

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 use the interactive reanalysis function. You will see how to start your interactive reanalysis session, and you will see an example of performing root cause analysis.

Why do reanalysis?



- Interactive reanalysis gives you easy navigation to abend and program information source
- If the abending program's source listing was not available to Fault Analyzer when the abend occurred
 - In the Real-Time Report, you got:

```
A system abend 0C7 occurred in module IDISCBL1
CSECT IDISCBL1 at offset X'3D0',
The abend was caused by machine instruction FD73D0B8D0A8(DP)
```

- You can apply a side file or compiler listing
 - Then with Reanalysis, you can get:

```
The cause of the failure was program IDISCBL1.
The COBOL source code that immediately preceded the failure was:
Source
Line #
-----
000051      DIVIDE NUMBERX BY ERROR_COUNT GIVING BAD_RESULT
```

3

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

When an abend occurs, fault analyzer automatically produces a real time report. So the question arises, why do a RE-analysis? Interactive reanalysis gives you much easier navigation to get to the information you are looking for compared to using a real time report. In addition, if source information was not available to fault analyzer when the abend occurred, it might report information like you see in the first box, where it shows the offset of the abend and machine instruction. But to become more productive you want information more like what you see in the second box, where it gives you source level information, including the abending source statement and the values of program variables. When you use interactive reanalysis, you can apply side files and compiler listings after the fact; after an abend has occurred, without having to re-create the abend.

Analyze an abend (1 of 20)



```
File Options View Services Help
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==> _____ ==> PAGE
Fault History File or View : 'FAULTANL'
{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 Program Offset Dups User_ID Sys/Job Abend
i F00905 DNET845X JOB15885 SAM2      39A      DNET845 DEMOMVS SOC7
  F00882 DNET845X JOB15573 SAM2      39A      4 DNET845 DEMOMVS SOC7
  F00881 DNET845X JOB15572 SAM2      39A      DNET845 DEMOMVS SOC7
  F00880 DNET845X JOB15571 SAM2      39A      DNET845 DEMOMVS SOC7
  F00878 DNET845X JOB15535 SAM2      39A      DNET845 DEMOMVS SOC7
  F00872 DNET845Y JOB15410 PSAM2     3DA      DNET845 DEMOMVS SOC7
  F00871 DNET845P JOB15408 PSAMM2    27A      DNET845 DEMOMVS SOC7
  F00869 DNET845X JOB15387 SAM2      39A      DNET845 DEMOMVS SOC7
** Bottom of data.
```

The **I** line command starts an interactive reanalysis session.

Enter

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

From the Fault Entry List, start interactive reanalysis by typing an **I** line command next to the abend you want to analyze. Press enter.

Analyze an abend (2 of 20)



```
File View Services Help
Interactive Reanalysis Report
Command ==>
JOBNAME: DNET845X  SYSTEM ABEND: 0C7

Fault Summary:
Module SAM2, program SAM2, source line # 89 : Abend S0C7 (Data 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: 1.68 megabytes.}

*** Bottom of data.
```

Col 1 8
==> HAL
15:45:0

Debug Clues:
✓ Abended in program SAM2 because of a data exception

What is information is in the Synopsis?

Enter

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

The interactive reanalysis menu is displayed. To research the root cause of a problem, you will make observations about the abend, come up with theories about what happened, and then test those theories. Already on this first screen there is some useful information. You see that the application abended in a program named SAM2, and that there was a data exception.

As this example is researched, a few notes will be shown on the right side of the screen. Here it was noted that the abend occurred in program SAM2 because of a data exception. In this section of the tutorial, you will be given questions, you will have a few moments to consider it, and you can pause the tutorial if you would like more time.

Notice is that there are some yellow highlighted fields. These are called point and shoot fields. If you use the tab key on your keyboard, the cursor will jump right to those yellow highlighted fields. When your cursor is on one of them and you press enter, It jumps to the screen indicated by the selection.

This is the main menu for interactive reanalysis, and options are: synopsis, event summary, and others. These are the same sections that are shown in real-time reports, but the menu gives you an easier way to get directly to a section you want to see. Here is the first question, and it is a review from a previous section, "viewing a real-time report". "What information does the synopsis section show?" Pause the tutorial now if you would like more time.

