table>
<thead>
<tr>
<th>Event #</th>
<th>Type</th>
<th>Fall Point</th>
<th>Module Name</th>
<th>Program Name</th>
<th>EP Name</th>
<th>Event Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Call</td>
<td>n/a</td>
<td>DBXLINPA</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>From LPA</td>
</tr>
<tr>
<td>2</td>
<td>Call</td>
<td>n/a</td>
<td>CEEBINIT</td>
<td>n/a</td>
<td>CEEORCMN</td>
<td>E+96A</td>
<td>POSIX Minmain routine; From LPA</td>
</tr>
<tr>
<td>3</td>
<td>Call</td>
<td>n/a</td>
<td>CEEPLPA</td>
<td>n/a</td>
<td>CEEEXRMD</td>
<td>E+1200</td>
<td>Run on down stack swap; From LPA</td>
</tr>
<tr>
<td>4</td>
<td>Call</td>
<td>n/a</td>
<td>Java</td>
<td>n/a</td>
<td>JavaMain</td>
<td>E+138C</td>
<td>From /u/dlpgcr/javas.8.1-31/j8s.8.1/bin/</td>
</tr>
<tr>
<td>5</td>
<td>Call</td>
<td>n/a</td>
<td>JavaMain.main</td>
<td>n/a</td>
<td>JavaMain.main</td>
<td>L95</td>
<td>From not determined</td>
</tr>
<tr>
<td>6</td>
<td>Call</td>
<td>n/a</td>
<td>libCobolBase.so</td>
<td>CEEVROMU</td>
<td>CEEVROMU</td>
<td>E+107E</td>
<td>CEL Common Runtime; From LPA</td>
</tr>
<tr>
<td>7</td>
<td>Call</td>
<td>n/a</td>
<td>libCobolBase.so</td>
<td>CEEVROMU</td>
<td>CEEVROMU</td>
<td>E+107E</td>
<td>CEL Common Runtime; From LPA</td>
</tr>
</tbody>
</table>
Problem Determination Tools V13.1 Highlights

Key focus on Optimization and Modernization
PD Tools capabilities are continued to be enhanced to meet the requirements of problem diagnosis scenarios in the mainframe operating environments today and future. We are continuing to invest in supporting 3270 terminal based users while we are making significant improvements for Eclipse-based graphical user interface (aka PD Tools Studio) for those who are taking advantage of modern integrated development environment.

Simplification
Continuing to simplify the deployment of our products. With PD Tools V13.1, 1) all GUI capabilities are supported using the PD Tools common server technology which enables quick deployment of PD Tools Studio, 2) File Manager provides XML-based APIs for creating templates to simplified the preparation of templates which describe your data layout, and 3) Fault Analyzer side-file mismatch reporting capability for identifying programs with wrong side-file information for analysis just to name a few.

Support for the latest z/OS, subsystems and languages
Continuing to support the latest level of z/OS (V2.1), languages (COBOL V5.1, PL/I V4.4, C/C++ for z/OS V2.1, Java V7, and HLASM) and subsystems (CICS V5.1, DB2 V11, IMS V13, MQ 7.5, and WASz 8.5).

Other enhancements
• Improved code coverage reporting.
• GUI for working with IMS databases.
• Enhanced debugging capabilities for mixed workload and IMS transactions.
• Browser-based access to Fault Analyzer reports.

Announcement: October 1, 2013
eGA: October 18, 2013

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
IBM Debug Tool Version 13 (Announce: October 1, GA: October 18)

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
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

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

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

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Application Performance Analyzer: Features added since V13 GA

- Enhancement to CAZBATCH and CAZAPI01 to allow the user to provide a specific sample file data set name using the new keyword 'SampDSN'. This addresses PMR55254.

- New S10 Error Messages report.

- New configuration setting 'ErrMsgStatus' that is used to indicate what conditions the ErrMsg status is to be displayed in the observation list.

