



# CICS Extreme Debugging - Basics

Ed Addison

IBM

Thursday, February 7, 2013

Session 12444



# DEBUG EXAMPLE ONE

## ABEND0C4 / AKEA / AFCZ

# ABEND0C4 / AKEA / AFCZ



- Customer called in with ABENDAKEA in CICS region

- Messages from SYSLOG

DFHFC0001 IYNXA An abend (code 0C4/AKEA) has occurred at offset X'FFFF' in module DFHFCVR.

DFHME0116 (Module:DFHMEME) CICS symptom string for message DFHFC0001 is LVLS/660 MS/DFHFC0001 RIDSD/DFHFCVR AB/S00C4 AB/UAKEA ADRS/0000FFFF

DFHDU0201 IYNXA ABOUT TO TAKE SDUMP. DUMPCODE: FC0001 , DUMPID: 17/0001

DFHDU0202 IYNXA SDUMPX COMPLETE. SDUMPX RETURN CODE X'00'

IEA611I COMPLETE DUMP ON DUMP.MV23.IYNXA.D100629.T164219.S00031

DUMPID=031 REQUESTED BY JOB (IYNXA ) FOR ASID (0044) INCIDENT TOKEN: SYSPLEX1 MV23 06/29/2010 15:42:19

- Messages from MSGUSR

DFHAC2236 16:46:15 IYNXA Transaction MESS abend AFCZ in program MESSITUP term TC13. Updates to local recoverable resources will be backed out.

DFHDU0203I 16:46:18 IYNXA A transaction dump was taken for dumpcode: AFCZ, Dumpid: 17/0004.

DFHAC2236 16:46:18 IYNXA Transaction MESS abend AFCZ in program MESSITUP term TC13. Updates to local recoverable resources will be backed out.

# ABEND0C4 / AKEA / AFCZ



- AKEA
  - Explanation: A program check has been detected by the kernel (KE) domain.
  - System Action: If an application is in control, the ASRA abend is presented to the application. Otherwise, the functional recovery routine of the CICS module in control at the time is given control. This recovery routine produces suitable diagnostics and may terminate CICS.
  - User Response: Look at the kernel domain section of the system dump to determine where the program check has occurred.
- AFCZ
  - Explanation: The transaction issued a file request resulting in a call to the main file control program (DFHFCFR). A "disastrous error" response was returned from DFHFCFR to its caller.
  - System Action: At the time the error is detected, CICS writes a message to the console, records an exception trace entry, and takes a system dump. The trace and dump identify the point of error. Subsequently, the task is abnormally terminated with a CICS transaction dump.
  - User Response: The system programmer should use the trace and dumps to determine what the error is, and why it has occurred.

# IPCS Primary Menu

```
----- IPCS PRIMARY OPTION MENU -----
OPTION  ==>>

      0  DEFAULTS    - Specify default dump and options
      1  BROWSE       - Browse dump data set
      2  ANALYSIS    - Analyze dump contents
      3  UTILITY     - Perform utility functions
      4  INVENTORY   - Inventory of problem data
      5  SUBMIT      - Submit problem analysis job to batch
      6  COMMAND     - Enter subcommand, CLIST or REXX exec
      T  TUTORIAL    - Learn how to use the IPCS dialog
      X  EXIT        - Terminate using log and list defaults

                                *****
                                *  USERID   - USASSC1
                                *  DATE      - 12/03/02
                                *  JULIAN    - 12.062
                                *  TIME     - 16:20
                                *  PREFIX   - USASSC1
                                *  TERMINAL - 3278
                                *  PF KEYS  - 24
                                *****
```

Enter END command to terminate IPCS dialog

```
F1=HELP   F2=SPLIT  F3=END    F4=RETURN  F5=RFIND  F6=MORE   F7=UP
F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT  F12=CURSOR
```

# IPCS Default Menu

```
----- IPCS Default Values -----  
Command ==>  
  
You may change any of the defaults listed below.  The defaults shown before  
any changes are LOCAL.  Change scope to GLOBAL to display global defaults.  
  
Scope ==> BOTH (LOCAL, GLOBAL, or BOTH)  
  
If you change the Source default, IPCS will display the current default  
Address Space for the new source and will ignore any data entered in  
the Address Space field.  
  
Source ==> DSNAME ('USASSC1.SHAREFC.DUMP')  
Address Space ==> ASID(X'0044')  
Message Routing ==> NOPRINT TERMINAL  
Message Control ==> FLAG(WARNING) NOCONFIRM VERIFY  
Display Content ==> MACHINE REMARK REQUEST NOSTORAGE SYMBOL  
  
Press ENTER to update defaults.  
  
Use the END command to exit without an update.
```

# MACHINE vs. NOMACHINE

- Specifying display content of **MACHINE**
  - ▶ Displays the ASID, virtual address and storage key
  - ▶ Here is an example:

```
command ==> ip 1 7000 length(20)
```

```
LIST 7000. ASID(X'0396') LENGTH(X'14') AREA  
ASID(X'0396') ADDRESS(7000.) KEY(88) ← Note key of '88'  
00007000. 02386EC4 C6C8D2C5 D2C3C240 40404040 A5900400
```

- Specifying display content of **NOMACHINE**
  - ▶ Does not display the storage key
  - ▶ Here is an example:

```
command ==> ip 1 7000 length(20)
```

```
LIST 7000. ASID(X'0396') LENGTH(X'14') AREA  
00007000. 02386EC4 C6C8D2C5 D2C3C240 40404040 A5900400
```

# IPCS Primary Menu

```
----- IPCS PRIMARY OPTION MENU -----
OPTION  ===>

      0  DEFAULTS      - Specify default dump and options
      1  BROWSE        - Browse dump data set
      2  ANALYSIS      - Analyze dump contents
      3  UTILITY       - Perform utility functions
      4  INVENTORY     - Inventory of problem data
      5  SUBMIT        - Submit problem analysis job to batch
      6  COMMAND      - Enter subcommand, CLIST or REXX exec
      T  TUTORIAL      - Learn how to use the IPCS dialog
      X  EXIT          - Terminate using log and list defaults

                                         *****
                                         * USERID  - USASSC1
                                         * DATE    - 12/03/02
                                         * JULIAN   - 12.062
                                         * TIME    - 16:20
                                         * PREFIX   - USASSC1
                                         * TERMINAL- 3278
                                         * PF KEYS  - 24
                                         *****
```

