

Abend Aid MVS
Musterlösungen zu den Übungen

4. Februar 2011

Eine Ausarbeitung von:

cps4it

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen
Fon: 06721-992611 • Fax: 06721-992613 • Mail: ralf.seidler@cps4it.de
Internet : <http://www.cps4it.de>
Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

Inhaltsverzeichnis

1	ALLGEMEINES	3
	ZUM SCHMUNZELN	3
	VORAUSSETZUNGEN UND REGELN	3
2	S0C7 – 1	4
	AUFGABE	4
	DUMP	4
	MUSTERLÖSUNG	4
3	S0C7 – 2	7
	AUFGABE	7
	DUMP	7
	MUSTERLÖSUNG	7
4	S0C7 – 3	10
	AUFGABE	10
	DUMP	10
	MUSTERLÖSUNG	10
5	S0C4 – 1	14
	AUFGABE	14
	DUMP	14
	MUSTERLÖSUNG	14
6	S0C4 – 2	17
	AUFGABE	17
	DUMP	17
	MUSTERLÖSUNG	17
7	S806	20
	AUFGABE	20
	DUMP	20
	MUSTERLÖSUNG	20
8	S0CB – 1	22
	AUFGABE	22
	DUMP	22
	MUSTERLÖSUNG	22
9	S0CB – 2	24
	AUFGABE	24
	DUMP	24
	MUSTERLÖSUNG	24
10	S0CB – 3	26
	AUFGABE	26
	DUMP	26
	ANMERKUNGEN	26
	ERLÄUTERUNGEN ZUM PROGRAMM TES46	26
	MUSTERLÖSUNG	27

1 Allgemeines

Zum Schmunzeln

"Glück ist: zu begreifen, wie alles zusammenhängt."

Sten Nadolny (*1942), dt. Schriftsteller

Compileliste TES39: u-id.DUMP.COMPILE.TES39
Compileliste TES46: u-id.DUMP.COMPILE.TES46
Compileliste TES47: u-id.DUMP.COMPILE.TES47

Voraussetzungen und Regeln

- Ruhe für die Suche nach der Abbruchursache ist notwendig im wahrsten Sinn des Wortes. Insbesondere sind wichtig:
 - Telefon umleiten oder abstellen
 - mit Führungskräften Zeitpunkte vereinbaren, wann ein Status gegeben wird (und Chef/in aus dem Zimmer werfen) ;-)
- Umwandlungsdatum für jedes zu analysierende Programm **immer** kontrollieren. In der Testumgebung ist sogar oft die Umwandlungsuhrzeit wichtig. Dieser Punkt ist in den Musterlösungen nicht erwähnt.
- Die Informationen von Abend Aid aufmerksam lesen. Jeder Hinweis kann interessant sein und den nachfolgenden Aufwand drastisch reduzieren.
- Jede gefundene Information prüfen, ob sie plausibel ist. Es kann beispielsweise niemals ein „SET ... TO TRUE“ oder ein „CALL“ zu einem SOC7 führen.
- Alle wesentlichen Schritte und Informationen notieren.

2 S0C7 – 1

Aufgabe

Suchen des abgebrochenen Befehls und der Inhalte aller betroffenen Felder

Dump

u-id. DUMP.JOBLOG.S0C7#01.txt

Musterlösung

wichtigste Informationen aus Dump:

A data exception occurred during execution of program TES47. The expected completion code is S0C7.

```
*****  
*           Analysis of Error           *  
*****
```

A Data Exception was caused by data referenced at displacement 032 from the start of TGT BLL cell 01 . The field contains X'FOF0F0F0F0F04C4C'. Refer to the data division map in the program listing to locate the field name.

The field causing the exception is located in a temporary work field in the DSA. The actual field in error is in the linkage section of program TES47.

```
*****  
*           Error Location           *  
*****
```

The next sequential instruction to be executed in program TES47 was at displacement 00000714.

suchen Abbruchstelle in Compileliste (von TES47)

find 0714 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0714

→ 000102 0006FA COMPUTE (also Zeile 102 beinhaltet den Befehl)

→ 000102 00101 COMPUTE FELD-ERGEBNIS-3 = FELD-5 / FELD-6

CL**5 37 59 61

An welcher Stelle sind die Felder im Dump zu suchen ...