- Enhancement to the C09 report, to display up to three possible interpretations of the instruction before the PSW. These are displayed under the heading 'Alternate Disassembly'

- Enhancement to enforce user level security in APA when creating, updating, reading and deleting sample file data sets, and those data sets containing exported sample files. This feature introduces a new APA started task configuration setting 'SampleDSUserLevelSecurity', and new APA PDTCC extension configuration settings 'SecuritySuppMsg' and 'SampleDSUserLevelSecurity'. Establish RACF user environment in APA started tasks, as required to support datasets with &USERID HLQ. This addresses RFE 512.

- Enhancement to the configuration setting AltCICSStepNm to allow six alternate CICS execution module names.

- Enhancement to support reporting for Parallel Access Volumes. This addresses RFE 632.

- New DB2I configuration setting ABENDSTOP, used to indicate if APA is to stop or continue sampling after DB2 intercept and traps an abend.

- Support for the CICS TS 5.2
File Manager: Features added since V13 GA

- For FM/IMS, session timeout support - via PI09504. FM/IMS has been modified to provide two new installation options a time-based check pointing option and a timeout option. These options provide two alternative ways of resolving resource contention problem.
- For FM/DB2, Clone and History table support - via PI05860. File Manager DB2 component has been updated to support edit of DB2 clone and history tables.
- DB2 Editor Time Out support – via PI19147. A new FMN2SSDM macro option EDITOR_TIMEOUT is provided, which can be specified for those DB2 systems where an editor inactivity time-out is required. EDITOR_TIMEOUT specifies whether FM/DB2 editor edit sessions are subject to an inactivity time-out, and the length of time that must elapse before the inactivity time-out is triggered. The inactivity time-out applies to edit sessions, not to view or browse.
- Support inserting and deleting segments when editing data using a segmented template – via PI04887.
- Improve performance of copy with reformatting – via PI07105. This APAR significantly improves reformatting performance especially for packed to zoned field type conversions and for when a large number of fields are present and most fields are copied as-is.
- **FM Plugin**
  - Support files with large number of columns in formatted editor by implementing segmented columns.
  - Support browsing files with large record length (more than tens of thousands) by implementing segmented records.
  - Adding 'Structure Mode' to Single Record view.
Fault Analyzer: Features added since V13 GA

- Enhanced with a COBOL Explorer. A post-mortem debugger
- Support for CICS 5.2

**FA Plugin**
- Implemented FAAPI.OpenReport function which enables users to open a Fault Analyzer report from their own application. You can now access Fault Analyzer report using a provided API which you can exploit with your own tools.

**Common Component**
Support for Security product stored certificates for SSL enablement. (RFE 41255) from APAR PI14699. The common server has been updated in order to accept a new configuration keyword SSL_KEYRING which provides the name of the keyring to be used. As the keyring is shared with the connecting users, the name should be specified in userid/ringname format. Userids (or groups) that are allowed to access the server need to be permitted UPDATE access to the profile IRR.DIGTCERT.LISTRING and also CONTROL access to IRR.DIGTCERT.GENCERT
Debug Tool: Features added since V13 GA

- Support for CICS 5.2
- CICS/TS 5.2 - Support "application versioning"
  - Supported using DTCN remote plugin only.
- Add capability to trap DB/2 SPs based on client ID
  - Using DTSP remote plug-in.
- Provide console interface to list Terminal Interface Monitor (TIM) users.
- Enhanced DT Code Coverage support for z/OS XL C.
- Enhanced DT Code Coverage support for RD/z and PD Tools Studio.
- SET LIST BY SUBSCRIPT ON support for COBOL for MFI. This includes QUERY LIST BY SUBSCRIPT.

```
02 ARRAY
  03 ARRAY-FIRST
    01 SUB(1) 'AB'
    01 SUB(2) 'FG'
    01 SUB(3) 'KL'
    01 SUB(4) 'PO'
    01 SUB(5) 'UV'
    01 SUB(6) 'Z1'
  03 ARRAY-LAST
    01 SUB(1) 'CD'
    01 SUB(2) 'MH'
    01 SUB(3) 'NM'
    01 SUB(4) 'PR'
    01 SUB(5) 'UW'
    01 SUB(6) '24'
  03 ARRAY-MIDDLE
    01 SUB(1) 'DE'
    01 SUB(2) 'JJ'
    01 SUB(3) 'NO'
    01 SUB(4) 'ST'
    01 SUB(5) 'XY'
    01 SUB(6) '34'
```
IBM Fault Analyzer
COBOL Explorer
(an interactive post-mortem debugger)
COBOL Explorer: Post-mortem debugger

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

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

Start
Main program

Statement you need to fix!!!