The synopsis is a good place to start your research. It contains an explanation of the abend code, and when available shows the program statement or machine instruction that caused the abend. To see it, tab to the point-and-shoot field for the synopsis section, and press enter.

Analyze an abend (3 of 20)



File View Services Help

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

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.

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: **F8**

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

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

Page forward to see active variables

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

Now the synopsis is displayed. You see that the abend occurred in a program called SAM2. Also notice that the source statement that caused the abend is shown - it was a COMPUTE statement.

Here is another question. What can happen in a program that can cause a data exception?

Fault analyzer is showing the description: "A decimal digit or sign was invalid". A data exception is caused when a program attempts an arithmetic operation, but one of the data elements contains non-numeric data. Scrolling down further in the synopsis, F8.

Analyze an abend (4 of 20)

IBM

File View Services Help

Synopsis
Command ==>

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

Source
Line #
000088 * *** Add this customer's BALANCE to the grand total ***
000089 COMPUTE BALANCE-TOTAL =
000090 BALANCE-TOTAL + CUST-ACCT-BALANCE

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

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

Data field values at time of abend:

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

*** Bottom of data.

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

Go look at the bad variable

Return to menu

F3

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

You can see the COMPUTE statement that abended. Here is the next question: What variable contained the bad data?

Near the bottom, fault analyzer flagged variable CUST-ACCT-BALANCE. Notice that it is defined as a comp-3 field, which is packed decimal. If you are familiar with packed decimal fields, you can see from the hexadecimal representation that it had bad data.

Now the local cause of the abend is known, although not necessarily yet the root cause. The goal of this exercise is to understand the problem in enough detail that it can actually be fixed, and the program can run again without an abend. More research is needed to find the root cause. F3 returns to the menu.

Analyze an abend (5 of 20)



```
File View Services Help
Interactive Reanalysis Report                               Line 1 Col 1 8
Command ==> 2                                           Scroll ==> PAG
JOBNAME: DNET845X  SYSTEM ABEND: 0C7                    DEMOMVS  2010/02/23 15:45:0

Fault Summary:
Module SAM2, program SAM2, source line # 89 : Abend SOC7 (Data 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: 1.68 megabytes.}

*** Bottom of data.
```

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data

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

Enter

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

Consider the strategy you might use to research an abend. There are a lot of different ways to go about it.

In this example, you already know that the abend occurred in program SAM2, and that the abend was caused because of bad data in a variable called CUST-ACCT-BALANCE. From here you could go a lot of directions to get more information. For example you could look at the source listing to understand the logic of the program to try to understand where the bad data came from. In this case since there is bad data in a variable, you will see a “follow the data” strategy used, tracking the bad data back to it’s origin.

The next step in this example is to look at the definition of the bad variable CUST-ACCT-BALANCE in SAM2, to see if other clues can be found there that will help with the analysis. One way to get there is through the event summary. You can either tab to the Event Summary point-and-shoot field, or specify option 2 on the command line.

Analyze an abend (6 of 20)



File View Services Help

Event Summary Line 1 Col 1 8
 Command ==> Scroll ==> PAG
 JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

{The following events are presented in chronological order.}

Event #	Type	Fail Point	Module Name	Program Name	EP Name	Event Location (*)	Load
1	Call		SAM1	SAM1	SAM1	L#312 P+D30 E+D30	DNET8
2	Call		IGZCPAC	n/a	IGZCFCC	E+2BE	CEE.S
3	Abend	S0C7	***** SAM2	SAM2	SAM2	L#89 P+39A E+39A	DNET8

(*) One or more of the following abbreviations are used in the "Event Location" column:
 F#n Source file number (refer to detailed event information for file identification)
 L#n Source file line number
 S#n Listing file statement number (refer to detailed event information for file identification)
 M+x Offset from start of load module

Enter

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data

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

Select detail for program SAM2

What is the relationship between programs SAM1 and SAM2?

The event summary shows the program call chain. The next question is: "Based on what is shown, what is the relationship between programs SAM1 and SAM2?". Pause the tutorial now if you would like more time.

The program at the top of the list is the first program on the call chain. Typically, this is the main program. In this case, SAM1 is the main program, and SAM2 is a subroutine that was called by SAM1. You may be wondering about the IGZ module in the middle. That is a system routine that serviced the call request. You may often see system modules in the call chain, and very often, as in this case, you can ignore them.

