

This is the tutorial for IBM's Fault Analyzer for z/OS®, one of the IBM zSeries® problem determination tools.



▪ Viewing a real-time fault analysis report

▪ Interactive reanalysis

- Interactive reanalysis tour
- How to use the File Manager interface

▪ Program source mapping during reanalysis

- Applying side files and compiler listings
- Automating the search for side files and listings with options
- Requesting prompts for file names during reanalysis
- Re-creating a compiler listing for an existing module

In this section, you will see how to view a real-time report from the online interface.

View a real-time analysis report



You can view an analysis report from:

- The Fault Analyzer TSO interface

- OR -

- The Fault Analyzer CICS® interface

- OR -

- The Fault Analyzer Eclipse GUI plug-in

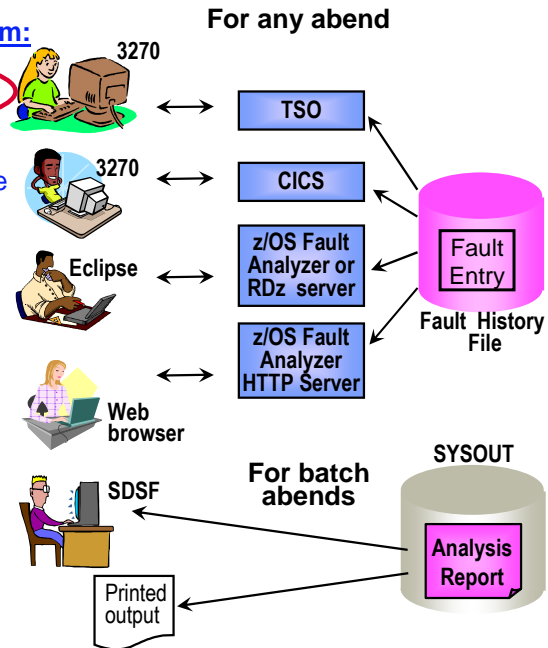
- OR -

- A web browser

- OR -

- Your batch job's SYSOUT

- Default is `sysout=*`
- DD name is IDIREPRT



3

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

In this example, the TSO online interface is used to view a real-time report. The person who installs Fault Analyzer on your system sets up a menu option or a command for you to get to the online interface. If you do not know how to get to the online interface on your system, contact your systems programmer or help desk.

Viewing a real-time analysis report



File Options View Services Help

IBM Fault Analyzer - Fault Entry List Line 1 Col 1 80
Command ==> Scroll ==> HALF

Fault History File or View : 'FAULTANL.V10R1.HIST'

{The following line commands are available: ? (Query), V or S (View saved report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H (Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault entry).}

	Fault_ID	Job/Tran	Job_ID	Dups	User_ID	Sys/Job	Abend	Date	Class
v	F00905	DNET845X	JOB15885		DNET845	DEMOMVS	S0C7	2010/02/23	A
—	F00904	ADPOT24L	JOB15840		ADPOT24	DEMOMVS	S522	2010/02/23	A
—	F00903	ADPOT24L	JOB15836		ADPOT24	DEMOMVS	S0C7	2010/02/23	A
—	F00902	DNET246F	JOB15775		DNET246	DEMOMVS	S0C7	2010/02/23	A
—	F00901	RDPT621	JOB15739		RDPT62	DEMOMVS	S0CB	2010/02/23	A
—	F00900	RDPT611	JOB15740		RDPT61	DEMOMVS	U4038	2010/02/23	A
—	F00899	RDPT621	JOB15738		RDPT62	DEMOMVS	U4038	2010/02/23	A
—	F00898	RDPT621	JOB15734		RDPT62	DEMOMVS	S0CB	2010/02/23	A
—	F00897	EPSL	STC11273		DNET215	CICSA0R5	4038	2010/02/23	A

Enter

4 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

The online interface is shown, and a fault history file has been opened. Notice that a list of available line commands is shown. Enter a V (for view) or S (for select) next to an entry to view its real time report. Press enter.

Collapse report sections



The screenshot displays the IBM Fault Analyzer interface. At the top, there is a menu bar with 'File', 'View', 'Services', and 'Help'. Below the menu bar, the text 'Saved Report' is visible on the left, and 'Line 1 Col 1 80' is on the right. A 'Command ==>' prompt is followed by a horizontal line. Below this line, the text '- Collapse all / + Expand all' is shown, with the minus sign circled in red. A callout box with a pointer to the minus sign contains the text: 'Tip: If you put the cursor here, and press Enter ...'. The main content area shows a report for 'IBM Fault Analyzer for z/OS V10R1M0 (HADQA10 2009/10/22)'. It includes a copyright notice: 'Copyright IBM Corp. 2000, 2009. All rights reserved.' and a system abend report: 'JOBNAME: DNET845X SYSTEM ABEND: 0C7 02/23 15:45:02'. The report text reads: '- <H1> I B M FAULT ANALYZER SYNOPSIS' followed by 'A system abend 0C7 occurred in module SAM2 program SAM2 at offset X'39A''. Below this, it states: 'A program-interruption code 0007 (Data Exception) is associated with this abend and indicates that:' and 'A decimal digit or sign was invalid.' A yellow 'Enter' button is located at the bottom right of the report area. At the bottom of the screenshot, there is a footer with the number '5' on the left, 'IBM Fault Analyzer for z/OS - V12 Tutorial' in the center, and '© 2012 IBM Corporation' on the right.