MOVE 0 AVALUE.
MOVE 100 ATOTAL.
CALL CALC USING AVALUE ATOTAL.

Subprogram CALC

LINKAGE SECTION.
01 X PIC 9(4).
01 Y PIC 9(4).

DIVIDE Y BY X GIVING Z.

Point of Failure
What is COBOL Explorer?

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

- Query your program to see all lines where a variable is used, all places a procedure is called, or the value of variables on any executable source line.

- IBM Fault Analyzer V13, PTF UI18641 introduced this new feature, documented in the latest User’s Guide and Reference SC19-4116-04

Replaces the traditional source (editor style) viewer with a collapse/expand interface.
What exactly is a view?

- You create views of your program based on an executable source line and one or more variables from that source line. In FA we'd usually start with the abending source line, for example:

  \[
  \text{COMPUTE BASE-AMOUNT} = \frac{\text{PRICE}}{\text{CC}}
  \]

  if the program abended because of a divide by zero, we'd only really be interested in variable CC and we'd create a view with just one variable.

- The view shows a procedure traceback from the source line back to the origin. Any other source lines that reference selected variables are inserted into the view. Selected variable declarations also appear in the view. The view is a summary of your source program, showing only the parts you need to know about.

- Views use collapse/expand rather than the traditional source editor viewer. You can expand the view incrementally to see all variables in a DATA DIVISION section, all members of a structure, all procedures in the PROCEDURE DIVISION, all lines in a procedure.
How does the Collapse/Expand viewer work?

- The COBOL source program is made up of sections. In the DATA DIVISION sections are FILE SECTION, WORKING-STORAGE SECTION, LOCAL-STORAGE SECTION and LINKAGE SECTION. In the PROCEDURE DIVISION sections are procedures (SECTIONs and Paragraphs).

- The view uses collapse/expand rather than the traditional source editor to view the source program.

- The view shows a procedure traceback from the source line to the beginning of the program, any source lines that use selected variables are inserted into the view.
What else does it do?

- Shows all variables and their values in a given Data Division section (i.e. FILE SECTION, WORKING-STORAGE SECTION, LOCAL-STORAGE SECTION, LINKAGE SECTION).

- Invoke FM (File Manager) with or without a template from any FD in the FILE SECTION (provided the dataset is still available).
Starting point: Fault Entry List

Select an entry by typing I next to the entry

<table>
<thead>
<tr>
<th>Fault ID</th>
<th>Job/Tran</th>
<th>User ID</th>
<th>Sys/Job</th>
<th>Abend</th>
<th>Date</th>
<th>Time</th>
<th>Dups</th>
<th>Lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>F14832</td>
<td>VHECKE</td>
<td>VHECKE</td>
<td>ZT01</td>
<td>SOC8</td>
<td>2014/07/08 12:42:44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F14827</td>
<td>CINT</td>
<td>STC</td>
<td>ADTMN10</td>
<td>IU24</td>
<td>2014/07/04 12:53:06</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select 2 to specify the location of your listing or side file

Compiler Listing Not Found

Module ADTCFG.ADLAB.LOAD(COBEYX1) containing COBOL program COBEYX1 entry point COBEYX1 compiled date 2014/07/08 time 12:42:44 does not have a matching listing/side-file/DWARF load module.
This is the point-of-failure program.
Select one of the following options and press Enter:
1. F9 Continue without compiler listing or side file for COBEYX1
2. Specify compiler listing or side file to use for COBEYX1
3. Entry search for compiler listing or side file for COBEYX1
4. Do not prompt again for any missing listing or side file
5. Only prompt for the point-of-failure program listing or side file
6. Add COBEYX1 to your side file search exclude list

The trace of the listing/side-file search follows.

Rejected - ADTCFG.ADLAB.SYSDEBUG SYSDEBUG side file
Failed - IDISYSDMB Member COBEYX1 not found

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
Provide listing or side file location. IDILANGX files are also supported

Specify the data set and member name containing the compiler listing or side file and press Enter.