Enter END command to terminate IPCS dialog

```
F1=HELP   F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
F8=DOWN   F9=SWAP     F10=LEFT  F11=RIGHT  F12=CURSOR
```



# IPCS Command Menu

----- IPCS Subcommand Entry -----

Enter a free-form IPCS subcommand or a CLIST or REXX exec invocation below:

====> **ST SYS**

----- IPCS Subcommands and Abbreviations -----

ADDDUMP	DROPDUMP, DROPD	LISTDUMP, LDMP	RENUM, REN
ANALYZE	DROPMAP, DROPM	LISTMAP, LMAP	RUNCHAIN, RUNC
ARCHECK	DROPSYM, DROPS	LISTSYM, LSYM	SCAN
ASCBEXIT, ASCBX	EPTRACE	LISTUCB, LISTU	SELECT
ASMCHECK, ASMK	EQUATE, EQU, EQ	LITERAL	SETDEF, SETD
CBFORMAT, CBF	FIND, F	LPAMAP	STACK
CBSTAT	FINDMOD, FMOD	MERGE	STATUS, ST
CLOSE	FINDUCB, FINDU	NAME	SUMMARY, SUMM
COPYDDIR	GTFTRACE, GTF	NAMETOKN	SYSTRACE
COPYDUMP	INTEGER	NOTE, N	TCBEXIT, TCBX
COPYTRC	IPCS HELP, H	OPEN	VERBEXIT, VERBX
CTRACE	LIST, L	PROFILE, PROF	WHERE, W

# Output from ST SYS Command

## MVS Diagnostic Worksheet

Dump Title: CICS DUMP: SYSTEM=IYNXA CODE=FC0001 ID=17/0001

CPU Model 2094 Version 00 Serial no. 23F6EA Address 00  
Date: 03/02/2012 Time: 16:42:20.881062 Local

Original dump dataset: **DUMP.MV23.IYNXA.D100629.T164219.S00031**

Information at time of entry to SVCDUMP:

HASID 0044 PASID 0044 SASID 0044 PSW 070C1000 A735197C

SYSTEM STATUS:

Nucleus member name: IEANUC01

I/O configuration data:

IODF data set name: SYS1.IODF02

IODF configuration ID: PLX1

EDT ID: P1

Sysplex name: SYSPLEX1

TIME OF DAY CLOCK: C633E348 684A6DB4 06/29/2010 16:42:20.881062 local

TIME OF DAY CLOCK: C633D5DF 2E0A6DB4 06/29/2010 15:42:20.881062 GMT

Program Producing Dump: SVCDUMP

Program Requesting Dump: DFHKETCB

**Incident token: SYSPLEX1 MV23 03/02/2012 15:42:19.979815 GMT**

**Note:** Original Dump Dataset name and Incident Token matches messages seen on SYSLOG

# IPCS Command Menu

```
----- IPCS Subcommand Entry -----  
Enter a free-form IPCS subcommand or a CLIST or REXX exec invocation below:
```

```
====> VERBX DFHPD660 `KE`
```

```
----- IPCS Subcommands and Abbreviations -----  
ADDDUMP          | DROPDUMP, DROPD  | LISTDUMP, LDMP   | RENUM,         REN  
ANALYZE          | DROPMAP, DROPM  | LISTMAP, LMAP   | RUNCHAIN, RUNC  
ARCHECK         | DROPSYM, DROPS  | LISTSYM, LSYM   | SCAN  
ASCBEXIT, ASCBX | EPTRACE         | LISTUCB, LISTU  | SELECT  
ASMCHECK, ASMK  | EQUATE, EQU, EQ | LITERAL         | SETDEF, SETD  
CBFORMAT, CBF   | FIND, F         | LPAMAP          | STACK  
CBSTAT          | FINDMOD, FMOD   | MERGE           | STATUS, ST  
CLOSE           | FINDUCB, FINDU  | NAME            | SUMMARY, SUMM  
COPYDDIR        | GTFTRACE, GTF   | NAMETOKN        | SYSTRACE  
COPYDUMP        | INTEGER         | NOTE, N         | TCBEXIT, TCBX  
COPYTRC         | IPCS HELP, H    | OPEN            | VERBEXIT, VERBX  
CTRACE          | LIST, L         | PROFILE, PROF   | WHERE, W
```

# VERBX DFHPD660 'KE'

```
* * * * * CICS 6.6.0 - IPCS EXIT * * * * *
```

```
CICS660 OPERANDS:
```

```
KE
```

```
=== SUMMARY OF ACTIVE ADDRESS SPACES
```

```
ASID(hex):      JOBNAME:  
0044            IYNXA
```

```
ADDRESS SPACE ASID NUMBER (HEX) = 0044
```

```
=== DUMP SUMMARY
```

```
DUMPID:      17/0001
```

```
DUMPCODE:    FC0001
```

```
DATE/TIME:   02/03/12 16:42:20 (LOCAL)
```

```
MESSAGE:     DFHFC0001 IYNXA An abend (code 0C4/AKEA) has occurred at offset X'FFFF' in module DFHFCVR.
```

```
SYMPTOMS:    PIDS/5655S9700 LVLS/660 MS/DFHFC0001 RIDS/DFHFCVR PTFS/UK57059 AB/S00C4 AB/UAKEA ADRS/0000FFFF
```

```
TITLE:       (None)
```

```
CALLER:      (None)
```

```
ASID:        X'0044'
```

# VERBX DFHPD660 'KE' Continued

## Find \*running

===KE: Kernel Domain KE\_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB	ERROR
004B	15647700	***Running**	0005E700	00051	MESS	146F5080	13CD6FF8	*YES*