The real-time report is shown. The real-time report is exactly the same, whether you view it in SYSOUT or in the history file from the online interface.

But an advantage of using the online interface is that you can collapse and expand report sections. To collapse all of the sections, put your cursor on the minus sign next to the “Collapse All” field, and press enter.

Real-time analysis report with sections collapsed



```
File View Services Help
-----
Saved Report                               Line 1 Col 1 80
Command ==>                               Scroll ==> PAGE
- Collapse all / + Expand all
*****
* IBM Fault Analyzer for z/OS V10R1M0 (HADQA10 2009/10/22) *
*                                                                 *
* Copyright IBM Corp. 2000, 2009. All rights reserved.         *
*****
JOBNAME: DNET845X  SYSTEM ABEND: 0C7                               02/23 15:45:02

+ <H1> I B M   F A U L T   A N A L Y Z E R   S Y N O P S I S
+ <H1> I B M   F A U L T   A N A L Y Z E R   E V E N T   S U M M A R Y
+ <H1> I B M   F A U L T   A N A L Y Z E R   E V E N T   D E T A I L S
+ <H2> E V E N T 1 O F 3: CALL (DSA ADDRESS 20594030)
+ <H3> A s s o c i a t e d   O p e n   F i l e s
+ <H3> A s s o c i a t e d   S t o r a g e   A r e a s
+ <H5> B L F = 0 0 0 0   a t   a d d r e s s   0 0 0 2 3 F
+ <H5> B L F = 0 0 0 1   a t   a d d r e s s   0 0 0 1 9 0
+ <H5> B L F = 0 0 0 2   a t   a d d r e s s   0 0 0 0 8 F 5 9
+ <H5> B L W = 0 0 0 0   a t   a d d r e s s   0 0 0 0 9 0 6 0
```

... it will "collapse" the report sections

Put the cursor on a + next to a section heading, and press Enter to expand a section

Now all report sections are collapsed. To expand any individual section, put your cursor on the plus sign next to the section header and press enter.

Expand report sections



```
File View Services Help
Saved Report Line 1 Col 1 80
Command ==> Scroll ==> PAGE
- Collapse all + Expand all
*****
* IBM Fault Analyzer for z/OS (009/10/22) *
* Copyright IBM Corp. 2006 reserved. *
*****
JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:02

+ <H1> I B M F A U L T A N A L Y Z E R S Y N O P S I S
+ <H1> I B M F A U L T A N A L Y Z E R E V E N T S U M M A R Y
+ <H1> I B M F A U L T A N A L Y Z E R E V E N T D E T A I L S
+ <H2> E V E N T 1 O F 3: C A L L ( D S A A D D R E S S 2 0 5 9 4 0 3 0 )
+ <H3> A s s o c i a t e d O p e n F i l e s
+ <H3> A s s o c i a t e d S t o r a g e A r e a s
+ <H5> B L F = 0 0 0 0 a t a d d r e s s 0 0 0 2 3 F 8 8
+ <H5> B L F = 0 0 0 1 a t a d d r e s s 0 0 0 1 9 0 0 8
+ <H5> B L F = 0 0 0 2 a t a d d r e s s 0 0 0 0 8 F 5 9
+ <H5> B L W = 0 0 0 0 a t a d d r e s s 0 0 0 0 9 0 6 0
```

Put the cursor here, and press Enter ...



Or you can expand all of the sections again by putting your cursor on the plus sign next to the “Expand all” field and press enter.

Real-time analysis report with sections collapsed



The screenshot shows a terminal window with a menu bar (File, View, Services, Help) and a report header. The report content is partially collapsed. Annotations include a callout box pointing to the 'Expand all' option and another pointing to the abend description. A yellow box labeled 'F8' is also present.

```
File View Services Help
Saved Report Line 1 Col 1 80
Command ==> Scroll ==> PAGE
- Collapse all / + Expand all
*****
* IBM Fault Analyzer for z/OS V10R1M0 (HADQA10 2009/10/22) *
*
* Copyright IBM Corp. 2000, 2009. All rights reserved. *
*****
JOBNAME: DNET845X SYSTEM ABEND: 0C7 /02/23 15:45:02

- <H1> I B M FA U L T A N A L Y Z E R S Y N O P S I S

A system abend 0C7 occurred in module SAM2 program SAM2 at offset X'39A'.
A program-interruption code 0007 (Data Exception) is associated with this abend
and indicates that:

A decimal digit or sign was invalid.
Cause of the abend
F8
```

And all report sections are displayed again. Next you will see a tour of the contents of the report.