The variable of interest is in SAM2, where the abend occurred. So tab down to event number 3, which is program SAM2, and press enter.

Analyze an abend (7 of 20)



File View Services Help

Event 3 of 3: Abend SOC7 *** Point of Failure *** Line 1 Col 1 8
Command ==> **bottom** Scroll ==> HAL
JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

Previous Event Details

Abend Code : SOC7
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 #	Code
	000079	PERFORM 100-CALC-BALANCE-STATISTICS.

COBOL Source Code:

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

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data

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

Go to the bottom of the program detail display

Enter

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

Now the detail report for event 3 is displayed. Again, the next step is to see variable CUST-ACCT-BALANCE and how it is defined. So go to the bottom of this section. One way to get there is with the “bottom” command.

Analyze an abend (8 of 20)



File View Services Help

Event 3 of 3: Abend SOC7 *** Point of Failure *** Line 83 Col 1 8
Command ==> Scroll ==> PAG

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

R6: 205910C0 (606016 bytes of storage addressable)
R7: 2050DA20 (1144288 bytes of storage addressable)
R8: 0003F7C8 (Module SAM2 program SAM2 WORKING-STORAGE SECTION BLW=0000 + X'0', symbol WS-FIELDS, source line # 36)
R9: 0003F5D0 (Module SAM2 program SAM2 + X'5D0', source line # 116)
R10: 0003F11C (Module SAM2 program SAM2 + X'11C')
R11: 0003F29C (Module SAM2 program SAM2 + X'29C')
R12: 0003F0FC (Module SAM2 program SAM2 + X'FC')
R13: 20594458 (592808 bytes of storage addressable)
R14: 8003F392 (Module SAM2 program SAM2 + X'392', source line # 89)
R15: 8003F224 (Module SAM2 program SAM2 + X'224')

Associated Messages

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

Associated Storage Areas

*** Bottom of data.

What is shown here?

Enter

Debug Clues:
✓ Abended in program SAM2 because:
✓ CUST-ACCT-BALANCE had bad numeric data

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

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

The cursor is tabbed to the point-and-shoot field for “associated storage areas”. Here is the next question: “What information is shown in associated storage areas?”

Pressing Enter...

Analyze an abend (9 of 20)



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

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

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

LINKAGE SECTION
```

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data

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

Enter

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

Associated storage areas shows the program's variables, and possibly other areas if the program explicitly allocated storage. An easy way to locate a variable is with a Find command. Type in F, space, and the variable name on the command line, and press enter.

Analyze an abend (10 of 20)



File View Services Help

Associated Storage Areas Line 18 Col 40 11
Command ==> Scroll ==> HAL

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

Here is CUST-ACCT-BALANCE, it is a packed decimal field (COMP-3),

ss

Data Value	Source (Starting at Line # 000053)
	01 CUST-REC.
	05 CUST-KEY.
*54321	10 CUST-ID PIC X(5).
*C	10 CUST-RECORD-TYPE PIC X.
*	10 FILLER PIC X(7).
0 *Aster, Dez	05 CUST-NAME PIC X(17).
*	
*@##\$%&	05 CUST-ACCT-BALANCE PIC S9(7)V99 COMP-3.
2	05 CUST-ORDERS-YTD PIC S9(4) COMP.
*Stormy Falls	05 CUST-CITY PIC X(15).
1 *Data Entry Opera*	05 CUST-OCCUPATION PIC X(28).
*tor	

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data (it is part of CUST-REC).

Here is the bad variable, CUST-ACCT-BALANCE

Scroll left

F11

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

Here is the variable. Notice that it contains the bad data. It does not look very numeric. Scrolling over to the right, F11.

Analyze an abend (11 of 20)



File View Services Help

Associated Storage Areas
Command ==> It is part of CUST-REC Col 55 13
JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

Source (Starting at Line # 000053)