## Find '004B '

004B	1564F020	0170	Bot	93B01F00	93B02316	000416	DFHKETA
004B	1564F190	0380	Dom	93B1C268	93B1C480	000218	DFHDSKE
004B	1564F510	0880	Dom	93B44D08	93B46048	001340	DFHXMTA
004B	1564FD90	0620	Dom	94A0B9A8	94A0C998	000FF0	DFHPPGP
			Int	+0002DC	94A0BB3A	000192	INITIAL_LINK
004B	156503B0	0D40	Dom	94D09000	800829CC	000000	DFHAPLI1
			Int	+00265C	94D09B82	000B82	CICS_INTERFACE
004B	156510F0	0500	Sub	94CE8300	94CE9AE8	0017E8	DFHEIFC
			Int	+001458	94CE87E6	0004E6	CALL_FCFR
004B	156515F0	08F0	Dom	950EDA00	950F4032	006632	DFHFCFR
			Int	+0045DE	950EE25E	00085E	ACCMTEST
004B	15651EE0	0A30	Sub	952D8E00	952DEFFE	0061FE	DFHFCVS
			Int	+002E20	952D93A4	0005A4	PROCESS_INTRO_REQUEST
			Int	+00313E	952DBC7C	002E7C	READ_RECORD
			Int	+006172	952DBF66	003166	VSAM
004B	15652910	09E0	Sub	952E7090	952E839E	00130E	*Y* DFHFCVR
004B	156532F0	0EA0	Dom	93B911F0	93B94EF0	003D00	DFHMEME
			Int	+003C34	93B927B8	0015C8	TAKE_A_DUMP_FOR_CALLER
004B	15654190	0670	Dom	93C34CD8	93C368EC	001C14	DFHDUDU
			Int	+000C76	93C34EEE	000216	SYSTEM_DUMP
			Int	+001BDE	93C35E54	00117C	TAKE_SYSTEM_DUMP

# VERBX DFHPD660 'KE' Continued

## Find Table

```
==KE: KE Domain Error Table Summary
```

ERR_NUM	ERR_TIME	KE_NUM	ERROR TYPE	ERR_CODE	MODULE	OFFSET
=====	=====	=====	=====	=====	=====	=====
00000001	16:42:14	004C	PROGRAM_CHECK	0C4/AKEA	DFHTSDM-	000012D6
00000002	16:42:14	004C	PROGRAM_CHECK	0C4/AKEA	DFHTSTS-	000012D6
00000003	16:42:19	004C	PROGRAM_CHECK	0C4/AKEA	DFHTSDU-	000012D6
<b>00000004</b>	16:42:19	<b>004B</b>	PROGRAM_CHECK	0C4/AKEA	UNKNOWN	UNKNOWN

## Find 'Error Number: 00000004'

```
=KE: Error Number: 00000004
```

KERRD 13C87618 KERNEL ERROR DATA

0000	F0C3F461	C1D2C5C1	018400C4	0000FFFF	C4C6C8C6	C3E5D940	152E7090	146F5080	*0C4/AKEA.d.D...DFHFCVR	13C87618
0020	0005E700	15647700	00000004	00000010	078D1000	8370181C	00020010	40404000	*..X.....c	13C87638
0040	8370181C	80800000	00000000	00000000	00000000	158163E8	00000000	1469635C	*c.....	13C87658
0060	00000000	146B0158	00000000	40404040	00000000	00000017	00000000	2732496F	*.....	13C87678
0080	00000000	00000017	00000000	0000096F	00000000	0000D4C2	00000000	03701F01	*.....?....	13C87698

# VERBX DFHPD660 'KE' Continued

Error Code: 0C4/AKEA Error Type: PROGRAM\_CHECK Timestamp: C633D5DE50E93E34

Date (GMT) : 02/03/12 Time (GMT) : 15:42:19.975315  
Date (LOCAL) : 02/03/12 Time (LOCAL) : 16:42:19.975315

KE\_NUM: 004B KE\_TASK: 15647700 TCA\_ADDR: 0005E700 DS\_TASK: 146F5080

Program DFHFCVR was in control, but the PSW was elsewhere.

Error happened under the CICS RB.

## CICS Registers and PSW.

**PSW: 078D1000 8370181C** Instruction Length: 2 Interrupt Code: 10 **Exception Address: 40404040**

Space at Program Check/Abend: Basespace Branch Event Address: 00000000\_0370171C

64-BIT REGISTERS 0-15

**REGS** 13C87660

0000	00000000	00000000	00000000	158163E8	00000000	1469635C	00000000	146B0158
0020	00000000	<b>40404040</b>	00000000	00000017	00000000	2732496F	00000000	00000017
0040	00000000	0000096F	00000000	0000D4C2	00000000	03701F01	00000000	83700F02
0060	00000000	837016D6	00000000	156530C0	00000000	837016D6	FFFFFFFF	00000000

Data at PSW: 8370181C Module: UNKNOWN Offset: UNKNOWN

**PSWDATA 0370181C**

Storage addressed by PSW cannot be accessed \*\*

# ABEND0C4 / AKEA / AFCZ What we know so far

- CICS Region IYNXA received ABEND0C4 / AKEA at offset X'FFFF' in module DFHFCVR
- CICS Region IYNXA produced a FC0001 System Dump
- Transaction MESS transaction number 00051 with TCA address 0005E700 received abend0C4 / AKEA/ AFCZ in program MESSITUP
- Program Status Word (PSW) for ABEND0C4 was 078D1000 8370181C
- Exception Address was 40404040
- Program DFHFCVR was in control, but the PSW was elsewhere
  
- **What module was in control for the ABEND0C4?**



# Finding PSW Address

## VERBX DFHPD660 'LD'

==LD: PROGRAM STORAGE MAP

PGM NAME	ENTRY PT	CSECT	LOAD PT.	REL.	PTF LVL.	LAST COMPILED	COPY NO.	USERS	LOCN	TYP	ATTRIBUTE	R/A	MODE
DFHCCNV	96400028	DFHYA660	16400000	660			1	0	ERDSA RPL	RESIDENT	-	-	
		DFHCCNV	164001B0	0660	HCI6600	I 30/05 02.44							
MESSITUP	96610000	DFHYA660	16610000	660			1	1	ESDSA RPL	REUSABLE	-	-	