The synopsis is the first section. In many ways this is the most important section, because it gives you high level information about the abend. First the abend code, 0C7, is shown. You see the name of the program where the abend occurred, and the offset, which in some cases is helpful. When possible, fault analyzer gives a brief explanation of the abend code. In this case the description is: "A decimal digit or sign was invalid". Now you know that this abend was caused by bad data in a field.

It also says that the abend occurred in program SAM2, in a load module called SAM2. Scrolling forward...

After scrolling down in the synopsis



```
File View Services Help
Saved Report Line 21 Col 1 80
Command ==> Scroll ==> PAGE
The cause of the failure was program SAM2 in module SAM2. The COBOL source code
that immediately preceded the failure was:

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:
```

Abending statement

F8

9 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

Since source information was available, and that is important, it shows the actual statement where the abend occurred. In this case, it was a compute statement. You see the definitions for the variables involved in the failure. Scrolling forward...

Event summary showing call chain



```
File View Services Help
Saved Report
Command ==>
BALANCE-TOTAL = 10948.44
CUST-ACCT-BALANCE = X'7C7B5B6C50' *** Invalid numeric data ***

- <H1> I B M   F A U L T   A N A L Y Z E R   E V E N T   S U M M A R Y

The following events are presented in chronological order.

Event # Type      Fail Point Name  Module Name  Program Name  EP Name  Event Location (*)  Loaded F
-----
1 Call           SAM1           SAM1       SAM1       SAM1       L#312 P+D30 E+D30  DNET845.
2 Call           IGZCPAC        n/a        IGZCFCC    E+2BE      CEE.SCEE
3 Abend S0C7     *****       SAM2       SAM2       SAM2       L#89 P+39A E+39A  DNET845.

(*) One or more of the following abbreviations might appear in the "Event Location" column:
F#n Source

SAM2 was the active program

F8
```

The bad data !

The Event Summary shows the call chain

10 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

... it shows the values of the variables that were referenced by the current statement. Notice that in this case it flags a variable, "CUST-ACCT-BALANCE", as having invalid numeric data.

So at this point you know that the abend occurred in a program called SAM2, you know which statement caused it, and you even know that it abended because it was trying to perform a computation using a variable that had bad numeric data. That was all in the first part of the report, the synopsis section.

The next section is the event summary. There are some terms that are important to know in fault analyzer. One is the word "event". Any time you see that word, event, think: "items in the call chain" or "programs in the call chain". The event summary shows the call chain and other system level events.

In this example there was a main program named SAM1, that called program named SAM2. The abend occurred in the SAM2 (called) program. You see the entire call chain shown here from the main program all the way down to the program where the abend occurred.

You may also see system level modules in the call chain. For example, modules that start with CEE for language environment, or DSN for DB2®. It is not unusual to see system modules listed in the call chain even if they were not explicitly called by your application programs. Understanding the system modules can sometimes help you in your abend analysis, but more often you can ignore them.

Also notice that it shows the current locations in each program. In the case of SAM2, the current statement was line 89, shown as L#89. Scrolling down again...

Start of event details for SAM1



File View Services Help

Saved Report Line 61 Col 1 80
Command ==> Scroll ==> PAGE

identification)
L#n Source file line number
S#n Listing file statement number (refer to de
file identification)
M+x Offset from start of load module
P+x Offset from start of program
E+x Offset from start of entry point

- <H1> IBM FAULT ANALYZER EVENT DETAILS

- <H2> EVENT 1 OF 3: CALL (DSA ADDRESS 20594030)

The source code below was executed via the following sequence of P... tateme

Source
Line #

F8

The detail report for program SAM1 starts here
Scroll down to the detail report for SAM2

11 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

Next there is a detail report for each program in the call chain. The next section is the detail report for event 1 of 3, which is the main program SAM1. Scrolling forward until the detail report for the abending program is reached.

Start of event details for SAM2



File View Services Help

Saved Report Line 553 Col 1 80
Command ==>

- <H2> EVENT 3 OF 3: ABEND S0C7

***** POINT OF FAILURE *****

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:

F8

12 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

Here is the beginning of the detail section for program SAM2, where the abend occurred. SAM2 was the third event in the call chain.

Source code involved in abend



```
File View Services Help
Saved Report Line 573 Col 1 80
Command ==> Scroll ==> PAGE
Source
Line #
-----
-5          GOBACK.
-4          100-CALC-BALANCE-STATISTICS.
-3          MOVE 'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
-2          *   *** Increment Record Count ***
-1          ADD +1 TO BALANCE-COUNT
000088      *   *** Add this customer's BALANCE to the grand total ***
000089      COMPUTE BALANCE-TOTAL =
000090          BALANCE-TOTAL + CUST-ACCT-BALANCE
+1          *   *** Calculate Average ***
+2          COMPUTE BALANCE-AVERAGE =
+3              BALANCE-TOTAL / BALANCE-COUNT
+4          *   *** Calculate Minimum ***
+5          IF WS-FIRST-TIME-SW = 'Y'
```

Data Field Declarations:

Source

Active statement at the time of abend

F8

13 | IBM Fault Analyzer for z/OS - V12 Tutorial | © 2012 IBM Corporation

You see the active statement in this program, a COMPUTE statement. Several lines of code before and after the abending statement are also shown to help you locate the statement in the program. Scrolling forward...

