What's New In the IBM Problem Determination Tools

Doug Stout
IBM Problem Determination Tools
stoutd@us.ibm.com

Session 17275: What's New In the IBM Problem Determination Tools
August 10, 2015
Notices and Disclaimers

Copyright © 2015 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

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

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

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

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM’s products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

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

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

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

- What are the IBM Problem Determination Tools
- Ongoing Enhancements
- 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.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
IBM Problem Determination Tools for z/OS (PD Tools)

Application Performance Analyzer for z/OS

Fault Analyzer for z/OS
- Abend analysis at source-level to speed up problem analysis.

Debug Tool for z/OS
- Interactive program debugging to monitor and control the execution of application programs on z/OS, COBOL modernization, code coverage

File Manager for z/OS
- Advanced editor, viewer, and z/OS data handling utilities to work with data sets, DB2, IMS, CICS, MQ, and HFS files

IBM Problem Determination Tools Modernization Solution Pack

Workload Simulator for z/OS & OS/390
HourGlass
Data Set Commander (formerly IPT)
Migration Utility
Choice of User Interfaces (3270, Eclipse, CICS, Web)

- PD Tools can be used in a variety of environments and interfaces, because we recognise that you work with heterogeneous environments with a diverse workforce.
  - ISPF interface is still the most commonly used interface for most products.
  - Our IPV technology allows us to display ISPF-like panels under CICS. This enables usage of FA and FM functionality directly from CICS environment without switching to ISPF environment.
  - Eclipse interface is available for all products. Integrates with seamlessly with IBM’s key Eclipse-based products (RDz, CICS Explooler and others).
  - Browser interface is available for Fault Analyzer.

1. ISPF interface

2. CICS interface

3. Eclipse interface

4. Browser interface

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Implementation of GUls made Simple!

Simplicity
- PD Tools GUls (FM, FA, APA, DT) talk to a single PD Tools Common Server address space
- PD Tools Studio: All PD Tools GUls package together including z/OS Explorer
- You can integrate into RDz or CICS Explorer or IMS Explorer or your own Eclipse for Development.
- Select capabilities you need from a single update site.
Choice of interfaces supports varying developer skills

- Reusing mainframe application assets requires expertise in both traditional and new zOS technologies
- To maximize productivity:
  - Some developers may prefer traditional interfaces
  - Others prefer GUls
- Use common Eclipse-based tooling across multiple platforms and languages
  - GUls can reduce training costs, and provide a more familiar environment for developers with experience on environments other than z/OS
Graphical user interfaces for PD Tools

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

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
PD Tools GUIs are Eclipse plug-ins

- The graphical interfaces are plug-ins for Eclipse platforms, such as:
  - The PD Tools Studio
  - Rational Developer for System z (RDz)
  - CICS Explorer
  - IMS Explorer
- If you do not already have RDz or other Eclipse software installed on your workstation:
  - Install the PD Tools Studio
    - It is a free download from IBM
    - It is a complete Eclipse environment, with the PD Tools plug-ins already installed
- If you are already using an Eclipse platform, you can:
  1. Install the PD Tools GUI plug-ins into your favorite Eclipse
    - The plug-ins are available as new perspectives and views after they are added
  2. Or, install the PD Tool Studio separately
    - Your original Eclipse platform is unchanged, and you can use the PD Tools Studio as a completely separate Eclipse environment

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
PD Tools Studio free download

- www.ibm.com/software/awdtools/deployment/pdtplugins/
- Download and follow the installation instructions on the download page

IBM Problem Determination Tools Studio and Plug-ins Downloads

Downloadable files

Abstract
Problem Determination Tools Studio and Plug-ins download packages.

Download Description

The IBM Problem Determination Tools Studio and Plug-ins provide easy access through a graphical user interface (GUI) on the workstation to the power of the host IBM Problem Determination Tools — Application Performance Analyzer for z/OS, Debug Tool for z/OS, Fault Analyzer for z/OS, File Manager for z/OS, and Workload Simulator for z/OS and OS/390.