```
01 CUST-REC.  
05 CUST-KEY.  
* 10 CUST-ID PIC X(5).  
* 10 CUST-RECORD-TYPE PIC X.  
* 10 FILLER PIC X(7).  
* 05 CUST-NAME PIC X(17).  
*  
* 05 CUST-ACCT-BALANCE PIC S9(7)V99 COMP-3.  
* 05 CUST-ORDERS-YTD PIC S9(4) COMP.  
* 05 CUST-CITY PIC X(15).  
pera* 05 CUST-OCCUPATION PIC X(28).  
*
```

Source (Starting at Line # 000064)

F10

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data (it is part of CUST-REC)

Here is the bad variable, CUST-ACCT-BALANCE

Scroll right

14

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

Notice that the bad variable is within a data structure called CUST-REC. Scroll left to see more information.

Analyze an abend (12 of 20)



File View Services Help
Line 18 Col 1 80

Associated Storage Areas
Command ==>
Scroll ==> HALF

JOBNAME: DNET845X SYSTEM ABEND: 0C7
DEMOMVS 2010/02/23 15:45:02

LINKAGE SECTION

BLL=0000 has not been assigned an address

Off	Hex Value	Data Value	Source (Starting a
BLL=0001	at address 00023F88		
			01 CUST-REC.
			05 CUST-KEY.
0	F5F4F3F2 F1	*54321	* 10 CUST-
5	C3	*C	* 10 CUST-
6	40404040 404040	*	* 10 FILLE
D	C1A2A385 996B40C4 85A94040 40404040	*Aster, Dez	* 05 CUST-NAME
1D	40	*	*
1E	7C7B5B6C 50	*@#%&	* 05 CUST-ACCT
23	0002	2	05 CUST-ORDE
25	E2A39699 94A840C6 819393A2 404040	*Stormy Falls	* 05 CUST-CITY
34	C481A381 40C595A3 99A840D6 97859981	*Data Entry Opera*	05 CUST-OCCU
44	A3969940 40404040 40404040	*tor	*

It is in Linkage Section.
What does that indicate?

Debug Clues:

- ✓ Abended in program SAM2 because:
- ✓ CUST-ACCT-BALANCE had bad numeric data (it is part of CUST-REC).

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

Return to program detail

F3

15
IBM Fault Analyzer for z/OS - V12 Tutorial
© 2012 IBM Corporation

Notice that the bad variable is in linkage section. And here is the next question: "Since it is in linkage section, what can be said about where the data came from?". A little experience with COBOL is needed to answer this one.

In COBOL, data is passed between a calling program and a called program in Linkage section. So the bad data in CUST-ACCT-BALANCE may have been passed from the main program.

Part of troubleshooting an abend involves developing theories about what may have caused it, and then testing those theories. So now there is a theory - the bad data may have been passed to this program from the main program. The bad variable is in a data structure called CUST-REC. COBOL programs pass data at the 01 data structure level. Also notice that CUST-REC is the first item in linkage, which means it is the first parameter in the list passed by the calling program. That will be helpful information in just a minute.

To test the theory, the next step is to go back to the main program and see what data it passed. F3 to exit.

Analyze an abend (13 of 20)



File View Services Help

Event 3 of 3: Abend SOC7 *** Point of Failure *** Line 83 Col 1 8
Command ==> Scroll ==> HAL

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

R6: 205910C0 (606016 bytes of storage addressable)
R7: 2050DA20 (1144288 bytes of storage addressable)
R8: 0003F7C8 (Module SAM2 program SAM2 WORKING-STORAGE SECTION BLW=0000 + X'0', symbol WS-FIELDS, source line # 36)
R9: 0003F5D0 (Module SAM2 program SAM2 + X'5D0', source line # 116)
R10: 0003F11C (Module SAM2 program SAM2 + X'11C')
R11: 0003F29C (Module SAM2 program SAM2 + X'29C')
R12: 0003F0FC (Module SAM2 program SAM2 + X'FC')
R13: 20594458 (592808 bytes of storage addressable)
R14: 8003F392 (Module SAM2 program SAM2 + X'392', source line # 89)
R15: 8003F224 (Module SAM2 program SAM2 + X'224')

Associated Messages

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

Associated Storage Areas

*** Bottom of data.

Debug Clues:

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

Go look at the passed data

Return to the events list

F3

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

That returned to the detail report for program SAM2. F3 again...

Analyze an abend (14 of 20)



File View Services Help

Event Summary Line 1 Col 1 8
 Command ==> Scroll ==> HAL
 JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

{The following events are presented in chronological order.}

Event #	Type	Fail Point	Module Name	Program Name	EP Name	Event Location (*)	Load
1	Call		SAM1	SAM1	SAM1	L#312 P+D30 E+D30	DNET8
2	Call		IGZCPAC	n/a	IGZCFCC	E+2BE	CEE.S
3	Abend	S0C7	***** SAM2	SAM2	SAM2	L#89 P+39A E+39A	DNET8

(*) One Loc SAM1 called SAM2. Next look at details for SAM1. "Event

F#n Source file number (refer to detailed event information for file identification)
 L#n Source file line number
 S#n Listing file statement number (refer to detailed event information for file identification)
 M+x Offset from start of load module

Enter

Debug Clues:

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

Go look at the passed data

... returns to the event summary. Now, tab to event one, which is the main program, SAM1. Press enter.

Analyze an abend (15 of 20)