000037	00036	10 FELD-ERGEBNIS-3	PIC S9(009) BINARY.	CL*31 BLW=0000+044 ,0000044 4C
000059	00058	10 FELD-5	PIC 9(009).	CL**2 BLL=00001+032 ,0000032 9C
000061	00060	10 FELD-6	PIC 9(009).	CL**2 BLL=00001+03C ,000003C 9C

Suchen im Dump nach den Inhalten der Felder

```
*****
*           Data Division of program TES47           *
*           Compiled on 22 JUN 2009   at 15:24:14   *
*****
```

...

Working-Storage Section

Working storage referenced by TGT **BLW cell** 0

```
Address  Offset  ----- Data -----
35E110F8 +00000  E3C5E2F4 F7404040 E3C5E2F6 F6404040  TES47  TES66
35E11108 +00010  00000000 00000000 00000000 00000000  .....
          LINES 35E11118-35E11118 SAME AS ABOVE
35E11128 +00030  00000000 F0F0F0F0 F0F0F0F3 C3000000  ....00000003C...
35E11138 +00040  00088C00 00000000 00000000 00000000  .....
35E11148 +00050  00000000 00000000 00000000 00000000  .....
          LINES 35E11158-35E111A8 SAME AS ABOVE
35E111B8 +000C0  00000000 0000
```

Linkage Section

Abend Aid MVS

Linkage storage referenced by TGT **BLL cell** 1

```
Address  Offset  ----- Data -----  
35E11058 +00000  F0F0F0F0 F0F0F0F0 F14FF0F0 F0F0F0F0 000000001!000000  
35E11068 +00010  F0F1F14F F0F0F0F0 F0F0F0F2 F24FF0F0 011!000000022!00  
35E11078 +00020  F0F0F0F0 F0F0F24F F0F0F0F0 F0F0F0F4 0000002!00000004  
35E11088 +00030  F44F0F0 F0F0F0F0 4C4C4C4F F0F0F0F0 4!000000<<<!0000  
35E11098 +00040  F0F0F0F6 F64FF0F0 40404040 40404040 00066!00  
35E110A8 +00050  00000000 00000000 00000000 00000000 .....
```

Feld FELD-ERGEBNIS-3 beinhaltet also Low-Values (klar, weil es das Zielfeld ist und vorher noch nicht benutzt wurde)

Feld FELD-5 beinhaltet 000000<<<

Feld FELD-5 beinhaltet 000000066

Den Inhalt des Feldes FELD-5 hat AbendAid schon als fehlerhaft angezeigt („wichtigste Informationen aus Dump“).

3 S0C7 – 2

Aufgabe

Suchen des abgebrochenen Befehls und der Inhalte aller betroffenen Felder

Dump

u-id. DUMP.JOBLOG.S0C7#03.txt

Musterlösung

wichtigste Informationen aus Dump:

```
*****  
*           Analysis of Error           *  
*****
```

A data exception occurred during execution of program TES47. The expected completion code is S0C7.

A Data Exception was caused by data referenced at displacement 00A from the start of TGT BLL cell 01 . The field contains X'F0F0F05BF0F0F0F0'. Refer to the data division map in the program listing to locate the field name.

```
*****  
*           Error Location           *  
*****
```

The next sequential instruction to be executed in program TES47 was at displacement 000006B4.

suchen Abbruchstelle in Compileliste (von TES47)

find 06B4 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 06B4

→ 000098 00069A COMPUTE (also Zeile 98 beinhaltet den Befehl)

→ 000098 00097 COMPUTE FELD-ERGEBNIS-1 = FELD-1 + FELD-2

An welcher Stelle sind die Felder im Dump zu suchen ...

```
000051 00050 10 FELD-1 PIC 9(009). CL**2 BLL=00001+00A,000000A 9C
000053 00052 10 FELD-2 PIC 9(009). CL**2 BLL=00001+014,0000014 9C
000033 00032 10 FELD-ERGEBNIS-1 PIC S9(009). CL*30 BLW=00000+034,0000034 9C
```