Specify Compiler Listing or Side File

Command --->

Data Set Name . . . : 'ADTCFG.ADLAB.LISTING'

Member . . . . . . . : COBEXI

Alternatively, place cursor on choice and press Enter to use previously specified load module data set name.

From the Interactive Reanalysis Report panel select 2 for Event Summary
Select the event that you are interested, in this case the Abend by typing 1

The cause of the Abend is a decimal-divide exception at line 175. Notice the COBOL Explorer selection (red circle). Press enter or double click to start the Explorer.
The COBOL Explorer presents a collapsed view of the event

Since we know the issue is with the value in variable CC, deselect the other two variables and press enter to create a new Explorer View with only the elements causing the Abend.
In above view notice the following:

- It shows a collapsed view showing only the execution path to the Abend.
- Next to each relevant line there is a plus sign (see red circle). You can double-click or place your cursor and press enter to expand each section.
- It also shows the place where CC is declared. Line 26 in the view.
- You can expand or collapse each section as needed.
- It also highlights the Abending line (175).
- By following the lines marked with the + sign you can determine that the execution path is: **CALC-TAX < PROCESS-CAR < PROCESS-VEHICLE < READ-FILE < PROCEDURE DIVISION**
Next lets take a look at variable CC more closely. So we expand line 26 by double clicking on + sign or placing cursor on plus sign and press enter.

In the circled areas above you can select to remove all comments from the declaration section or remove the Level 88 variables, or the Redefines. This is done by double clicking on the minus sign next each option or placing the cursor on top of the minus sign and pressing enter.
z/OS Problem Determination Tools

If you are interested in seeing all the values for the variables in a section then from the same view you just need to double click on the + sign next to the section desired.

After selecting the Working-Storage Section.

<table>
<thead>
<tr>
<th>Associated Storage Areas</th>
<th>Line 1 Col 1 80</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME: VCHECKER</td>
<td>SYSTEM ABEND: 0CB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WORKING-STORAGE SECTION</th>
<th>Data Value</th>
<th>Source (Starting at)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off Hex Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>Level 88 Items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 05F5F0F0F0F00</td>
<td>50000</td>
<td>03 PRICE</td>
</tr>
<tr>
<td>3C F0F0F0F00</td>
<td>0.00</td>
<td>03 CARBON-TAX</td>
</tr>
<tr>
<td>41 130C5E2D3A 40404040</td>
<td>0</td>
<td>03 SPECIFICATI</td>
</tr>
<tr>
<td>51 40404040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55 D406C45C 4040E240</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65 40404040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69 13F2F0F1F0</td>
<td>2010</td>
<td>05 MAKE</td>
</tr>
<tr>
<td>6D 4F4</td>
<td>4</td>
<td>05 ENGINE</td>
</tr>
<tr>
<td>Level 88 Items</td>
<td></td>
<td>07 FUEL</td>
</tr>
<tr>
<td>6E 00F0</td>
<td>0</td>
<td>07 CYLIN</td>
</tr>
<tr>
<td>70 00F0F0F00F00</td>
<td>0</td>
<td>07 CC</td>
</tr>
<tr>
<td>74 00F0F0F0F00</td>
<td>0</td>
<td>07 KW</td>
</tr>
<tr>
<td>78 00F0F0F0F0000</td>
<td>3.9</td>
<td>07 ZERO-</td>
</tr>
<tr>
<td>78 00F0F2</td>
<td>2</td>
<td>03 FEATURES</td>
</tr>
<tr>
<td>7D 00F0</td>
<td>1</td>
<td>05 SEATING</td>
</tr>
<tr>
<td>Level 88 Items</td>
<td></td>
<td>05 TRANSMIS</td>
</tr>
<tr>
<td>7E 00F8</td>
<td>8</td>
<td>05 COLOUR</td>
</tr>
</tbody>
</table>

PF7 and PF8 can be use to scroll down and up in the view.
z/OS Problem Determination Tools

Let's go back to the source view. You now know that the + and - signs next to each entry in the source are used to expand/collapse sections of the view.