Installation Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Manager installation</td>
<td>Create your own custom workbench by selecting products from a list of IBM compatible products including the Problem Determination Tools plug-ins.</td>
<td>Link to install instructions</td>
</tr>
</tbody>
</table>

IBM Problem Determination Tools Studio is a standalone, Eclipse-based environment that contains the capability found in each of the individual product plug-ins for Application Performance Analyzer, Debug Tool, Fault Analyzer, File Manager, and Workload Simulator, including:

- With Application Performance Analyzer, it can be used for submitting new observation requests and navigating performance analysis reports.
- With Debug Tool, it helps debug load modules running on z/OS, including those run in CICS.
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.

Performance Analysis

Abend analysis involving mixed legacy and Java workload

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Day 1 support for upgrades of z Systems hardware, z/OS, languages and middleware

- **IBM z13 hardware and z/OS V2.1**
  - Support for all new compilers that exploits z13 hardware.

- **Enterprise COBOL V5.2 and V5.1 compilers**
  - Support for the latest compiler advancements.
  - Direct support for the new DWARF format side file.

- **CICS Transaction Server V5.2**
  - PD Tools support CICS TS V5.2 environment.
  - Exploitations of the key CICS TS V5.2 enhancements (eg. Support for “applications”).
  - Support for CICS TS V5.3 Open Beta program.

- **Enhancements in support of major languages and subsystems**
  - Enterprise PL/I V4.5 and z/OS XL C/C++ V2.1 compilers
  - DB2 11 for z/OS
  - IMS 13 Transaction and Database Servers
  - WebSphere Application Server for z/OS V8.5
  - WebSphere MQ for z/OS V8.0

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Upgrade example… SIMD (Single Instruction Multiple Data)
A new instruction set with z13

Operate on multiple data-elements (vectors) simultaneously
Can offer dramatic speed-up to data-parallel operations. For COBOL and PL/I we can speed up some string operations

Vector registers are 128-bits wide and can be used to operate concurrently on:
- Sixteen 8-bit characters
- Four 32-bit integer values
- Two 64-bit floating point values
- Four 32-bit floating point values
- Etc
COBOL V5.2 Exploitation of SIMD

- INSPECT TALLYING counts characters in a string
- INSPECT REPLACING replaces characters in a string

Sample code:

```
WORKING-STORAGE SECTION.
01 VARS.
   02 STR PIC X(255).
   02 C PIC 9(5) COMP-5 VALUE 0.
PROCEDURE DIVISION.
   MOVE ALL 'abc def ghi jkl ' TO STR
   PERFORM 100000000 TIMES
       INSPECT STR TALLYING C FOR ALL ' '
   END-PERFORM
GOBACK
```

30 times faster
97% less CPU
INSPECT TALLYING/REPLACING

V5.1 -- ARCH(10)

```
LHI R0,0xff
XR R1,R1
LA R12,152(,R8)  # STR
```

L1: EQU *

```
CLI 0,(R12),X'40'
JNOP L2
LA R1,1,(R1)
```

L2: EQU *

```
LA R12,1,(R12)
BRCT R0,L1
A R1,407,(R8)  # C
ST R1,407,(R8)  # C
```

V5.2 -- ARCH(11)

```
LHI R0,0xfe
XR R1,R1
LA R12,152(,R8)  # STR
```

```
VREPIB VRF27,0x40
VGBM VRF25,0x0
```

L1: EQU *

```
VLL VRF24,R0,0,(R12)
AHI R12,0x10
VCEQB VRF24,VRF24,VRF27
AHI R0,0xffff0
VLCB VRF24,VRF24
VAB VRF25,VRF25,VRF24
JNL L1
VGBM VRF26,0x0
VSUMB VRF25,VRF25,VRF26
VSUMQF VRF25,VRF25,VRF26
VLGVG R1,VRF25,1,(R1)
A R1,407,(R8)  # C
ST R1,407,(R8)  # C
```

Timing (100 million times in a loop)

V5.1 : 46.63 cpu seconds
V5.2 : 1.54 cpu seconds

30 times faster
97% less CPU
Decimal Floating Point

• Packed Decimal (COMP-3) data items convert directly from/to DFP (decimal floating point)