Value of field causing abend



```
File View Services Help
Saved Report Line 593 Col 1 80
Command ==> Scroll ==> PAGE
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:

BALANCE-TOTAL      = 10948.44
CUST-ACCT-BALANCE = X'7C7B5B6C50' *** Invalid numeric data ***

The SYSDEBUG file used for the above was found in ADTOOLS.ADLAB.SYSDEBUG(SAM2).

Load Module Name. . . . . : DNET845.ADLAB.LOAD(SAM2)
At Address. . . . . : 0003F000
Load Module Length. . . . : X'1428'
Link-Edit Date and Time . : 2009/10/06  22:19:20

Program and Entry Point Name: SAM2
At Address. . . . . : 0003F000 (Module SAM2 offset X'0')
Program Length. . . . . : X'7FB'
```



Next, you see the variables referenced by the active statement and their values. A message is shown stating that a Sysdebug file was found for this program. Fault analyzer is using information found in the Sysdebug file to display program statements and variables. One of the advantages of Sysdebug files is that fault analyzer in many cases can find them automatically. There is more information about how it finds files for source mapping in later sections.

After that comes information about the load module. The load module's storage address is shown, along with the linkage editor time stamp. Next is the program entry point name. Notice the address where the program was loaded. If you were so inclined, you could do your abend analysis the old-fashioned way, where you sit down with the dump, a compiler listing, and a hexadecimal calculator. Then you would calculate offsets in the program to find your variables and statements. Fault analyzer provides all the information you need to do that. But of course one of the big advantages of having fault analyzer is that you do not have to do that. That is the purpose of source mapping, and it is one of the main reasons you use it. So typically you can ignore this address. Scrolling forward...

Compiler information



```
File View Services Help
Saved Report Line 613 Col 1 80
Command ==> Scroll ==> PAGE
Program Language. . . . . : COBOL (Compiled using IBM Enterprise COBOL for
                           : z/OS and OS/390 V3 R4 M1 on 2009/10/06 at
                           : 22:19:19)
Compiler Options Used . . . : ADV QUOTE ARITH(COMPAT) NOAWO NOCICS NOCURRENCY
                           : DATA(31) NODATEPROC DBCS NODECK NODLL NODUMP DYNAM
                           : NOEXPORTALL NOFASTSRT INTDATE(ANSI) LIB LIST MAP
                           : NONAME NONUMBER OBJ NOOFFSET NOOPTIMIZE
                           : OUTDD(SYSOUT) NOPFDSGN NORENT RES RMODE(24)
                           : SEQUENCE SIZE(MAX) SOURCE NOSQL NOSSRANGE NOTERM
                           : TEST(ALL SYM SEPARATE) NOTHREAD TRUNC(STD) NOVNBREF
                           : NOWORD XREF ZWB
Machine Instruction . . . . : FA442005301E AP 5(5,R2),30(5,R3)
At Address. . . . . : 0003F39A (Program SAM2 offset X'39A')
AMODE . . . . . : 31
Failing Operand . . . . . : Second operand
First Operand Address . . : 00009115 (Module SAM2 program SAM2 LINKAGE SECTION
                           : BLL=0002 + X'5', symbol BALANCE-TOTAL, so
                           : # 66 - 241387 bytes of storage addressable)
First Operand Length. . . : 5
```

15

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

It shows which compiler was used and the compile time stamp. Time stamps can be very important. If you are in a situation where you get one of those 2:00 in the morning telephone calls because a program abended, it is a good idea to look at the time stamps to see if the program was compiled, say, just the previous afternoon. Sometimes a recent recompile can be a red flag that something has changed. In any case, it is good information.

The next thing shown is the machine instruction of the active statement. In this case it was an AP (add packed). Generally when you are working with high level programs like COBOL, PLI, or C, you do not spend time working with the machine instruction. But when you are working with an assembler program, this can be very useful information.

General purpose registers



```
File View Services Help
Saved Report Line 633 Col 1 80
Command ==> Scroll ==> PAGE
First Operand Storage . . : 00109484 4C *..md<*
Second Operand Address. . : 00023FA6 (Module SAM2 program SAM2 LINKAGE SECTION
                                BLL=0001 + X'1E', symbol CUST-ACCT-BALANCE, source
                                line # 59 - 131162 bytes of storage addressable)
Second Operand Length . . : 5
Second Operand Storage. . : 7C7B5B6C 50 *@#$$%&*

Program Status Word (PSW) . : 078D2000 8003F3A0

General Purpose Registers:
R0: 20594600 (592376 bytes of storage addressable)
R1: 205941A0 (593504 bytes of storage addressable)
R2: 00009110 (Module SAM2 program SAM2 LINKAGE SECTION BLL=0002 + X'0',
             symbol CUST-BALANCE-STATS, source line # 64)
R3: 00023F88 (Module SAM2 program SAM2 LINKAGE SECTION BLL=0001 + X'0',
             symbol CUST-REC, source line # 53)
R4: 0003F038 (Module SAM2 program SAM2 + X'38')
R5: 0000B1BC (233028 bytes of storage addressable)
R6: 205910C0 (606016 bytes of storage addressable)
R7: 2050DA20 (1144288 bytes of storage addressable)
```