Suchen im Dump nach den Inhalten der Felder

```
*****
*           Data Division of program TES47           *
*           Compiled on 23 JUN 2009 at 15:04:40      *
...
Working storage referenced by TGT BLW cell 0

Address Offset ----- Data -----
360110F8 +00000 E3C5E2F4 F7404040 E3C5E2F6 F6404040 TES47 TES66
36011108 +00010 00000000 00000000 00000000 00000000 .....
LINES 36011118-360111A8 SAME AS ABOVE
360111B8 +000C0 00000000 0000 .....
...
Linkage Section

Linkage storage referenced by TGT BLL cell 1
...
Address Offset ----- Data -----
36011058 +00000 F0F0F0F0 F0F0F0F0 F14F0F0 F05BF0F0 000000001!000$00
36011068 +00010 F0F0F04F F0F0F0F0 F0F0F0F2 F24FF0F0 000!000000022!00
36011078 +00020 F0F0F0F0 F0F0F24F F0F0F0F0 F0F0F0F4 0000002!00000004
```


Abend Aid MVS

```
36011088 +00030 F44FF0F0 F0F0F0F0 4C4C4C4F F0F0F0F0 4!000000<<<!0000  
36011098 +00040 F0F0F0F6 F64FF0F0 40404040 40404040 00066!00
```

Feld FELD-ERGEBNIS-1 beinhaltet also Low-Values (klar, weil es das Zielfeld ist)

Feld FELD-1 beinhaltet 000\$00000

Feld FELD-2 beinhaltet 000000022

Den Inhalt des Feldes FELD-1 hat AbendAid schon als fehlerhaft angezeigt („wichtigste Informationen aus Dump“).

4 S0C7 – 3

Aufgabe

Suchen des abgebrochenen Befehls und der Inhalte aller betroffenen Felder
Anschließend: „Umfeld“ des abgebrochenen Statements und der betroffenen Daten ansehen; es gibt weitere „Fehler“ in den Daten.

Dump

u-id. DUMP.JOBLOG.S0C7#02.txt

Musterlösung

wichtigste Informationen aus Dump:

```
*****  
*           Analysis of Error           *  
*****
```

```
A Data Exception was caused by data referenced at displacement 032 from  
the start of TGT BLL cell 01 . The field contains  
X'FOF0F0F0F0F04C4C'. Refer to the data division map in the program  
listing to locate the field name.
```

```
The field causing the exception is located in a temporary work field in  
the DSA. The actual field in error is in the linkage section of program  
TES47.
```

...

```
*****  
*           Error Location           *  
*****
```

```
The next sequential instruction to be executed in program TES47 was at  
displacement 00000714.
```

suchen Abbruchstelle in Compileliste (von TES47)

find 0714 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0714

→ 000102 0006FA COMPUTE 000103 000722 DISPLAY

→ also Zeile 102 beinhaltet den Befehl

→ 000102 00101 COMPUTE FELD-ERGBNIS-3 = FELD-5 / FELD-6

An welcher Stelle sind die Felder im Dump zu suchen ...

000059	00058	10 FELD-5	PIC 9(009).	CL**2 BLL=00001+032 ,0000032 9C
000061	00060	10 FELD-6	PIC 9(009).	CL**2 BLL=00001+03C ,000003C 9C
000037	00036	10 FELD-ERGBNIS-3	PIC S9(009) BINARY.	CL*31 BLW=00000+044 ,0000044 4C

Suchen im Dump nach den Inhalten der Felder

```
*****  
*                    Data Division of program TES47                    *  
*                    Compiled on 23 JUN 2009                    at 15:04:40                    *  
*                    Working-Storage Section
```

Working storage referenced by TGT BLW cell 0

...

Address	Offset	-----	Data	-----	
360110F8	+00000	E3C5E2F4	F7404040	E3C5E2F6	F6404040 TES47 TES66
36011108	+00010	00000000	00000000	00000000	00000000
		LINES 36011118-36011118 SAME AS ABOVE			
36011128	+00030	00000000	F8F5F9F1	F5F5F9F7	C4000000 85915597D...
36011138	+00040	00088C00	00000000	00000000	00000000
36011148	+00050	00000000	00000000	00000000	00000000
		LINES 36011158-360111A8 SAME AS ABOVE			
360111B8	+000C0	00000000	0000	

...