Next we are going to expand the Procedure Division by Double Clicking on the + sign on line 101.
The Procedure Division is now expanded but note the following:
• All procedures are still collapsed
• You can expand incrementally by clicking the + sign next to the section
• If you want to expand all procedures you can double click on the Expand All selection at the top of the view
• Or reverse your action by selecting Collapse all at the top of the view
• Collapsing/Expanding Comments, Level 88 Variables, and Redefines is also an option.

We understand that real state is precious!!
After all the navigation hints, let's go back to finding more about what caused the abend. Remember that the issue was a divide by zero because CC = 0.

Notice in the view the following:
• Line 126 is colored different, and the reason is because WS-REC is a redefine for VEHICLE-RECORD and CC is part of it.
• The collapsed view is showing all the relevant declarations
  • Line 26 the declare for VEHICLE-RECORD
  • Line 81 the redefine for HEADINGS
  • Line 84 the redefine for WS-RECORD
We need to take a close look at Line 126. Double click on the line.
IBM z/OS Problem Determination Tools

All important data associated with line 26 shows in the new view.

After selecting FS-REC we get a new view that includes where it is declared as well as the statements where FS-REC is used. Notice that it is part of FILE SECTION.
Integration with IBM File Manager

Previous slide showed the FILE SECTION where FS-REC is declared. If you expand that section and then double click on the FD for FS-REC the IBM File Manager is invoked to provide additional information about the record.

This is a VSAM record and because I choose Edit I can modify the value of CC directly. Since I don’t have a FM template defined the display is not a neat as expected for File Manager.
Cross reference data and other navigation facilities available with the COBOL Explorer.
You can double click on any data item. I am choosing MAKE on line 39
IBM z/OS Problem Determination Tools

The Data Cross-Reference view

- In the view you can see all statements that reference MAKE.
- If the variable is a GROUP then its MAP is shown. See red circle
- Double clicking on any entry in the popup view will navigate to that line and expand the source around the line.
z/OS Problem Determination Tools

After expanding the source by double clicking on line 160, the Explorer creates a new view starting with that line.

Using PF7 (Scroll Up) you can see that line 160 is part of STRING-NAME proc.
Procedure Cross-Reference: In the previous slide we found about Procedure STRING-NAME. By double clicking on STRING-NAME the explorer creates the Cross-Reference View.

The Cross-Reference shows us that STRING-NAME is called from PROCESS-VEHICLE.

```
Procedure Cross-Reference

Procedure: 090157 STRING-NAME.
References:
  090125 PROCESS-VEHICLE.
  090130 PERFORM STRING-NAME

*** Bottom of data.
```

```
F1=Help  F3=Exit  F5=RptFind  F7=Up  F8=Down  F12=Cancel

000062  05 COLOUR     PIC 9.
000072  05 BODY       PIC 9.
+ 000081  01 HEADINGS REDEFINES VEHICLE-RECORD.
+ 000084  01 WS-REC REDEFINES VEHICLE-RECORD.
+ 000098 LINKAGE SECTION.
+ 000101 PROCEDURE DIVISION USING PARMs.
  000105 PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
+ 000119 READ-FILE.
  000122 NOT AT END PERFORM PROCESS-VEHICLE
+ 000125 PROCESS-VEHICLE.
  000126 MOVE FS-REC TO WS-REC.
  000133 PERFORM PROCESS-CAR
+ 000140 PROCESS-CAR.
  000151 PERFORM CALC-TAX.
  000152 STRING-NAME.
```
Procedure Cross-Reference. After double clicking on PROCESS-VEHICLE the view is updated to show the start of the procedure.

You can double click on any procedure or GOTO. In this example the view is repositioned to show PROCESS-COMMERCIAL.

Complete your session evaluations online at www.SHARE.org/Pittsburgh-Eval
IBM Fault Analyzer

COBOL Explorer

(an interactive post-mortem debugger)

Watch the video @ http://www.youtube.com/watch?v=ZXwsaBnfk2Q
Debug Tool IMS Private Message Region Enhancement

Insert Custom Session QR if Desired.
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 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.

![Image of terminal window with JCL data set list]

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

![JCL panel](image-url)
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”.

![Image of a computer screen showing a transaction entry dialog]

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