**Note:** PSW Address 0370181C not loaded or known by CICS

## IP L 0370181C L(x'1000')

```
***** TOP OF DATA *****  
LIST 0370181C. ASID(X'0044') LENGTH(X'03E8') AREA  
0370181C. LENGTH(X'03E8')==>Storage not available  
***** END OF DATA *****
```

**Note:** PSW Address not in dumped z/OS storage

## IP WHERE 0370181C

```
***** TOP OF DATA *****  
ASID(X'0044') 0370181C. IDA019L1+03881C IN EXTENDED PLPA  
***** END OF DATA *****
```

# ABEND0C4 / AKEA / AFCZ What we know so far

- CICS Region IYNXA received ABEND0C4 / AKEA at offset X'FFFF' in module DFHFCVR
  - CICS Region IYNXA produced a FC0001 System Dump
  - Transaction MESS transaction number 00051 with TCA address 0005E700 received abend0C4 / AKEA/ AFCZ in program MESSITUP
  - Program Status Word (PSW) for ABEND0C4 was 078D1000 8370181C
  - Exception Address was 40404040
  - Program DFHFCVR was in control, but the PSW was elsewhere
  - PSW was in VSAM Load Module IDA019L1 offset x'03881C'
- 
- Are we done debugging?

# VERBX DFHPD660 'TR=2'

```
AP 00E1 EIP ENTRY READ                                REQ(0004) FIELD-A(00140648 ....) FIELD-B(08000602 ....)

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-966100BE TIME-20:57:37.9100501835 INTERVAL-00.0000017656 =05531

AP E160 EXEC ENTRY READ FILE('BIGG ' AT X'16610148') INTO( AT X'40404040') LENGTH(200 AT X'001406AC') RIDFLD( AT X'80140894')
EQUAL NOHANDLE ASM

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-80082436 TIME-20:57:37.9100559648 INTERVAL-00.0000057812 =05531
1-0000 003D0000 000D1661 01700602 F0002800 00800000 01010C16 610148C2 C9C7C740 *...../....0...../..BIGG
0020 40404000 02A00440 40404001 03030600 1406AC00 C8000404 04801408 94 * .... .....H.....m
2-0000 16610170 16610148 40404040 001406AC 80140894 *./.../.. .....m
3-0000 001406B0 *....
4-0000 0205737C 0110196F D4E4C3D2 0000060C E3C3F0F2 00000004 00007D00 00000000 *...@...?MESS....TC02.....'....
0020 00000000 00000000 00000000 00000000 00000040 40404040 40404000 00000000 *.....
0040 00000000 00000000 00000000 00000000 00000000 00 *.....
5-0000 00140648 *....
6-0000 00000000 0014005C 00000000 966100BE 00000000 40404040 001406B0 8004F400 *.....*....o/..... .....4
0020 96610028 156393B0 15639CD4 14D12780 15639858 147E3B55 40404040 00140008 *o/....l....M.J....q..=... ..
0040 001400D0 008B2000 *...}....

AP 04E0 FCFR ENTRY - FUNCTION(READ_INT0) FILE_NAME(BIGG) BUFFER_ADDRESS(40404040) BUFFER_LENGTH(C8) ENVIRONMENT_IDENTIFIER(0000000)
RECORD_ID_ADDRESS(80140894) GENERIC(NO) KEY_COMPARISON(EQUAL) READ_INTEGRITY(FCT_VALUE) RECORD_ID_TYPE(KEY)
CONDITIONAL(NO) BYPASS_SECURITY_CHECK(NO)

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-948FE2E8 TIME-20:57:37.9100619335 INTERVAL-00.0000029531 =05531
1-0000 00880000 00000038 00000000 00000000 B46AC12E 7C800000 01000100 00000000 *.h.....A.@.....
0020 00000000 C2C9C7C7 40404040 00000000 00000000 00000000 00000000 40404040 *...BIGG .....
0040 000000C8 00000000 00000000 00000000 00000000 00000000 80140894 00000000 *...H.....m...
0060 00000000 00000000 00000000 00000000 00000000 00000202 01000002 01020002 *.....
0080 00000200 00004040 *.....

AP 04B0 FCVS ENTRY - FUNCTION(READ_INT0) FILE_NAME(BIGG) BUFFER_ADDRESS(40404040) BUFFER_LENGTH(C8) ENVIRONMENT_IDENTIFIER(0000000)
FCTE_POINTER(15831030) RECORD_ID_ADDRESS(80140894) WORK_ELEMENT_ADDRESS(15842300) GENERIC(NO) KEY_COMPARISON
(EQUAL) READ_INTEGRITY(FCT_VALUE) RECORD_ID_TYPE(KEY) CONDITIONAL(NO) BYPASS_SECURITY_CHECK(NO)

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-950F1932 TIME-20:57:37.9100753398 INTERVAL-00.0000076093 =05531
1-0000 00880000 00000038 00000000 00000000 B46ED12E 7C800000 01000100 00000000 *.h.....>J.@.....
0020 00000000 C2C9C7C7 40404040 00000000 00000000 00000000 00000000 40404040 *...BIGG
```