Sample code:

```ada
WORKING-STORAGE SECTION.
  01 VARS.
    02 A PIC S9(25) Packed-Decimal VALUE +1234567890123456789012345.
    02 B PIC S9(25) Packed-Decimal VALUE +2468097531246809753124680.
    02 C PIC S9(25) Packed-Decimal VALUE 0.

PROCEDURE DIVISION.
  PERFORM 100000000 TIMES
     DIVIDE A BY B GIVING C
  END-PERFORM

Timing (100 million times in a loop)
  V5.1 : 2.53 cpu seconds
  V5.2 : 1.65 cpu seconds

36% less CPU

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
```
Using new instructions for Decimal Floating Point

V5.1 -- ARCH(10)

```
XGR R0, R0
ICMH R0, X'1', 152 (,R8) # A
L R0, 153 (,R8) # A
LG R1, 157 (,R8) # A
CXSTR FP0, R0
XGR R0, R0
ICMH R0, X'1', 165 (,R8) # B
L R0, 166 (,R8) # B
LG R1, 170 (,R8) # B
CXSTR FP1, R0
DXTR FP4:FP6, FP0:FP2, FP1:FP3
FIXTR FP0:FP2, 9, FP4:FP6
CSXTR R0:R1, 0, FP0:FP2
STCMH R0, X'1', 178 (,R8) # C
ST R0, 179 (,R8) # C
STG R1, 183 (,R8) # C
ZAP 178 (13, R8), 178 (13, R8) # C
```

V5.2 -- ARCH(11)

```
CXPT FP0:FP2, 152 (13, R8), 0x8
CXPT FP1:FP3, 165 (13, R8), 0x8
DXTR FP4:FP6, FP0:FP2, FP1:FP3
FIXTR FP0:FP2, 9, FP4:FP6
CPXT FP0:FP2, 178 (13, R8), 0x9
```

Timing (100 million times in a loop)
V5.1 : 2.53 cpu seconds
V5.2 : 1.65 cpu seconds

36% less CPU

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Enterprise COBOL V5.1 and 5.2 Support

• Enterprise COBOL V5.2
  • Includes all enhancements delivered in COBOL V5.1 PTFs
  • Provides Day 1 support for z13 processor
    – Expanded use of Decimal Floating Point for PACKED-DECIMAL data
    – Uses new SIMD instructions for INSPECT TALLYING or REPLACING statements
• Improves Application Performance
  – Well-structured, compute intensive applications running on z13 have shown CPU reduction up to 14%
  (compared with COBOL v5.1 on ZEC12)

• PD Tools Exploitation
  • The new DWARF format side file of COBOL V5.1 brings a significant advantage
    – Program information required for problem diagnosis is stored in the same member as the load module.
    – Simpler management of side files - it is no longer necessary to maintain load modules and matching side files in separate libraries
    – It eliminates the possibility of side file mismatch
    – No performance penalty for selecting the DWARF option. DWARF information is not loaded into storage while program executes normally. It is only read from when it is needed by PD Tools
• This new DWARF format is used by:
  – Debug Tool
  – Fault Analyzer
  – Application Performance Analyzer
• All PD Tools support new z13 hardware instructions exploited by the compiler.
• File Manager for z/OS continues to support COBOL V5.1 level COPYBOOKs.

1 Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

developerWorks Article: Prepare to upgrade to Enterprise COBOL for z/OS V5
Enterprise COBOL for z/OS V5.1.1 Migration Guide
Automatic Binary Optimizer for z/OS Open Beta Support

• Standalone Optimization Technology based on Enterprise COBOL V5
  – Automatically performs optimization for existing programs without recompiling the source program

• Value To Clients
  – Increase performance of business critical applications
  – IBM internal benchmarks show performance averaging > 20% over already optimized COBOL benchmarks
  – Requires no source recompiling or intensive performance options tuning
  – Initial support planned for COBOL but technology can be extended to benefit PL/I and C/C++ applications

• Supports programs compiled by COBOL compiler V3 and V4. Investigating support for OS/VS COBOL and VS COBOL II.