Next the general-purpose registers are shown. Again, when working with high level programs, you may not typically need to know register values. But with assembler programs this data can be critical. Notice that it shows where the registers are pointing. For example, if register three is an address, in this case it is pointing to a variable named CUST-REC in program SAM2. And that variable happens to be defined at source line number 53 in the program.

Associated message information



The screenshot shows the IBM Fault Analyzer interface. At the top, there is a menu bar with 'File', 'View', 'Services', and 'Help'. Below the menu bar, the text 'Saved Report' and 'Command ==>' is visible. On the right side, there is a status bar showing 'Line 663 Col 1 80' and 'Scroll ==> HALF'. A callout box with the text 'Automatic lookup of system message' points to the 'Associated Messages' section. The main content area displays the following text:

```
- <H3> Associated Messages
```

CEE3207S The system detected a data exception (System Completion Code=0C7).
CEE3207S The system detected a data exception (System Completion Code=0C7).
Explanation: Your program attempted to use a decimal instruction incorrectly.
See a Principles of Operation manual for a full list of data exceptions.
Programmer response: Check the variables associated with the failing statement
to make sure that they have been initialized correctly.
System action: The thread is terminated.
Symbolic Feedback Code: CEE347

At the bottom right of the main content area, there is a yellow box with the text 'F8'. Below the main content area, there is a dashed line and the text '- <H3> Associated Storage Areas'.

At the bottom of the screenshot, there is a footer bar with the page number '17', the text 'IBM Fault Analyzer for z/OS - V12 Tutorial', and the copyright notice '© 2012 IBM Corporation'.

Scrolling down again, you see “Associated messages”. Fault Analyzer reports messages that were produced by the system. It automatically looks them up in Bookmanager, so you do not have to, and reports the explanation text for messages and error codes from the appropriate manuals.

Associated open files



```
File View Services Help
Saved Report
Command ==>
Files open to this program
Line 103 Col 1 80
Scroll ==> HALF
-----
- <H3> Associated Open Files
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 *

```

F8

18 | IBM Fault Analyzer for z/OS - V12 Tutorial | © 2012 IBM Corporation

Next is the “associated open files”. Here you see a list of files that were open to this program. You see the DD name, the file name, how the file was opened, and the record buffers.

Associated open files




```
File View Services Help
Saved Report Line 160 Col 1 80
Command ==> Scroll ==> HALF

File Name . . . . . : TRANFILE
Data Set Name . . . . . : DNET845.DNET845X.JOB15885.D0000102.?
File Attributes . . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=SEQUENTIAL,
                           RECFM=FIXED
Last I/O Function . . . . . : READ
Open Status . . . . . : INPUT
File Status Code. . . . . : 0

Current Record. . . . . : Record data length 80
Address  Offset      Hex                                EBCDIC
-----
00019008          D7D9C9D5 E3404040 40404C7E 7E40D7D9 *PRINT <== PR*
00019018      +10  C9D5E340 C3E4E2E3 D6D4C5D9 40D3C9E2 *INT CUSTOMER LIS*
00019028      +20  E3404040 40404040 40404040 40404040 *T *
00019038      +30  40404040 40404040 40404040 40404040 * *
           Line 00019048 same as above

-----
- <H3> Associated Storage Areas
```



19

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

Scrolling down again, here is another file. The DD name, file name, and other items are reported. You will see this level of detail for every file that was open to this program at the time of the abend. The current record buffer shows the record that the program was processing at the time of the abend. The previous and next record buffers may also be shown, if they happen to be in memory at the time of the abend. That can be helpful when you are trying to locate a record in a file.

Associated storage areas



File View Services Help

Saved Report
Command ==>

- <H3> Associated Storage Areas

This section displays all program variables for a program

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

WORKING-STORAGE SECTION

Off	Hex Value	Data Value	Source (Starting at
- <H5> BLW=0000 at address 0003F7C8			
			01 WS-FIELDS.
0	C3C1D3C3 E4D3C1E3 C9D5C740 C2C1D3C1	*CALCULATING BALA*	05 WS-PROGRAM-
10	D5C3C540 E2E3C1E3 E2404040 4040	*NCE STATS	*
1E	D5	*N	* 05 WS-FIRST-TI
1F	0000000C	0	05 WS-WORK-NUM
23	0000000C	0	05 WS-WORK-NUM
27	0000000C	0	05 WS-WORK-NUM
2B	0000000C	0	05 WS-WORK-NUM
2F	0000000C	0	05 WS-WORK-NUM

LINKAGE SECTION

20

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

F8

The next section, “associated storage areas” can be the most important part of the real-time report, because it shows the variables and data values in the program.