# VERBX DFHPD660 'TR=2'

```
AP 0492 FCVR EVENT ISSUE_VSAM_RPL_REQUEST - REQUEST(GET) OPTION(DIR WTX ) KEY(D9C5C3F3)

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-952E05FE TIME-20:57:37.9100850898 INTERVAL-00.0000097500 =05531
1-0000 0000004C 00000000 00000000 00000000 00000000 00000000 15892080 00000000 *...<.....i.....
0020 40404040 80140894 40840000 00000000 000000C8 000000C8 00000000 00000000 * .m d.....H...H.....
0040 00000000 00000080 158295D4 *.....bnM
2-0000 158293E8 *..bLY
3-0000 D9C5C3F3 *REC3

AP 0495 FCVR *EXC* - RECOVERY_ROUTINE_ENTERED

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-952E05FE TIME-20:57:37.9148596054 INTERVAL-00.0047745156 =05532
1-0000 00000000 98580000 158293E0 D1000004 50081577 01 *...q...bl\J...&....
2-0000 F0C3F461 C1D2C5C1 018400C4 0000FFFF C4C6C8C6 C3E5D940 152E91C0 146F4200 *0C4/AKEA.d.D...DFHFCVR ..j{.?.
0020 0005E080 155FD100 0000000D 00000010 078D1000 8370181C 00020010 40404000 *..\.~J.....c.....
0040 8370181C 80800000 00000000 00000000 00000000 158293E8 00000000 14696C6C *c.....bLY.....%
0060 00000000 146B2158 00000000 40404040 00000000 00000017 00000000 2732596F *.....
0080 00000000 00000017 00000000 0000096F 00000000 0000D4C2 00000000 03701F01 *.....?.....MB.....
00A0 00000000 83700F02 00000000 837016D6 00000000 1563C0C0 00000000 837016D6 *...c.....c..O.....{...c..
00C0 00000000 00000000 00000000 00000002 00000000 00000000 00000000 00000000 *.....
00E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....
0100 00000000 00000000 078D1000 8370181C 00020010 40404000 8370181C 80800000 *.....c.....c.....
0120 00000000 00000000 00000000 158293E8 00000000 14696C6C 00000000 146B2158 *.....bLY.....%.....,
0140 00000000 40404040 00000000 00000017 00000000 2732596F 00000000 00000017 *....?.....
0160 00000000 0000096F 00000000 0000D4C2 00000000 03701F01 00000000 83700F02 *.....?.....MB.....c..
0180 00000000 837016D6 00000000 1563C0C0 00000000 837016D6 00000000 00000000 *...c..O.....{...c..O.....

ME 0301 MEME ENTRY - FUNCTION(SEND_MESSAGE) MESSAGE_NUMBER(1) SYSTEM_DUMP CODE(FC0001) INSERT1(155FD3D0 , 00000008) INSERT2(155FD3D0
, 00000002) INSERT3(152EB7B0 , 00000008) COMPONENT_ID(FC)

TASK-00051 KE_NUM-004D TCB-QR /008F8220 RET-952EA4CE TIME-20:57:37.9148624804 INTERVAL-00.0000028750 =05532
1-0000 00F80000 00000026 00000001 00000000 B5E00200 00000000 01000000 00000000 *.8.....\.....
0020 00000000 00000001 00000000 C6C3F0F0 F0F14040 155FD3D0 00000008 155FD3DE *.....FC0001 .~L}.....~L
0040 00000002 152EB7B0 00000008 00000000 00000000 00000000 00000000 00000000 *.....
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....
```

# Finding Information About the Request

## DFHPD660 'AP=3'

```
TCA.00051 0005E700 Task Control Area (User Area)
0000 0005E800 00000001 15810D50 0004FB48 15652910 157FE030 00000000 00000000 *..Y.....a.&....."\..... *
0020 0000051C 00000000 00000000 9525D932 00000000 00000000 008B3000 0014005C *.....n.R.....* *
0040 96610064 00086D84 80050400 96610028 156503B0 15650CD4 00000004 00004000 *o/...._d....o/..... *
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... *

SYSEIB.00051 0005EB14 System EXEC Interface Block
-0008                                     5CE2E8E2 C5C9C240 *                                     *SYSEIB *
0000 0164213C 0110180F D4E4C3D2 0000051C E3C3F1F3 00000004 00007D06 01000000 *.....MESS....TC13...'.... *
0020 000000C2 C9C7C740 40404040 00000000 000000C2 C9C7C740 40404000 00000000 *...BIGG .....BIGG .... *
0040 00000000 00000000 00000000 00000000 00000000 00                                     *..... *

EIUS.00051 00140008 EXEC Interface User Structure
0000 00B46EC4 C6C8C5C9 E4E24040 40404040 00000000 00000000 00000000 00000000 *..>DFHEIUS ..... *
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... *
0040 00000000 00000000 001400D0 00000000 00140648 00000000 00000000 00000000 *.....}..... *
0060 80087E14 96610028 00000512 00140050 80050400 14D0F3B0 156503B0 15650CD4 *..=.o/.....&..}3..... *
0080 14D0B780 15650858 147E3B55 15647700 00140008 15733F54 008B3000 00000000 *..}.....=..... *
00A0 00000000 00140050 00140054 00000000 00000000                                     *.....&..... *

EIB.00051 001400D0 EXEC Interface Block
-0010                                     00656EC4 C6C8C1D7 6DC4C6C8 C5C9C25C *                                     ..>DFHAP_DFHEIB *
0000 0164213C 0110180F D4E4C3D2 0000051C E3C3F1F3 00000004 00007D06 02000000 *.....MESS....TC13..... *
0020 00000000 00000000 00000000 00000000 00000040 40404040 40404040 00000000 *..... *
0040 00000000 00000000 00000000 00000000 00000000 00                                     *..... *
```

**EIB Function READ**

**Application Register Savearea**

# IPCS Primary Menu

```
----- IPCS PRIMARY OPTION MENU -----
OPTION  ===>

      0  DEFAULTS      - Specify default dump and options
      1  BROWSE       - Browse dump data set
      2  ANALYSIS     - Analyze dump contents
      3  UTILITY      - Perform utility functions
      4  INVENTORY    - Inventory of problem data
      5  SUBMIT       - Submit problem analysis job to batch
      6  COMMAND      - Enter subcommand, CLIST or REXX exec
      T  TUTORIAL     - Learn how to use the IPCS dialog
      X  EXIT         - Terminate using log and list defaults

                                     *****
                                     *  USERID  - USASSC1
                                     *  DATE    - 12/03/02
                                     *  JULIAN   - 12.062
                                     *  TIME    - 16:20
                                     *  PREFIX  - USASSC1
                                     *  TERMINAL- 3278
                                     *  PF KEYS - 24
                                     *****
```

Enter END command to terminate IPCS dialog

```
F1=HELP   F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
F8=DOWN   F9=SWAP     F10=LEFT  F11=RIGHT F12=CURSOR
```

# Browse Mode – Register Savearea

```
ASID(X'0044')
Command ==> L 140648

00140648      00000000      00000000      0014005C      | .....* |
00140650      00000000      966100BE      00000000      40404040      | ....o/..... |
00140660      001406B0      80050400      96610028      156503B0      | .....o/..... |
00140670      15650CD4      14D0B780      15650858      147E3B55      | ...M.}.....=. |
00140680      40404040      00140008      001400D0      008B3000      | .....}.... |
00140690      00000000      00000000      00000000      00140648      | ..... |
```

**Note:** Registers 14 through 12 are stored in the Application Savearea at offset X'C' when a CICS call is issued.

Register 14 will point to where the CICS call was issued from.