PD Tools Support

• FA can analyze abends in Optimus optimised programs and offer source level reporting
• APA can report performance with source mapping
• DT: planned support for debugging of Optimus optimized programs, with some limitations

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
CICS TS V5.2 Exploitation and CICS TS V5.3 Open Beta

CICS Multi-Versioning

- Increased operational efficiency – reducing # of CICS regions

Debug Tool supports CICS multi-versioning through DTCN remote plug-in.

Fault Analyzer reports the application information.

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

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
IBM Debug Tool Version 13

What’s new?

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

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
**Problem trying to address:**

- In IMS online environments, debugging a transaction in Debug Tool can monopolize the scheduling environment for a given message class.
- 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:**

- In Debug Tool utilities, 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.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
**Delayed Debug capability**

- By default, Debug Tool starts a debug session at the main program
  - (The first entry compile unit of the initial load module of an application)
- Without delay Debug, you can set an ‘AT ENTRY’ breakpoint to stop at a subroutine after the debugger starts
- With the Delay Debug feature, you can bypass the startup of Debug Tool at the main program, and begin the debugging session at a named subprogram
  - This can simplify debugging
  - It can also improve performance for complex long-running applications
- Debug Tool is in a dormant state in delay debug mode, and monitors only a few events


Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Other debug Tool enhancements

• Code coverage eclipse and web interface
  ▪ Browser interface for reviewing code coverage report.
  ▪ Quickly understand coverage of regression testing effort.
  ▪ Drill down to source line level for detailed analysis.
  ▪ Code coverage services can be accessed via APIs.

• Load Module Analyzer Eclipse plug-in
  ▪ Analyzes MVS load modules or program objects to determine the language translator (compiler or assembler) used to generate the object for each CSECT.
  ▪ Invoke Load Module Analyzer from Eclipse environment.

• JCL Wizard
  ▪ An ISPF edit macro that can be used to modify a JCL or procedure member, creating the statements to invoke Debug Tool in various environments.
  ▪ It has the capability to create statements that can be used to invoke Debug Tool for z/OS for the Terminal Interface Manager, Mainframe Interface, or Remote GUI.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Other debug Tool enhancements

• **Debug Tool usage log**
  - Log when each Debug Tool session starts and ends

• **TRACE LOAD and LIST TRACE LOAD commands**
  - Displays an ongoing trace of all load modules loaded by the application

• **Consolidation of Debug Tool LE user exits**
  - User exits EQADBCXT, EQADCCXT and EQADICXT have been consolidated into EQAD3CXT
  - Use the same exit now for all environments: batch, CICS, IMS

• **SET LIST BY SUBSCRIPT ON support for COBOL for 3270**

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
IBM File Manager Version 13

- **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 graphical 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/Orlando-Eval
File Manager enhancements

• IMS enhancements:
  • Support for viewing / editing IMS Databases using the eclipse plug-in
  • IMS Template Editor – Create / Modify in the eclipse plug-in
  • Extract and Print have new keywords KEYRANGE and MAXRSEGS

• Multiple Layout Dynamic Dynamic Templates
  • You can now create a template, without using copybooks, that supports files with multiple types of records

• Template editor, and support for XML templates
  • You can create and update templates using a batch utility
  • You can export/import a template to an XML file. Also the XML file can be used directly as a template.

• DB2 support enhancements:
  • XML and LOB (large object) support

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
File Manager: Features added since V13 GA

• For FM/IMS, session timeout support - via PI09504. Aimed at resolving resource contention problems.

• 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. 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 data segments when editing data using a segmented template – via PI04887.
File Manager: Features added since V13 GA

• 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.
  – Added 'Structure Mode' to Single Record view.
Little known / under-used features of File Manager

• Support for MQ queues, CICS TS/TD queues, HFS file
  • Interactively edit/view queues with the File Manager editor and viewer
  • Use a queue or HFS file as either input or output for the copy utility. For example:
    – Copy from a file to an MQ queue
    – Copy from an MQ queue to a TS queue
• Similar access from the Print and Compare utilities

Data scrambling
  • While copying data, replace data with a different characters (or bytes) of data.
  • Protecting sensitive information in data, typically used while preparing test data from production data sources.