Working storage display



```
File View Services Help
Saved Report Line 297 Col 1 80
Command ==> Scroll ==> HALF

WORKING-STORAGE SECTION ← Working-Storage
Off Hex Value Data Value Source (Starting at)
-----
- <H5> BLW=0000 at address 00009060
01 SYSTEM-DATE-AND
05 CURRENT-DAT
0 F1F0 10 10 CURRENT
2 F0F2 2 10 CURRENT
4 F2F3 23 10 CURRENT
05 CURRENT-TIM
6 F1F5 15 10 CURRENT
8 F4F5 45 10 CURRENT
A F0F2 2 10 CURRENT
C F4F7 47 10 CURRENT
01 WS-FIELDS.
* 05 WS-CUSTFILE
10 F0F0 * 05 WS
12 F0F0 * 05 WS
14 F0F0 *00 * 05 WS
16 40 * * 05 WS
```

21 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

Scroll to the right to see variable names

F11

For a COBOL program, as in this example, you will see the different parts of storage, such as file section, working storage, and linkage. Here you see a heading for the working storage section. Variable values are shown on the left side of the report both in hex and character. You can scroll to the right with F11.

Working storage variables after scrolling



```
File View Services Help
Saved Report Line 297 Col 40 119
Command ==> Scroll ==> HALF

Data Value Source (Starting at Line # 000095)
-----
01 SYSTEM-DATE-AND-TIME.
05 CURRENT-DATE.
10 CURRENT-YEAR PIC 9(2).
2 CURRENT-MONTH PIC 9(2).
23 CURRENT-DAY PIC 9(2).
05 CURRENT-TIME.
15 CURRENT-HOUR PIC 9(2).
45 CURRENT-MINUTE PIC 9(2).
2 CURRENT-SECOND PIC 9(2).
47 CURRENT-HNDSEC PIC 9(2).
01 WS-FIELDS.
*00 * 05 WS-CUSTFILE-STATUS PIC X(2) VALUE
*00 * 05 WS-TRANFILE-STATUS PIC X(2) VALUE
*00 * 05 WS-REPORT-STATUS PIC X(2) VALUE
* * 05 WS-TRAN-FILE-EOF PIC X VALUE SI
```

Program variables
(after scrolling to
the right)

F10

F8

22 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

On the right side of the report are the variable names and their definitions. In some programs Associated Storage Areas is a very large report section, because of the large numbers of variables. Here F10 is pressed to scroll back to the left, and F8 a couple of times to scroll down.

Start of system-wide information



```
File View Services Help
Saved Report Line 733 Col 1 80
Command ==> Scroll ==> PAGE
- <H1> S Y S T E M - W I D E I N F O R M A T I O N

=====
- <H2> OPEN FILES

-----
- <H3> Non-Event-Related Open Files

File Name . . . . . : CEEDUMP
Data Set Name . . . . . : DNET845.DNET845X.JOB15885.D0000112.?
File Attributes . . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=n/a,
                           RECFM=FIXED BLOCKED ASA
Last I/O Function . . . . . : WRITE
Open Status . . . . . : OUTPUT

Current Record . . . . . : Record data length 133
Address  Offset  Hex                                     EBCDIC
-----
20621000      40404040 40404040 40404040 40404040 *
```

System Files

F8

23 IBM Fault Analyzer for z/OS - V12 Tutorial © 2012 IBM Corporation

That is an example of what is shown in a detail report for a program. There is a detail report for every application program in your call chain. The next section is System-wide information. The event detail reports show items that are related to each program. By contrast, this section shows items that are common to all parts of the application. For example, here is information about a file with a DD name of CEEDUMP that was opened by the system.

System files (continued)



```
File View Services Help
-----
Saved Report                                     Line 753 Col 1 80
Command ==>                                     Scroll ==> PAGE
      Lines 20621010-20621070 same as above
20621080      +80  40404040 40                  *                  *

File Name . . . . . : SYSOUT
Data Set Name . . . . : DNET845.DNET845X.JOB15885.D0000108.?
File Attributes . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=n/a,
                        RECFM=FIXED BLOCKED ASA
Last I/O Function . . . . : WRITE
Open Status . . . . . : OUTPUT

Current Record. . . . . : Record data length 121
Address  Offset      Hex                                     EBCDIC
-----
205BD000      40404040 40404040 40404040 40404040 *                  *
      Lines 205BD010-205BD060 same as above
205BD070      +70  40404040 40404040 40                  *                  *

=====
- <H2> STORAGE AREAS
```



Scrolling forward, you see other system files.