Abend Aid MVS

Linkage Section

Linkage storage referenced by TGT BLL cell 1

```
Address  Offset  -----  Data -----
36011058 +00000 F0F0F0F0 F0F0F0F0 F14FC8C5 C9A1C5C5 000000001!HEI.EE
36011068 +00010 C9D5E24F F0F0F0F0 F0F0F0F2 F24FF0F0 INS!000000022!00
36011078 +00020 F0F0F0F0 F0F0F24F F0F0F0F0 F0F0F0F4 0000002!00000004
36011088 +00030 F44FF0F0 F0F0F0F0 4C4C4C4F F0F0F0F0 4!000000<<<!0000
36011098 +00040 F0F0F0F6 F64FF0F0 40404040 40404040 00066!00
```

Feld FELD-ERGBNIS-3 beinhaltet also Low-Values (klar, weil es das Zielfeld ist); es ist binär und 4 Bytes lang.

Feld FELD-5 beinhaltet 000000<<<

Feld FELD-6 beinhaltet 000000066

Den Inhalt des Feldes FELD-5 hat AbendAid schon als fehlerhaft angezeigt („wichtigste Informationen aus Dump“).

Der erste Teil der Aufgabe ist also gelöst.

Bei näherem Hinsehen auf die Dump-Inhalte fällt auf, dass da noch ein Inhalt **HEI.EEINS** vorhanden ist. Die Adresse, bei der dieser Inhalt anfängt, ist BLL Zelle 1 Offset 00A. Wenn wir uns an die erste Aufgabe erinnern, müsste da doch das Feld FELD-1 stehen. Dies verifizieren wir.

Extrakt aus Compile-Liste:

```
000051          00050          10 FELD-1          PIC  9(009).          CL**2  BLL=00001+00A,000000A 9C
```

Es stimmt also, das auf dieser Adresse FELD-1 zu finden ist. Schauen wir noch mal in den Programmcode:

```
000098          00097          COMPUTE FELD-ERGBNIS-1 = FELD-1 + FELD-2          CL**5 33 51 53
000099          00098          DISPLAY '* ERGBNIS ADDITION          : ' FELD-ERGBNIS-1          CL**5 33
000100          00099          COMPUTE FELD-ERGBNIS-2 = FELD-3 * FELD-4          CL**5 35 55 57
000101          00100          DISPLAY '* ERGBNIS MULTIPLIKATION : ' FELD-ERGBNIS-2          CL**5 35
000102          00101          COMPUTE FELD-ERGBNIS-3 = FELD-5 / FELD-6          CL**5 37 59 61
000103          00102          DISPLAY '* ERGBNIS DIVISION          : ' FELD-ERGBNIS-3          CL**5 37
000104          00103          COMPUTE FELD-ERGBNIS-4 = FELD-6 / FELD-7          CL*31 39 61 63
```

Da erkennen wir, dass der Compute mit FELD-1 **vor** dem Compute mit FELD-6 steht. Beide Felder beinhalten ungültige numerische Werte. Aber warum kommt es beim ersten Compute nicht zu einem S0C7?

Schauen wir uns den Inhalt von FELD-1 noch einmal an. Wir schreiben es in der Form, wie es im ISPF-Edit sichtbar ist:

```
CCCACCCDE
```

```
859155952
```

```
HEI.EEINS
```

H ist also C8, E ist C5 und so weiter.

Der COBOL-Compiler geht bei einer numerischen Operation mit PIC n(9)- oder PIC Sn(9)-Feldern so vor, dass er Rechenfelder zunächst packt (in das Format PACKED DECIMAL / COMP-3 umformatiert) und dann die Rechenoperation ausführt. Danach macht er – wenn erforderlich – die Konvertierung in das Format des Zielfeldes. Auf unseren Fall angewendet heißt das, dass die Inhalte von FELD-1 und FELD-2 zu packen sind. Wie geht die Pack-Operation vor sich? Der Compiler nimmt jeweils das zweite Halbbyte und überträgt dies in sein temporäres Hilfsfeld. Er schreibt also in das Zielfeld die Halbbytes 859155952 und setzt in das letzte Halbbyte noch ein Vorzeichen, das C. Dieses Feld ist **sauber** numerisch. Es kann also bei dieser Operation nicht zu einem S0C7 kommen. Diese Vorgehensweise wählt er, um einen zur Laufzeit möglichst schnellen Code zu erzeugen. Wären in irgendeinem der „unteren“ Halbbytes keine Ziffer, sondern A oder B,C,D,E, F, würde es zu einem S0C7 kommen. Wenn wir uns den Inhalt von FELD-5 ansehen, haben wir auch 3 Mal ein C im zweiten Halbbyte.