```
File View Services Help
Event 1 of 3: Call (PSA Address 20594
Command ==> bottom
JOBNAME: DNET845X SYSTEM ABEND: 0C7

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

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

Data Field Declarations:
Source
```

What passed variable contained the bad data?

This is the CALL statement

Enter

Debug Clues:

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

Go look at the passed data

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

That displays the detail report for the main program. The active statement in the main program is shown; it is the CALL statement for the subroutine. You can see the names of the variables that were passed. Here is the next question, which passed variable might have contained the bad data?

In the subprogram, you just saw that the bad data was in the first passed parameter. So, you know that variable CUST-REC is the one of interest, because it is the first variable passed by the CALL statement. The next step is to look at the data in CUST-REC, to verify that it has the bad data. One way to see it is to go to the associated storage areas. The “bottom” command is used.

Analyze an abend (16 of 20)



File View Services Help

Event 1 of 3: Call (DSA Address 20594030) Line 68 Col 1
Command ==> Scroll ==> P

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45

R10: 00007124 (Module SAM1 program SAM1 + X'124')
R11: 00007798 (Module SAM1 program SAM1 + X'798')
R12: 000070FC (Module SAM1 program SAM1 + X'FC')
R13: 20594030 (593872 bytes of storage addressable)
R14: 80007D32 (Module SAM1 program SAM1 + X'D32', source line # 312)
R15: A05142B0 (Module IGZCPAC + X'2B0')

Associated Open Files

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

Associated Storage Areas

Next Event Details

*** Bottom of data.

Enter

Debug Clues:

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

Look at CUST-REC

Go to variables and storage for this program

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

The cursor is tabbed to the point-and-shoot field for “associated storage areas”. Press enter.

Analyze an abend (17 of 20)



File View Services Help
Line 1 Col 1 8

Associated Storage Areas

Command ==>

JOBNAME: DNET845X SYSTEM ABEND: 0C7

Task Global Table (TGT) at address 000088

FILE SECTION (File CUSTFILE)

Collapse hex

Off	Hex	Value	Data	Value	Source (Starting
BLF=0000 at address 00023F88					
0	F5F4F3F2	F1	*	54321	01 CUST-REC.
5	C3		*	C	05 CUST-KEY
6	40404040	404040	*		10 CUST
D	C1A2A385	996B40C4 85A94040 40404040	*	Aster, Dez	10 FILL
1D	40		*		05 CUST-NAM
1E	7C7B5B6C	50	*	@#\$\$%	05 CUST-ACC
23	0002		*	2	05 CUST-ORD
25	E2A39699	94A840C6 819393A2 404040	*	Stormy Falls	0
34	C481A381	40C595A3 99A840D6 97859981	*	Data Entry Opera	0
44	A3969940	40404040 40404040	*	tor	0

Debug Clues:

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

Here is CUST-REC

It is in File Section. What does that indicate?

What is the file DD name?

The bad data

*#\$\$%

F3

In the associated storage areas, you can see the program variables. CUST-REC happens to be near the top of the section, although if it were not you could use a Find command to locate it quickly. You see that the CUST-REC structure contains CUST-ACCT-BALANCE. Of course, you could scroll to the right to see the rest of the variable name. The same bad data is shown, so that validates that the data has been tracked back to here. So far, the theory that the bad data was passed from the calling program still makes sense.

There is another clue on this screen about where the data may have come from. And here is the next question. "This data is in File Section. What does that mean about where the data may have come from?"

File Section has variables that are read from or written to records in files. Since the bad data is in File section, another reasonable theory is that the data may have been read from a file. The DD name is CUSTFILE.

In the next step the file buffers are examined, to see if the bad data was in a record that was read from the file. F3...

Analyze an abend (18 of 20)



File View Services Help

Event 1 of 3: Call (DSA Address 20594030) Line 68 Col 1 8
Command ==> Scroll ==> PAG

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:0

R10: 00007124 (Module SAM1 program SAM1 + X'124')
R11: 00007798 (Module SAM1 program SAM1 + X'798')
R12: 000070FC (Module SAM1 program SAM1 + X'FC')
R13: 20594030 (593872 bytes of storage addressable)
R14: 80007D32 (Module SAM1 program SAM1 + X'D32', source line # 312)
R15: A05142B0 (Module IGZCPAC + X'2B0')

Associated Open Files

File Name CUSTFILE
File Name : CUSTRPT
File Name : TRANFILE

Associated Storage Areas

Next Event Details

*** Bottom of data.

Cursor select the file name

Enter

Debug Clues:

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

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

...returns to the detail report for program SAM1. Here is a list of files that were open at the time of the abend. You can tab down to the point-and-shoot field for CUSTFILE. Enter.

Analyze an abend (19 of 20)

IBM

The screenshot displays the IBM Fault Analyzer interface. At the top, the title is "Analyze an abend (19 of 20)" and the IBM logo is in the upper right. The main window is titled "File View Services Help" and shows "File Information" for a file named "CUSTFILE". A yellow callout box asks "What is the full name of the CUSTFILE file?". The "Data Set Name" field contains "DNET845.ADLAB.FILES(CUST2FA)", which is circled in red. Another yellow callout box states "CUSTFILE is an input file". The "Open Status" is "INPUT". Below this is a hex dump of the previous record, with a yellow callout box containing "F8" pointing to the hex value "F0F0F0F4". To the right, a "Debug Clues" panel lists several clues with checkmarks, including "Abended in program SAM2 because:", "CUST-ACCT-BALANCE has bad numeric data, which was passed from a calling program", "SAM1 called SAM2", "SAM1 passed bad data in CUST-REC", and "The bad data was read from file CUSTFILE". The bottom of the window shows the page number "22", the text "IBM Fault Analyzer for z/OS - V12 Tutorial", and the copyright "© 2012 IBM Corporation".

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

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

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

Current Record. . . . . : Record data length 80
Address  Offset  Hex                               EBCDIC
00023F80          F0F0F0F4 F0F0F0F4 F0F0F0F4 F0F0F0F4 * * * *
00023F88      +10  40404040 40404040 40404040 40404040 * * * *
00023F98      +20  40404040 40404040 40404040 40404040 * * * *
00023FA8      +30  40404040 40404040 40404040 40404040 * * * *
00023FB8      +40  40404040 40404040 40404040 40404040 * * * *
```