Register 1 will contain the parameters when the CICS call was issued.

# Browse Mode – Register 14

```
ASID(X'0044') ADDRESS(166100BE.) STORAGE -----
Command ==> L 166100BE
166100BE                                D203 |                                K. |
166100C0  D318B04C  D503D318  313C4780  30B4D503 | L..<N.L.....N. |
166100D0  D3183140  477030DE  47F030E2  4110D068 | L.. .....0.S..} |
166100E0  41E03151  41F03120  4100D180  90E01000 | .\...0....J..\.. |
166100F0  41E03146  41F0D24C  90EF100C  96801010 | .\...0K<....o... |
16610100  58F03138  0DEF47F0  30E24110  D06841E0 | .0.....0.S..}\ |
16610110  315A50E0  10009680  100058F0  31380DEF | .!&\..o....0.... |
```

## Backup from R14 to start of module

```
16610000  C4C6C8E8  C1F6F6F0  58F0021C  58F0F0D0 | DFHYA660.0...00} |
16610010  58F0F014  58F0F00C  58FF000C  07FF5CC6 | .00..00.....*F |
16610020  C9D3D3C9  D55C0000  47F0F028  23D9C5C1 | ILLIN*...00..MES |
16610030  C4E4D7C4  E34DE45D  40F0F661  F2F961F1 | SITUP(U) 06/29/1 |
16610040  F040F1F6  4BF3F440  A9D6E2F6  F6F04040 | 0 16.34 zOS660 |
16610050  90ECD00C  183FA715  0004031C  000058F0 | ..}...x.....0 |
16610060  312805EF  50D01004  18F1BF1F  D018A784 | ....&}...1..}.xd |
```

**Note:** Register 14 will point to where the CICS call was issued from. Backing up from Register 14 will tell you the module that made the call.



# Browse Mode – Register 1

## Parameters

```
ASID(X'0044') ADDRESS(1406B0.) STORAGE -----  
Command ==> L 1406B0  
001406B0 16610170 16610148 40404040 001406AC | ./.../.. .... |  
001406C0 80140894 00000000 00000000 00000000 | ...m..... |
```

## First Parameter (EIB Function Code 0602 READ)

```
ASID(X'0044') ADDRESS(1406B0.) STORAGE -----  
Command ==> L 16610170  
16610170 0602F000 28000080 000604F0 00082800 | ..0.....0.... |
```

## Second Parameter (File Name BIGG)

```
ASID(X'0044') ADDRESS(16610148.) STORAGE -----  
Command ==> L 16610148  
16610148 C2C9C7C7 40404040 | BIGG |
```

## Third Parameter (INTO AREA passed from application)

```
ASID(X'0044') ADDRESS(40404040.) STORAGE -----  
Command ==> L 40404040  
27D4C000.:7F0BFFFF. LENGTH(X'57374000')--Storage not available
```

**Note:** Register 1 will point to the parameters when CICS was called.

# ABEND0C4 / AKEA / AFCZ What we know

- CICS Region IYNXA received ABEND0C4 / AKEA at offset X'FFFF' in module DFHFCVR
- CICS Region IYNXA produced a FC0001 System Dump
- Transaction MESS transaction number 00051 with TCA address 0005E700 received abend0C4 / AKEA/ AFCZ in program MESSITUP
- Program Status Word (PSW) for ABEND0C4 was 078D1000 8370181C
- Exception Address was 40404040
- Program DFHFCVR was in control, but the PSW was elsewhere
- PSW was in VSAM Load Module IDA019L1 offset x'03881C'
- Program MESSITUP issued EXEC CICS READ on file BIGG and passed 40404040 as the INTOAREA.
  
- What can be done?
  - ▶ Command Protect (CMDPROT) - CICS will test the first byte of passed parameters to ensure they are accessible. If not, the task will abend AEYD.

END OF DEBUG EXAMPLE ONE

ABEND0C4 / AKEA / AFCZ

# Problem Two: FCPSWAIT FCDSRECD Enqueue

## Problem Two

- Customer called Support Center indicating access to their main production File EDZFILE stalled
- IPCS Option 6 (Command) '**ST SYS**' shows:

SYSTEM STATUS:

Nucleus member name: IEANUC01

Sysplex name: EDZPLEX

TIME OF DAY CLOCK: BD969635 343B9A40 16:16:45.011897 local

TIME OF DAY CLOCK: BD96C9D0 C1DE4040 20:07:38.329572 GMT

Program Producing Dump: SVCDUMP

Program Requesting Dump: DFHKETCB

Incident token: EDZPLEX

**20:07:36.996768 GMT**

# Problem Two - VERBX DFHPD630 'DS=3'

DS_TOKEN	KE_TASK	T	S	F	P	TT	RESOURCE TYPE	RESOURCE_NAME	W	TIME OF SUSPEND	TIMEOUT DUE	DTA (DSTSK)	AD ATTACHER TOKEN	M	SUSPAREA	XM_TXN_TOKEN
02020001	09729080	N	<b>R</b>									0E7AF200	XN 0B3603F0	QR		0B3603F000 <b>08957C</b>
050A0025	0A1FF780	N	S	P	N	-	FCPSWAIT	EDZFILE	C	20:01:44.442	-	0E71B800	XM 0B3F47D8	QR	0A61C535	0B3F47D80013008C
050E0023	0A1FEB00	N	S	P	N	-	FCPSWAIT	EDZFILE	C	20:02:33.005	-	0E71BB00	XM 0B3F4C70	QR	0A61C535	0B3F4C700013120C
0510000D	0A1FF080	N	S	P	N	-	FCPSWAIT	EDZFILE	C	20:02:14.952	-	0E71BC80	XM 0B3F4AE8	QR	0A61C535	0B3F4AE80013071C
05800067	0A31E780	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:55:47.812	-	0E746080	XM 0B3A4340	QR	0A61C535	0B3A43400012054C
05820087	0A301B00	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:57:51.212	-	0E746200	XM 0B3A4960	QR	0A61C535	0B3A49600012374C
05844E79	0A376B00	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:52:06.948	-	0E746380	XM 0B3714C8	QR	0A4FEC85	0B3714C80011373C
05868EE7	0A33BB00	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:49:34.307	-	0E746500	XM 0970A4C8	QR	0B34FA4C	0970A4C80010882C
058CACBB	0A31EB00	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:51:38.836	-	0E746980	XM 0970A7D8	QR	0A4FEC85	0970A7D80011290C
058E0019	0A31E080	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:57:08.312	-	0E746B00	XM 0B3A47D8	QR	0A61C535	0B3A47D80012261C
05902EC7	0A33B400	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:54:11.273	-	0E746C80	XM 0B3717D8	QR	0A4FEC85	0B3717D80011733C
05920009	0A2C6400	N	S	P	N	-	FCPSWAIT	EDZFILE	C	20:00:14.734	-	0E746E00	XM 0B3CCAE8	QR	0A61C535	0B3CCAE80012747C
0600A279	0A394080	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:53:35.267	-	0E747080	XM 0B371340	QR	0A4FEC85	0B3713400011613C
0604C013	0A359780	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:51:10.696	-	0E747380	XM 0970A960	QR	0A4FEC85	0970A9600011204C
0606B575	0A3B1400	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:44:57.451	-	0E747500	XM 0970A1B8	QR	0B348CEC	0970A1B80010048C
06080CC5	0A3B1080	N	S	P	N	-	ENQUEUE	FCDSRECD	S	19:44:53.971	-	0E747680	XM 0970A340	QR	0E747680	0970A3400010033C
060A001F	0A2E3400	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:58:36.040	-	0E747800	XM 0B3CC4C8	QR	0A61C535	0B3CC4C80012520C
060E004D	0A2A8080	N	S	P	N	-	FCPSWAIT	EDZFILE	C	20:01:14.040	-	0E747B00	XM 0B3F44C8	QR	0A61C535	0B3F44C80012918C
06108E1D	0A359080	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:44:53.997	-	0E747C80	XM 0970A650	QR	0A7D8CDC	0970A6500010037C
068014B9	0A2C6780	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:59:44.382	-	0E756080	XM 0B3CC960	QR	0A61C535	0B3CC9600012685C
0684EC17	0A2E3080	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:59:47.687	-	0E756380	XM 0B3CC650	QR	0A61C535	0B3CC6500012695C
0686A7ED	0A3CE400	N	S	P	N	-	<b>FCPSWAIT EDZFILE</b>		C	<b>19:54:29.698</b>	-	0E756500	XM 0B371C70	QR	0A61C535	0B371C7000 <b>11794C</b>
06880657	0A301780	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:58:13.938	-	0E756680	XM 0B3A4AE8	QR	0A61C535	0B3A4AE80012456C
068AB401	0A394780	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:51:46.592	-	0E756800	XM 0B3711B8	QR	0A4FEC85	0B3711B80011337C
068C63E7	0A33B080	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:54:45.801	-	0E756980	XM 0B371DF8	QR	0A61C535	0B371DF80011840C
06902DE1	0A359B00	N	S	P	N	-	ENQUEUE	FCDSRECD	C	19:52:57.186	-	0E756C80	XM 0B371030	QR	0A4FEC85	0B3710300011500C
0692006F	0A2E3B00	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:58:23.868	-	0E756E00	XM 0B3CC030	QR	0A61C535	0B3CC0300012481C
07083A63	0A394400	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:56:04.976	-	0E784680	XM 0B3A41B8	QR	0A61C535	0B3A41B80012088C
070A2D33	0A2C6B00	N	S	P	N	-	FCPSWAIT	EDZFILE	C	19:59:13.894	-	0E784800	XM 0B3CC7D8	QR	0A61C535	0B3CC7D80012612C
070C6D39	0A394B00	N	S	P	N	-	ZCLOWAIT	DFHZARQ1	S	17:16:50.098	-	0E784980	XM 0970AC70	QR	0E784980	0970AC700074140C
070ECA83	0A359400	N	S	P	N	-	ENQUEUE	FCDSRECD	S	19:53:08.475	-	0E784B00	XM 0B371650	QR	0E784B00	0B3716500011526C

Note: None of the tasks have a TIMEOUT DUE value. This indicate the transactions do not have DTIMOUT coded.