5 S0C4 – 1

Aufgabe

Suchen nach der Abbruchursache.

Dump

uid.DUMP.JOBLOG.S0C4#01.txt

Musterlösung

wichtigste Informationen aus Dump:

Es gibt keinen Dump (DD-Name ABENDAID) !!! Da wurde so viel zerschossen, dass AbendAid keine Chance mehr hatte. Hinweis darauf:

```
1          J E S 2  J O B  L O G  --  S Y S T E M  E N T W  --  N O D E  N T Z D
0
16.51.24 JOB18154 ---- MONDAY,    24 JAN 2011 ----
16.51.24 JOB18154 IRR010I  USERID VKA2      IS ASSIGNED TO THIS JOB.
16.51.37 JOB18154 ICH70001I VKA2      LAST ACCESS AT 16:48:30 ON MONDAY, JANUARY 24, 2011
16.51.37 JOB18154 $HASP373 VKA2002  STARTED - WLM INIT  - SRVCLASS BATWLM_M - SYS ENTW
16.51.37 JOB18154 IEF403I  VKA2002 -  STARTED - TIME=16.51.37
16.51.37 JOB18154 +AB5A8- ABEND-AID ESPIE EXIT WAS INVOKED
```

Das heißt: AbendAid hatte keine Chance mehr!

Weitersuchen:

Informationen aus SYSOUT:

CEE3204S The system detected a protection exception (System Completion Code=0C4).
From compile unit TES39 at entry point TES39 at compile unit offset +00000560 at entry offset +00000560 at address 35B01180.

Das merken wir uns.

Informationen aus CEEDUMP:

Traceback:

DSA	Entry	E Offset	Statement	Load Mod	Program Unit	Service	Status
1	CEEHDSP	+0000420E		CEEPLPKA	CEEHDSP	UK61610	Call
2	IGZCDSP	+000008A2		IGZCPAC	IGZCDSP		Exception
3		+00000000					Call
4	IGZCFCC	+000002BE		IGZCPAC	IGZCFCC	UK49571	Call
5	TES39	+00000560		TES39	TES39		Call

DSA	DSA Addr	E Addr	PU Addr	PU Offset	Comp Date	Compile Attributes
1	0001DAE0	10D57A18	10D57A18	+0000420E	20101027	CEL
2	0001D510	10C60B80	10C60B80	+000008A2	20090317	LIBRARY
3	0001D378	00000000	00000000	+00000000	*****	COBOL
4	0001D198	10C61A40	10C61A40	+000002BE	20090825	LIBRARY
5	0001D018	35B00C20	35B00C20	+00000560	20110124	COBOL

Was ist da nur los?

Da scheint ein Call gewesen zu sein und irgendwo hat es da geknallt.

Hinweis: Die Adresse im SYSOUT ist die gleiche wie beim TRACEBACK (+00000560). Was ist das?

suchen Abbruchstelle in Compileliste (von TES39)

find 0560 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0560

```
→ 000077 000526 CALL                                000081 00056A DISPLAY
→ 000077      1 00076                                CALL TES47 USING EINGABE-ZEILE                CL*68 23 24
```

Es ist also tatsächlich ein Call und zwar auf TES47.

Zwischenergebnis:

Es sieht also so aus, als ob in TES47 der Speicher zerschossen wurde.

Also: Umwandlung mit der Compileoption SSRANGE und erneuter SUBMIT liefert, ...

Ergebnis:

... dass es ein Tabellenüberlauf war. Man schaue sich die DISPLAYs an. ;-

6 S0C4 – 2

Aufgabe

Suchen nach der Abbruchursache.