File Manager provides four types of scrambling:
  • “Random”: data scrambled to produce a random output value on each invocation.
  • “Repeatable”: data scrambled to produce the same output value on each invocation.
  • “Translate”: data scrambled using input and output values held in the value data set.
  • “Exit”: data scrambled using a scrambling exit (write your own scrambling algorithm).
IBM Application Performance Analyzer Version 13

- **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/Orlando-Eval
Application Performance Analyzer: Features added since V13 GA

- Enhancement to CAZBATCH and CAZAPI01 allows override of sample file name
- New S10 Error Messages report.
- Enhancement to the C09 report, to display up to three possible interpretations of the instruction before the PSW
- Enhancement to enforce user level security in APA when creating, updating, reading and deleting sample file data sets, and data sets containing exported sample files.
- Enhancement to the configuration setting AltCICSStepNm to allow alternate CICS execution module names.
- Support for CICS TS 5.2
IBM Fault Analyzer Version 13

- **IBM Fault Analyzer** improves developer productivity and decreases deployment costs by helping to analyze and correct application failures quickly (CICS/DB2/IMS/MQ/COBOL/PLI/ASM/C/C++/ASM/JAVA).

- Develop and test new and existing applications more productively, helping to reduce costs along the way.

- Proven 3270-based interface and free graphical user interface.

What’s new?

- Optimize
  - Sidefile checking and reporting
  - Customer specific module and abend descriptions
  - ISPF Virtual Relief

- Modernize
  - Enhanced web interface using Dojo
  - Enhanced Java support
  - Support for DB2 V11, IMS V13, CICS TS 5.1, z/OS 2.1, COBOL V5, PL/I V4.4, Java V7

- A range of key customer requirements

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

- Enhanced with COBOL Explorer. A post-mortem debugger
- Support for CICS 5.2
- Implemented FAAPI. An OpenReport API which enables users to open a Fault Analyzer report from their own application.
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
  - Navigate your application’s execution path from a point of failure interactively
  - Examine a variable and how the value changed which led to failure

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

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

  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).
z/OS Problem Determination Tools

Starting point: Fault Entry List

Select an entry by typing I next to it

IBM Fault Analyzer - Fault Entry List
Command ===> Scroll ===> CSR

Fault History File or View : 'ADTCFG.IDI.HIST'

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

Compiler Listing Not Found
Command ===> Scroll ===> CSR

Module ADTCFG.ADLAB.LOAD(COBEX1) containing COBOL program COBEX1 entry point COBEX1 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. (F3) Continue without compiler listing or side file for COBEX1
    2. Specify compiler listing or side file to use for COBEX1
    3. Retry search for compiler listing or side file for COBEX1
    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 COBEX1 to your side file search exclude list

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

Rejected - ADTCFG.ADLAB.SYSDEBUG SYSDEBUG side file
Failed - IDSYSDB
Member COBEX1 not found
Provide listing or side file location. IDILANGX files are also supported

Specify Compiler Listing or Side File

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

Data Set Name: ADTCFG.ADLAB.LISTING
Member: COBEX1

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

Interactive Reanalysis Report

Fault Summary:
Module COBEX1, program COBEX1, source line # 175 : Abend S0CB (Decimal-Divide Exception).

Select one of the following options to access further fault information:
1. Synopsis
2. Event Summary
3. Open Files
4. Storage Areas
5. Messages
6. Language Environment Heap Analysis
7. Abend Job Information
8. Fault Analyzer Options

Fault Analyzer maximum storage allocated: 212 megabytes.
Select the event of interest, 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). Cursor-select “COBOL Explorer”.

Abend Code: S60B
Program-Interruption Code: 000B (Decimal-Divide Exception)

The divisor was zero in a signed decimal division.