Debug Clues:

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

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

And the File Information panel for CUSTFILE is displayed. Here is the last question: "What is the file name for the CUSTFILE DD?".

The file name is shown in the data set name field, and it is DNET845.ADLAB.FILES(CUST2FA). Notice that the file is opened for input only, and therefore it is likely that data in the record buffer actually came from the file. Scrolling down, F8.

Analyze an abend (20 of 20)



File View Services Help

File Information Line 18 Col 1
Command ==> Scroll ==> HA

JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:00

Current Record Record data length 80

Address	Offset	Hex	EBCDIC
00023F88		F5F4F3F2 F1C34040 40404040 40C1A2A3	*54321C Ast*
00023F98	+10	85006B40 C485A940 40404040 40407C7B	*er, Dez @H*
00023FA8	+20	5B6C5010 02E2A396 9994A840 C6819393	*\$%&. Stormy Fall*
00023FB8	+30	A2404040 C481A381 40C595A3 99A840D6	*s Data Entry 0*
00023FC8	+40	97859981 A3969940 40404040 40404040	*perator *

Next Record Record data length 80

Address	Offset	Hex	EBCDIC
00023FD8		F5F5F5F5 F5C34040 40404040 40C485D4	*55555C DeM*
00023FE8	+10	8195956B 40C8A487 88404040 40400001	*ann, Hugh ..*
00023FF8	+20	23400C00 03C68189 99A58985 A6404040	*. . .Fairview *
00024008	+30	40404040 D496A389 A581A389 96958193	* Motivational*
00024018	+40	40E29785 81928599 40404040 40404040	* Speaker *