Dump

uid.DUMP.JOBLOG.S0C4#02.txt

Musterlösung

wichtigste Informationen aus Dump:

```
A storage reference exception occurred during execution of program TES47
. The expected completion code is S0C4.
```

```
*****
*           Analysis of Error           *
*****
```

```
The system completion code of 0C4 is issued when the program attempts to
use storage that is not accessible.
```

```
*** List of Probable 0C4 Causes ***
```

etc.

```
*****
*           Error Location           *
*****
```

```
The next sequential instruction to be executed in program TES47 was at
displacement 00000966.
```

suchen Abbruchstelle in Compileliste (von TES47)

find 0966 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0966

→ 000179 000954 MOVE (also Zeile 179 beinhaltet den Befehl)

→ 000179 00178 MOVE SPACES TO LINKAGE-ZUSATZ

An welcher Stelle sind die Felder im Dump zu suchen ...

```
000065            00064 01       LINKAGE-ZUSATZ                    PIC X(5000) .                    CL*32 BLL=00002+000                    5000C
```

Das Feld ist definiert.

```
000068            00067    PROCEDURE DIVISION USING EINGABE-ZEILE LINKAGE-ZUSATZ .
```

Es ist in der USING Leiste.

Die Suche, ob mit dem Feld noch etwas getan wird liefert kein Ergebnis.

Wir müssen also den CALL vom Aufrufer prüfen.

wichtigste Informationen aus Dump (2):

```
*****
*           Call Trace Summary           *
*****

*****Calling*****  ****Return****      *****Called*****
Load-Mod  Program      Type      Value      Program      Load-Mod

*SYSTEM                                     Links to TES39      TES39

TES39      TES39                                     Calls              CEEBINIT

CEEBINIT                                     DISP  00CBD948  Calls      TES39      TES39

TES39      TES39      DISP  00000562  Calls              IGZCPAC

IGZCPAC                                     DISP  110C74B4  Calls      TES47      TES47      *
```

Der Call ist auf Displacement 000562 zu suchen.

suchen CALL in Compileliste (von TES39)

find 0526 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0526

→ 000077 000526 CALL (also Zeile 77 beinhaltet den Befehl)

→ 000077 1 00076 CALL TES47 USING EINGABE-ZEILE

Using-Leiste stimmt nicht überein.

7 S806

Aufgabe

Suchen nach der Abbruchursache.

Dump

uid.DUMP.JOBLOG.S806.txt

Musterlösung

wichtigste Informationen aus Dump:

```
*****  
*           Analysis of Error           *  
*****
```

Make sure the program name specified is correct. Also, make sure the proper library is specified in a job library (JOBLIB/STEPLIB) or link library.

The IBM message that corresponds to the condition is:

```
CEE3501S The module TES66 was not found.
```

Das sieht man auch im SYSOUT:

```
CEE3501S The module TES66      was not found.  
      From compile unit TES47 at entry point TES47 at compile unit offset +00000932 at entry offset +00000932 at  
      address 3601D0F2.
```

suchen Abbruchstelle in Compileliste (von TES47)

find 0932 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0937

Oder. Einfach nach „TES66“ in der Compileliste suchen.

Weitere Analysen sollten nicht notwendig sein.

8 S0CB – 1

Aufgabe

Suchen nach der Abbruchursache.

Dump

uid.DUMP.JOBLOG.S0CB#01.txt

Musterlösung

wichtigste Informationen aus Dump:

```
*****  
*           Analysis of Error           *  
*****
```

A Decimal Divide Exception was caused by data referenced at displacement 046 from the start of TGT BLL cell 01 . The field contains X'FOFOFOFOFOFOFOFO'. Refer to the data division map in the program listing to locate the field name.

The field causing the exception is located in a temporary work field in the DSA. The actual field in error is in the linkage section of program TES47.

A decimal divide exception is caused when the divisor is zero.

```
*****  
*           Error Location           *  
*****
```

The next sequential instruction to be executed in program **TES47** was at displacement **0000746**.

suchen Abbruchstelle in Compileliste (von TES47)

find 0746 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0746

→ 000104 00072C COMPUTE (also Zeile 104 beinhaltet den Befehl)

→ 000104 00103 COMPUTE FELD-ERGBNIS-4 = FELD-6 / FELD-7

An welcher Stelle sind die Felder im Dump zu suchen ...