The source code below was executed via the following sequence of PERFORM statements:

```cobol
PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
       NOT AT END PERFORM PROCESS-VEHICLE
       PERFORM PROCESS-CAR
       PERFORM CALC-TAX.
```

COBOL Source Code:

```cobol
000175
COMPUTE BASE-AMOUNT = PRICE / CC
```

Data Field Declarations:

```cobol
000033 03 PRICE  PIC 9(6).
000049 07 CC    PIC 9(4).
000094 01 BASE-AMOUNT PIC 9(3)V99 COMP-3.
```

Data Field Values:

- BASE-AMOUNT = 0.00
- CC = 0
- PRICE = 50000
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.
z/OS Problem Determination Tools

The Explorer View

Notice:
• A collapsed view is displayed, showing only the execution path to the abend
• It also shows the place where CC is declared. Line 26 in the view.
Next lets take a look at variable CC more closely. We expand line 26 by cursor-selecting the + sign.

In the circled areas above you can comments, 88 levels, or redefines.
If you want to see all the variables in a section then from the same view then cursor-select the + sign next to the section wanted.

After selecting the Working-Storage Section.
Lets 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.
IBM z/OS Problem Determination Tools

The Procedure Division is now expanded:
• All procedures are still collapsed

Exploring COBOL Program COBEX1
Command ===>
JOBNAME: VHEECKEA SYSTEM ABEND: OCB ZT01 2014/07/08 12:42:44
+ Expand all / - Collapse all + Comments + Level 88 - Redefines