Associated File Control Blocks

*** Bottom of data.

Here is the bad data in the record

Debug Clues:

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

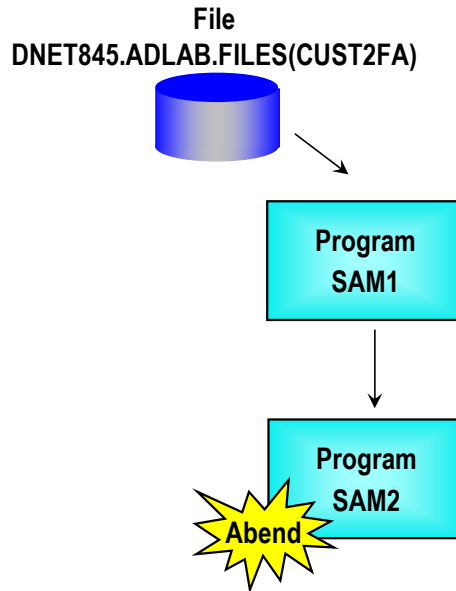
Here is the current record in the file, and sure enough, the bad data is there.

At this point, enough information has been gathered to develop a complete theory for understanding the root cause of this abend.

Analyze an abend ... what happened:



1. SAM1 read a record from an input file that contained bad data in the CUST-ACCT-BALANCE field
2. Program SAM1 called SAM2, passing the bad data
3. SAM2 abended when it tried to calculate using the bad data



24

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

SAM1 is the main program. It read a record from a file. The record contained bad data in the CUST-ACCT-BALANCE field. SAM1 passed the record area with the bad data to a subprogram named SAM2. Finally, SAM2 tried to do arithmetic with the bad data, and that caused the abend.

The example you just saw is not atypical of the process you might work through to troubleshoot an abend. You will look at the symptoms, develop theories about what may have caused the problem, and examine the available information to test those theories.

- 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 program source information
- Automating source information search with options
- Re-creating source information for existing modules

Fault Analyzer can be set up to interface with IBM File Manager for z/OS. Next you will see how to access File Manager to browse or edit files.

Fix the cause of the abend



File View

Fault Analyzer can interface with File Manager to edited the file to fix the bad data

```
File Information Line 18 Col 1 80
Command ==> Scroll ==> HALF
JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:02
Current Record . . . . . : Record data length 80
Address Offset Hex EBCDIC
00023F88 F5F4F3F2 F1C34040 40404040 40C1A2A3 *54321C Ast*
00023F98 +10 85996B40 C485A940 40404040 40407C7B *er, Dez @#*
00023FA8 +20 5B6C5000 02E2A396 9994A840 C6819393 *$%&..Stormy Fall*
00023FB8 +30 A2404040 C481A381 40C595A3 99A840D6 *s Data Entry 0*
00023FC8 +40 97859981 A3969940 40404040 40404040 *perator *
```

Next Record : Record data length 80

```
Address Offset Hex EBCDIC
00023FD8 F5F5F5F5 F5C34040 40404040 40C485D4 *55555C DeM*
00023FE8 +10 8195956B 40C8A487 88404040 40400001 *ann, Hugh ..*
00023FF8 +20 23400C00 03C68189 99A58985 A6404040 *. ...Fairview *
00024008 +30 40404040 D496A389 A581A389 96958193 * Motivational*
00024018 +40 40E29785 81928599 40404040 40404040 * Speaker *
```

Associated File Control Blocks

*** Bottom of data.

F7

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

To use this interface, IBM File Manager for z/OS has to be installed on your system, and your systems programmer must have enabled the interface between Fault Analyzer and File Manager. Scroll to the top of the File Information section with F7.