000039	00038	10 FELD-ERGBNIS-4	PIC S9(009) BINARY.	CL*31 <u>BLW=0000+049</u> ,0000049 4C
000061	00060	10 FELD-6	PIC 9(009).	CL**2 <u>BLL=00001+03C</u> ,000003C 9C
000063	00062	10 FELD-7	PIC 9(009).	CL*31 <u>BLL=00001+046</u> ,0000046 9C

Suchen im Dump nach den Inhalten der Felder

```
*****
*           Data Division of program TES47           *
*           Compiled on 22 JUN 2009   at 15:24:14   *
*****

Working storage referenced by TGT BLW cell      0

35E11138 +00040 00088C00 00000000 00000000 00000000 .....

Linkage storage referenced by TGT BLL cell      1

35E11088 +00030 F44FF0F0 F0F0F0F0 F0F0F54F F0F0F0F0 4!0000000005!0000
35E11098 +00040 F0F0F0F6 F64FF0F0 F0F0F0F0 F0F0F040 00066!00000000
```

Nun muss danach geforscht werden, wer diesen Inhalt liefert. Erster Hinweis: FELD-7 steht in der Linkage. Bei der weiteren Analyse des Programms TES39 findet man, dass das Feld FELD-7 direkt von den gelesenen Eingabedaten stammt.

9 S0CB – 2

Aufgabe

Suchen nach der Abbruchursache.

Dump

uid.DUMP.JOBLOG.S0CB#02.txt

Musterlösung

wichtigste Informationen aus Dump:

A decimal-divide exception occurred during execution of program TES39.
The expected completion code is S0CB.

```
*****  
*           Analysis of Error           *  
*****
```

A Decimal Divide Exception was caused by data referenced at displacement 02C from the start of TGT BLW cell 00 . The field contains X'FOFOFOFOFOFOFOFO' . Refer to the data division map in the program listing to locate the field name.

The field causing the exception is located in a temporary work field in the DSA. The actual field in error is in Working-Storage of program TES39.

```
*****  
*           Error Location           *  
*****
```

The next sequential instruction to be executed in program TES39 was at displacement 00000518.

suchen Abbruchstelle in Compileliste (von TES39)

find 0518 (wenn mit LIST umgewandelt)

→ Befehl wird angezeigt

find hexloc (wenn mit NOLIST umgewandelt)

suchen Befehl mit Offset <= 0518

→ 000075 0004FE COMPUTE (also Zeile 75 beinhaltet den Befehl)

→ 000075 1 00074 COMPUTE FELD-3 = FELD-1 / FELD-2 CL*74 33 29 31

Aber: Schauen wir den Code mal genauer an – TES39:

```
000069 00068 WHEN I1-MAX-EINGABE = 6 CL*74 25
000070 00069 * FELD-1 NACH FELD-2 UEBERTRAGEN UND ZURUECK GEBEN LASSEN CL*74
000071 1 00070 MOVE 1 TO FELD-1 CL*74 29
000072 1 00071 MOVE 0 TO FELD-2 CL*74 31
000073 1 00072 CALL TES47 USING BY CONTENT EINGABE-ZEILE CL*74 23 24
000074 1 00073 END-CALL CL*74
000075 1 00074 COMPUTE FELD-3 = FELD-1 / FELD-2 CL*74 33 29 31
```

Und in TES47:

```
000077 00076 EVALUATE (I1-MAX-EINGABE) CL**6 49
...
000083 1 00082 WHEN (6) PERFORM FKT-6 CL*34 187
...
000187 00186 FKT-6 SECTION. CL*34
000188 00187 * CL*34
000189 00188 MOVE FELD-1 TO FELD-2 CL*34 51 53
```

Da müsste doch in FELD-2 der Wert von FELD-1 stehen. Das war die Absicht des Programmierers.

Er hat aber den CALL falsch geschrieben:

```
CALL TES47 USING BY CONTENT EINGABE-ZEILE
```

10 S0CB – 3

Aufgabe

Suchen nach der Abbruchursache.

Dump

uid.DUMP.JOBLOG.S0CB#03.txt

Anmerkungen

Zunächst ein paar Anmerkungen zu diesem Fehler. Es taucht in vielen Programmen eine Logik auf der Art, dass bei einer komplexen fachlichen Überprüfung versucht wird, eine komplette Prüfung vorzunehmen. Beispiel Compiler: Wenn ein Compiler nach jedem Fehler sofort seine Arbeit beenden würde, wären wir fast nur mit dem Warten auf die Umwandlungen beschäftigt. Daher hat es sich eingebürgert, dass Schalter für bestimmte Fehler(-Typen) gesetzt werden. Diese werden am Anfang initialisiert und am Ende der Durchführung werden alle Schalter geprüft. Die implementierte Logik in dem benutzten Programm basiert auf einer solchen Logik. Dabei ist bei der Programmierung aber ein kleiner Fehler passiert. Und diesen gilt es zu finden.