+ 000001 IDENTIFICATION DIVISION.
+ 000003 ENVIRONMENT DIVISION.
+ 000011 FILE SECTION.
+ 000017 WORKING-STORAGE SECTION.
+ 000026 01 VEHICLE-RECORD.
  000038  03 SPECIFICATION.
  000042       05 ENGINE.
  000049       07 CC PIC 9(4).
+ 000081 01 HEADINGS REDEFINES VEHICLE-RECORD.
+ 000084 01 WS-REC REDEFINES VEHICLE-RECORD.
+ 000098 LINKAGE SECTION.
  000101 PROCEDURE DIVISION USING PARMs.
  000102     DISPLAY '*** COBEX1 - STARTING...'.
  000103     PERFORM INIT.
  000104     OPEN I-0 VEHICLE-FILE.
  000105     PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
  000106     CLOSE VEHICLE-FILE.
  000107     PERFORM PRINT-TOTALS.
  000108     DISPLAY '*** COBEX1 - END OF PROGRAM'.
  000109     GOBACK.
+ 000111 INIT.
+ 000119 READ-FILE.
  000122     NOT AT END PERFORM PROCESS-VEHICLE
+ 000125 PROCESS-VEHICLE.
  000126     MOVE FS-REC TO WS-REC.
  000133     PERFORM PROCESS-CAR
+ 000149 PROCESS-CAR.
  000151     PERFORM CALC-TAX.
+ 000153 PROCESS-COMMERCIAL.
+ 000157 STRING-NAME.
+ 000165 UNDO-TAX.
+ 000170 REDO-TAX.
+ 000173 CALC-TAX.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
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 differently, 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. Cursor-select the line.

```
Exploring COBOL Program COBEX1
Command ===> Scroll ===>

JOBNAME: VCHECKEA SYSTEM ABEND: 0CB ZT01 2014/07/08 12:42:44
+ Expand all / - Collapse all + Comments + Level 88 - Redefines

+ 000601 IDENTIFICATION DIVISION.
+ 000609 ENVIRONMENT DIVISION.
+ 000611 FILE SECTION.
+ 000617 WORKING-STORAGE SECTION.
+ 000626 01 VEHICLE-RECORD.
  000638   03 SPECIFICATION.
  000642       05 ENGINE.
+ 000649       07 CC PIC 9(4).
+ 000681 01 HEADINGS REDEFINES VEHICLE-RECORD.
+ 000684 01 WS-REC REDEFINES VEHICLE-RECORD.
+ 000698 LINKAGE SECTION.
+ 00101 PROCEDURE DIVISION USING PARM.
+ 00105   PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
+ 00119 READ-FILE.
+ 00122 NOT AT END PERFORM PROCESS-VEHICLE
+ 00125 PROCESS-VEHICLE.
  00126   MOVE FS-REC TO WS-REC.
  00133   PERFORM PROCESS-CAR.
+ 00149 PROCCESS-CAR.
  00151   PERFORM CALC-TAX.
+ 00173 CALC-TAX.
  00175   COMPUTE BASE-AMOUNT = PRICE / CC
  00178   COMPUTE BASE-AMOUNT = CC / CYLINDERS
```
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.
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.

```
Data Cross-Reference
Command ==>  Scroll ==>  CSR

Data Item:
000039 05 MAKE PIC X(20).

References:
000125 PROCESS-VEHICLE.
000126 DISPLAY 'VEHICLE: ' MAKE MODEL
000157 STRING-NAME.
000160 MAKE DELIMITED BY SPACES
000161 ' ' DELIMITED BY SIZE
000162 MODEL DELIMITED BY SPACES
000163 INTO NAME-STRING.

Map:
1 VEHICLE-RECORD              DS 9CL80       Group
  2 SPECIFICATION              11x DS 9CL58       Group
  3 MAKE                       11x DS 26C       Display

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

000048 07 CYLINDERS PIC 9(2).
000049 07 CC          PIC 9(4).
000050 07 KW          PIC 9(4).
000051 07 ZERO-100    PIC 9(2)V9.
000052*****************************************************************************
000052* VEHICLE FEATURES
000054*****************************************************************************
000055 03 FEATURES.
000056 05 SEATING       PIC 9(2).
000057 05 TRANSMISSION  PIC 9.
000058 05 COLOUR       PIC 9.
000072 05 BODY         PIC 9.
+ 000081 01 HEADINGS REDEFINES VEHICLE-RECORD.
+ 000084 01 WS-REC REDEFINES VEHICLE-RECORD.
+ 000088 LINKAGE SECTION.
```
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

```plaintext
Procedure Cross-Reference

Procedure:
000157 STRING-NAME.

References:
000125 PROCESS-VEHICLE.
000130    PERFORM STRING-NAME

*** Bottom of data.
```

```plaintext
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 PARGS.
000105   PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
+ 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.
+ 000157   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.
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.

Comp on evaluations online at www.SHARE.org/Orlando-Eval
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’s Debug Tool Utilities (ISPF) enhanced 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.IMSTEMPDTEMPCREATE FACILITY.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
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.
Select JCL cards to import to template

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

• The administrator types a forward slash (/) to modify the parameters. This allows the administrator to assign a message class to the template.
“Classes” field will contain new message class.

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

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

• A developer knows that a certain transaction has a problem.
• The developer would like to debug to determine the location & cause of the failure, in a private message region.
• After debugging, the developer codes a fix and would like to test it without affecting other users.
• The developer would use a private message region template to accomplish these tasks.
Select DTU option 4, sub-option 3

- User selects sub-option 3 to test with a private message region template.
Select a pre-defined IMS region template

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

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

---

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

![Image showing the console interface for modifying the TEST parameter.]

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
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.
Mobile to Mainframe DevOps for Dummies Book

Author: Rosalind Radcliffe, IBM Distinguished Engineer

Available as ebook and hard copy

Meet the author while at SHARE
Book signing at the IBM Booth #312
Atlantic Hall C
Monday and Tuesday 6-7:30 PM
First 250 copies

8 Chapters – 72 pages

1. Understanding the Value of the Mainframe
2. Typical Mainframe Application Development Challenges
3. DevOps and Mainframe, "Mainlain? Possible?
4. DevOps Best Practices for Mainframe
5. Building for the modern omni channel world
6. DevOps Success Stories in the Enterprise
7. Making a DevOps transition
8. Where DevOps can take you

Why the unique power of the mainframe is perfect for developing hybrid cloud applications
How mobile front ends can rejuvenate those back-end systems to reach new customers
The special considerations of using DevOps to accelerate mainframe software delivery

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Additional Information

Introduction to IBM z/OS Tools – CICS Tools and Problem Determination Tools
IBM Rational YouTube Channel
IBM Education Assist – training materials for PD Tools, CICS Tools and much more.

IBM Problem Determination Tools for z/OS
IBM Rational Developer for System z

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