Cursor select the data set name



```
File View Services Help
File Information Line 1 Col 1 80
Command ==> Scroll ==> PAGE
JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:02

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

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

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

Select the File

Enter

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

To edit the file and look at the bad data, you have to have authorization to the file. And of course to fix the bad data, it still has to be in the file. But if those are not problems then you can browse or edit the file. Tab to the file name, and press enter.

Select edit



File View Services Help Data Set Actions

Data Set . . . : DNET845.ADLAB.FILES(CUST2FA)

Select one of the following options and press Enter:

- 1 1. Edit
- 2. Browse

Use template currently associated with this dataset y (Y/N)

This will open File Manager in Edit or Browse mode

Previous Record . . . : Record data length 80

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

Current Record . . . : Record data length 80

Address	Offset	Hex	EBCDIC
---------	--------	-----	--------

Enter

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

A panel is displayed, where you can choose browse or edit. Option one for edit is entered.

File Manager edit session



The **FE** (Fields in Error) command will search for bad data

```
Process Options Help
Edit DNET845.ADLAB.FILES(CUST2FA) Top of 61
Command ==> fe #7 Scroll CSR
Record AT TOP Format TABL
CUST-ID RECORD-TYPE FILLER NAME ACCT-BALANCE ORDERS-YTD
#3 #4 #5 #6 #7 #8
AN 1:5 AN 6:1 AN 7:7 AN 14:17 PD 31:5 BI 36:2
<---> - <---+> <-----1-----> <---+---1> <--->
***** **** Top of data ****
000001 01001 C Lynn, Amanda 67.68 9
000002 02200 C Graham, Anna 610.05 10
000003 02202 C Major, Art 1234.56 5
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000005 03003 C Prentice, Anna 396.63 7
- - - - - PRODUCT-RECORD - - - - - 2 Line(s) suppressed
000008 03390 C Deeds, Darren 74.00 3
000009 05500 C Parker, Ford 233.27 5
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
ly, Brighton 311.08 10
- - - - - 1 Line(s) suppressed
der, Annette 489.84
- - - - - 1 Line(s) suppressed
000015 06711 C Dubree, Dustin 192.98
```

File Manager tip: **TV** (template view) command can be used to specify a copybook or template to format records

Enter

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

The file manager editor is displayed. The abend was caused because of bad data in a record in the ACCT-BALANCE field. In the File Manager editor, there is an easy way to locate bad data in numeric fields: the FE command, which stands for Fields in Error. It scans records looking at numeric fields and stops at the next one that has bad data.

Fix the bad data



Process Options Help

Edit DNET845.ADLAB.FILES (CUST2FA) 1 error(s) found
Command ==> _____ Scroll CSR
Record 60 Format TABL

CUST-ID	RECORD-TYPE	FILLER	NAME	ACCT-BALANCE	ORDERS-YTD
#3	#4	#5	#6	#7	#8
AN 1:5	AN 6:1	AN 7:7	AN 14:17	PD 31:5	BI 36:2
<--->	-	<---+>	<-----1----->	<-----1>	<--->
000060	54321	C	Aster, Dez	*****	2
000061	55555	C	DeMann, Hugh	1234.00	3
*****	****	End of data	****		

Fix the bad data by overtyping it's value

1234.56****
1234.00

Enter

30

IBM Fault Analyzer for z/OS - V12 Tutorial

© 2012 IBM Corporation

To fix the bad data, overtype the field with a valid value.

Save the change



```
Process  Options  Help
-----
Edit          DNET845.ADLAB.FILES (CUST2FA)          Rec 60 of 61
Command ==>                                     Scroll CSR
                                                Record 60   Format TABL
CUST-ID RECORD-TYPE FILLER  NAME          ACCT-BALANCE ORDERS-YTD
#3      #4          #5      #6              #7          #8
AN 1:5  AN 6:1     AN 7:7  AN 14:17      PD 31:5     BI 36:2
<--->  -          <---+> <-----1-----> <---+---1>   <--->
000060 54321      C              Aster, Dez      1234.56      2
000061 55555      C              DeMann, Hugh   1234.00      3
***** **** End of data ****
```



Press F3 to save the change and exit.

File Manager returns to Fault Analyzer



```
File View Services Help
File Information Line 1 Col 1 80
Command ==> Scroll ==> PAGE
JOBNAME: DNET845X SYSTEM ABEND: 0C7 DEMOMVS 2010/02/23 15:45:02

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

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

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

That returns to Fault Analyzer. At this point, the cause of the abend has been fixed. Of course, it is a good programming practice to perform numeric validity checks on data that comes from an external source. Because the error was pinpointed, you know one of the places where a numeric check is needed in the application.

That is the end of this section, an example of using interactive reanalysis

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_FAv12s06InterReanalysis.ppt

This module is also available in PDF format at: [../FAv12s06InterReanalysis.pdf](http://FAv12s06InterReanalysis.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, 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.