Erläuterungen zum Programm TES46

Es wurde in den Unterlagen eine Version der Compile-Liste benutzt, die bei der Umwandlung mit Xpediter erzeugt worden ist. Die sieht ein wenig anders aus, als die „gewohnte“ Compile-Liste. Aber alle notwendigen Informationen sind vorhanden nur in einem veränderten Format.

Das Programm besteht aus 3 Teilen, von denen nur 2 für uns relevant sind.

1. Teil 1 besteht aus der Section ANALYSE-SETZEN. Diese Section initialisiert alle unsere Fehler-Schalter mit dem Wert 1.
2. Teil 2 besteht aus der Funktion, bei der Fehler auftreten könnten und dann die Schalter verändert werden. In unserem Fall werden in dieser Section keine Fehler gefunden. Die Schalter sollten also alle auf 1 stehen. Diese Section braucht uns also nicht zu interessieren.
3. Teil 3 besteht aus der Section ANALYSE-PRUEFEN. Dort werden die Schalter abgefragt. Sollte irgendein Schalter nicht auf dem Wert 1 stehen, wird ein Fehlerzähler um den Wert 1 hochgezählt. Am Ende aller Prüfungen wird dieser Fehlerzähler geprüft. Steht dieser nicht auf 0, wird ein harter Abbruch (Division durch 0) erzwungen.

Es gilt also heraus zu finden, warum – obwohl angeblich richtig kodiert wurde – eine Division durch 0 passiert.

Musterlösung

wichtigste Informationen aus Dump (wegen Umwandlung für Xpediter mit Source-Code-Unterstützung):

A decimal-divide exception occurred during execution of program TES46.
The expected completion code is SOCB.

```
*****  
*           Analysis of Error           *  
*****
```

The decimal-divide exception occurred in the following statement:

```
000257 00256          DIVIDE FEHLER-ZAEHLER BY ZAHLNULL GIVING ZAHLNULL
```

```
                Current values of fields on abending statement  
Level/Field Name      Picture/Type Offs      Value      Type  
-----  
000116      02 FEHLER-ZAEHLER      S9(09)  COMP-3      +1      Work  
000044      03 ZAHLNULL      9(09)      000000000      Work  
-----
```

Okay. Nach der Vorgeschichte ist das ja klar. Aber wo ist der krumme Hund? Der Referent behauptet ja, er habe richtig kodiert und die Division durch 0 dürfe nicht auftreten.

Also: Die Fehler heißen alle ZAHL-G, -H, -I, -J, -K usw. Diese werden abgefragt. In der Section Analyse-setzen heißen sie auch so. Hmm. Man könnte ja auch ein wenig schöner kodieren, dass die Sachen schön untereinander stehen. Hmm. Wenn man jetzt mit View in der Compileliste ist (nicht mit Browse) kann man das ja mal machen. Siehst Du schon was? Okay. Weiter.

Schauen wir mal in die Cross-Referenz:

98	ZAHL-A		
99	ZAHL-B		
102	ZAHL-C		
103	ZAHL-D		
104	ZAHL-E		
105	ZAHL-F		
108	ZAHL-G	M194 208
109	ZAHL-H	M195 214
110	ZAHL-I	220
111	ZAHL-J	M197 226
112	ZAHL-K	M198 232
113	ZAHL-L	M199 238
114	ZAHL-M	M200 244
115	ZAHL-N	M201 250
100	ZAHL-1	M196
101	ZAHL-2		

Noch Fragen?

Fazit 1: Es lohnt sich, den Code sauber auszurichten.

Fazit 2: Es lohnt sich, Felder mit gleicher Funktion gleich zu benennen und erst „hinten“ zu unterscheiden.

Fazit 3: Es lohnt sich, zu überlegen, wo welche Information stehen könnte, die mir bei der Suche behilflich sein könnte.

Fazit 4: Präge Dir ein, welche Informationen in der Compileliste vorhanden sind.

Fazit 5: Glaube niemals dem Referenten, ehe Du seine Aussagen nicht überprüft hast. ;-)