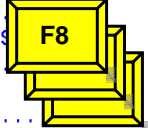
Hex dumped storage report



```
File View Services Help
Saved Report Line 773 Col 1 80
Command ==> Scroll ==> PAGE

- <H3> Hex-Dumped Storage

Address Offset Hex EBCDIC
-----
Event 1 Program SAM1 GPR 9 (Address 000088B8)
Event 1 Program SAM1 TGT (Address 000088B8)
000088B8 00000000 00000000 * .....*
000088C0 +8 00000000 00000000 00000000 00000000 *.....*
Lines 000088D0-000088F0 same as above
00008900 +48 F3E3C7E3 00000000 06000000 64430260 *3TGT.....-*
00008910 +58 20590100 0000B7FC 00000000 00000003 *.....*
00008920 +68 000006C4 00000000 00000000 0001A1E8 *...D.....~Y*
00008930 +78 00000000 00000000 2050DA20 00000198 *.....&.....q*
00008940 +88 00000000 00000000 00000000 00000001 *.....*
00008950 +98 E2E8E2D6 E4E34040 C9C7E9E2 D9E3C3C4 *SYSOUT IGZ
00008960 +A8 00000000 00000000 00000000 00000000 *.....*
Lines 00008970-00008990 same as above
000089A0 +E8 000070FC 00000000 00008A3C 205903F0 *.....*
```



25

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

Still in the System-wide information section, is the hex dumped storage report. This is a formatted dump of storage. Certain storage areas are identified by Fault Analyzer. For example, notice the label for “Event 1 program SAM1 TGT”. TGT is a special storage area in COBOL programs. There will also be labels for Working Storage, File FDs, and other important storage areas. Notice that Fault Analyzer gives you the information you need to find your data the hard way in storage. That is, you can still use a storage map in a compile listing to calculate offsets from the program’s starting address to locate statements and variables in storage. But with automatic program source mapping, there may be no reason to work with the storage dump. As you saw in the detail reports, program variable values can be reported automatically.

Start of job info section



```
File View Services Help
Saved Report Line 1032 Col 1 80
Command ==> Scroll ==> PAGE
- <H1> I B M FAULT ANALYZER ABEND JOB INFO

IBM Fault Analyzer Abend Job Information:
Abend Date. . . . . : 2010/02/23
Abend Time. . . . . : 15:45:02
System Name . . . . . : DEMOMVS
Job Type. . . . . : Batch
Job ID. . . . . : JOB15885
Job Name. . . . . : DNET845X
Job Step Name . . . . . : RUNSAM1
ASID. . . . . : 37
Abend TCB Address . . . . . : 006FF048
Job Execution Class . . . . . : A
Region Size . . . . . : 4M
EXEC Program Name . . . . . : SAM1
User ID . . . . . : DNET845
Accounting Information. . . : ACCTG
```

F8

26 | IBM Fault Analyzer for z/OS - V12 Tutorial | © 2012 IBM Corporation

Here is a tip: an easy way to get to the next major section of the report is to find the string “H1”, which stands for Header 1. That will position you to the next main heading of the report.

The “fault analyzer abend job info” section displays high level information, like the name of the job, the name of the step, the program on the EXEC statement, and when the application abended.

Event related programs




```
File View Services Help
-----
Saved Report                               Line 1052 Col 1 80
Command ==>                               Scroll ==> PAGE
Data Sets:

  DDname   Data Set or Path Name
  -----
  STEPLIB  DNET845.ADLAB.LOAD

Event-Related Application Programs:

The following list of event-related application programs is sorted by module
link-edit date/time and program compilation date/time in reverse chronological
order.

Module   Link-Edit      Program  Compilation
Name     Date       Time     Name      Date       Time
-----
SAM2     2009/10/06 22:19:20 SAM2      2009/10/06 22:19:19
SAM1     2009/10/06 22:19:19 SAM1      2009/10/06 22:19:19

Point Of Failure LINKEDIT Map: 
```

The “event related application programs” shows the application programs that were in the call chain, with their linkage editor and compiler time stamps. Since time stamp information can be so important, it is good to be able to see it for all of the programs in one place.

Point of failure linkedit map



```
File View Services Help
Saved Report Line 1069 Col 1 80
Command ==> Scroll ==> HALF

Point Of Failure LINKEDIT Map:

Address  Offset  Length  Type  Date      Time      RMODE  AMODE  Language  Nam
-----  -
0003F000      0    1428  MODULE 2009/10/06 22:19:20      ANY
0003F000      0      0  EP
0003F000      0    7FB  CSECT 2009/10/06 22:19:19 24  ANY  COBOL  SAM
0003F800    800    18  CSECT 2008/03/19      ANY  MIN  ASM  CEE
0003F818    818    28  CSECT 2008/03/19      ANY  MIN  ASM  CEE
0003F840    840    B0  CSECT 2008/03/19      ANY  MIN  ASM  CEE
0003F8F0    8F0    580  CSECT 2008/03/19      ANY  31  ASM  IGZ
0003FE70    E70    B8  CSECT 2008/03/19      ANY  MIN  ASM  CEE
0003FF28    F28    298  CSECT 2008/03/19      ANY  31  ASM  CEE
000401C0   11C0    E2  CSECT 2008/03/19      ANY  MIN  ASM  CEE
000402A8   12A8    70  CSECT 2008/03/19      ANY  MIN  ASM  CEE
00040318   1318    A4  CSECT 2008/03/19      ANY  MIN  ASM  CEE
000403C0   13C0    5C  CSECT 2008/03/19      ANY  MIN  ASM  CEE
00040420   1420     8  CSECT 2008/03/19      ANY  MIN  ASM  CEE
```

F8

28 | IBM Fault Analyzer for z/OS - V12 Tutorial | © 2012 IBM Corporation

The “point of failure link edit map” shows a storage map of the load module where the abend occurred. This can be especially helpful when you are working with a complex composite load module that contains more than one program.

System information



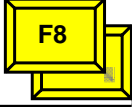
```
File View Services Help
-----
Saved Report                               Line 1088 Col 1 80
Command ==>                               Scroll ==> PAGE

Execution Environment:

Operating System . . . . . : z/OS V1R10M0
Data Facility Product . . . : DFSMS z/OS V1R10M0
Job Entry Subsystem . . . . : JES2
Language Environment . . . . : V1 R10.0
CPU Model . . . . . : 2094

Language Environment Run-Time Options:

Last Where Set          Option
-----
PARMLIB (CEEPRM00)     ABPERC (NONE)
PARMLIB (CEEPRM00)     ABTERMENC (ABEND)
PARMLIB (CEEPRM00)     NOAIXBLD
PARMLIB (CEEPRM00)     ALL31 (ON)
PARMLIB (CEEPRM00)     ANYHEAP (16384, 8192, ANYWHERE, FREE)
PARMLIB (CEEPRM00)     NOAUTOTASK
PARMLIB (CEEPRM00)     BELOWHEAP (8192, 4096, FREE)
```



29

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

The “execution environment” section shows information about the system where the abend occurred. Next is the “language environment run time options”, which you typically do not need, but sometimes it is important to have these documented just in case there was an issue with the LE options.

Fault Analyzer options



The screenshot shows the IBM Fault Analyzer interface. At the top, there is a menu bar with 'File', 'View', 'Services', and 'Help'. Below the menu bar, the text 'Saved Report' is displayed on the left, and 'Line 1164 Col 1 80' is on the right. A green prompt 'Command ==>' is followed by a horizontal line, and another green prompt 'Scroll ==>' is followed by 'HALF'. The main content area is titled 'IBM Fault Analyzer Options in Effect:' and lists the following options: Detail (Medium), NoErrorHandler, Language (ENU), NoLocale, MaxMiniDumpPages(1000) - not exceeded, NoDup(CICSfast(5),NoIncludeExecCicsDump) - not a duplicate, NoDup(ImageFast(5)) - not a duplicate, NoDup(Normal(0),NoJobname) - not enabled, NoPermitLangx, PreferredFormattingWidth(80), NoPrintInactiveCOBOL, RetainDump(Auto), Source, SpinIDIREPRT, StoragePrintLimit(256K) - not exceeded, and SystemWidePreferred(StorageAreas(Hex)). A yellow callout box with a black border and an arrow pointing to the first few options contains the text: 'These are set by the system installer, they can be overridden.' In the bottom right corner of the main content area, there is a yellow button with the text 'F8'. At the bottom of the window, there is a footer bar with '30' on the left, 'IBM Fault Analyzer for z/OS - V12 Tutorial' in the center, and '© 2012 IBM Corporation' on the right.

Scrolling down, you see the “IBM fault analyzer options” section. It shows the options that were in effect when fault analyzer processed the abend.

End of real-time analysis report



```
File View Services Help
Saved Report Line 1187 Col 1 80
Command ==> Scroll ==> HALF
preallocated, specified via DataSets options, or provided as defaults.

DDname Data Set or Path Name
-----
IDIBOOKS FAULTANL.V10R1.SIDIBOOK
IDIDOC FAULTANL.V10R1.SIDIDOC1
IDIHIST FAULTANL.V10R1.HIST
IDIMAPS FAULTANL.V10R1.SIDIMAPS
IDIVSENU FAULTANL.V10R1.IDIVSENU

Fault Analyzer was invoked via the LE CEEEXTAN exit (IDIXCEE). Its execution
time was 0.83 seconds. It used 0.25 CPU seconds and 1.43 megabytes of allocated
storage.

Fault ID F00905 assigned in history file FAULTANL.V10R1.HIST.

End of Fault Analyzer report (2010/02/23 15:45:03).

*** Bottom of data.
```

Name of the Fault History File and Fault Entry ID

Scrolling down again, you see files and resources that fault analyzer used when it processed the abend. And finally at the bottom, you see the fault ID and the name of the fault history file where the fault entry was stored.

That is the end of the example of viewing a real time report. To learn more about how you can put all this information together, view the tutorial sections about “Interactive reanalysis”, where you will see a root-cause analysis of an abend.

Feedback



Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_FAv12s05ViewRealTimeReport.ppt

This module is also available in PDF format at: [../FAv12s05ViewRealTimeReport.pdf](..FAv12s05ViewRealTimeReport.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, CICS, DB2, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

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, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.