Many of the tasks are in FCPSWAIT for EDZFILE

Task waiting longest time in FCPSWAIT is 11794

Currently running task on QR TCB is 08957

## Problem Two – What We Know So Far

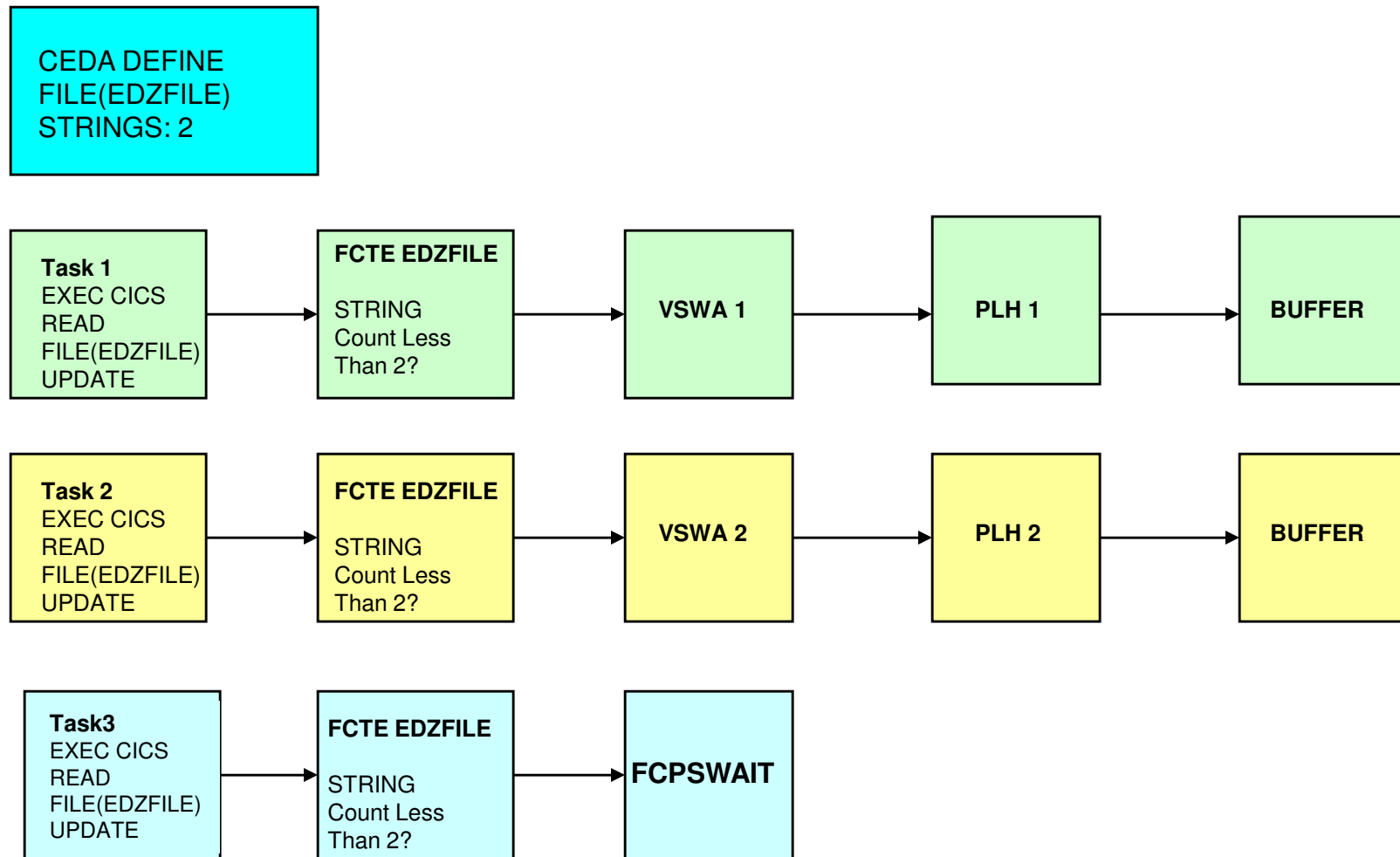
- Customer's access to file EDZFILE stalled
- Customer provided a dump taken at 20:07:36.996768 GMT
- CICS Dispatcher shows many tasks in FCPSWAIT for file EDZFILE
  - ▶ Earliest task in FCPSWAIT is 11794 at 19:54:29.698 GMT
  - ▶ DTIMOUT not set on the Transaction (no Timeout due)

## Problem Two

- FCPSWAIT
  - ▶ Waiting on a VSAM String
  - ▶ CICS commands that hold strings
    - Read for Update
      - Released at Rewrite, Delete without RIDFLD, Unlock
    - StartBrowse / ReadNext / ReadPrev / ResetBrowse
      - Released at EndBrowse or Unlock
    - MassInsert
      - Released at Unlock
    - Generic Delete
      - Released when all records are deleted
- Which tasks own the Strings for File EDZFILE?



# Problem Two - Example of FCPSWAIT



# Problem Two – FCPSWAIT - 'FCP=3'

## Find EDZFILE

==FCP: **FILE CONTROL TABLE ENTRIES**

Key for FCTE summary table :

ACC : Access type, VSAM or BDAM  
TYPE : File type, KSDS, ESDS, RRDS, VRRDS  
MODE : File mode, PATH, AIX=Alternate index, BASE  
LSR : LSR pool ID  
REM : Remote file ?  
**SLG** : **Use system log ?**  
SREQS : Servreq settings, R=Read, U=Update,  
A=Add, D=Delete, B=Browse,  
STATUS : File status, OPEN=OPEN, CLOS=CLOSE,  
OING=OPENING, CING=CLOSING,  
ENA=ENABLED, DIS=DISABLED, UNE=UNENABLED  
JID : Journal ID  
DSNB-OJB : Address of the object DSNB  
DSNB-BAS : Address of the base DSNB  
FR : Forward recovery ?  
FRL : Forward recovery log ID  
TIME OPEN : Time file opened (store clock value)

ADDRESS	FILENAME	ACC	TYPE	MODE	RLS	LSR	REM	SLG	SREQS	STATUS	JID	DSNB-OBJ	DSNB-BAS	FR	FRL	TIME	OPEN
0A61BD98	BATCHFIL	VSAM	KSDS	BASE	NO	1	NO	YES	RUADB	OPEN	ENA	0	0A61AD30	0A61AD30	YES	2	9/09/05 06:57:03
0A61C030	DDDFILE	VSAM	KSDS	PATH	NO	1	NO	YES	RUADB	OPEN	ENA	0	0A61AE00	0A61AD30	YES	2	9/09/05 06:57:03
<b>0A61C510</b>	<b>EDZFILE</b>	<b>VSAM</b>	<b>KSDS</b>	<b>BASE</b>	<b>NO</b>	<b>1</b>	<b>NO</b>	<b>YES</b>	<b>RUADB</b>	<b>OPEN</b>	<b>ENA</b>	<b>0</b>	<b>0A61D1D0</b>	<b>0A61D1D0</b>	<b>YES</b>	<b>2</b>	<b>9/09/05 06:57:12</b>
0A61C648	FFFFFILE	VSAM	KSDS	BASE	NO	1	NO	YES	RUADB	OPEN	ENA	0	0A61D2A0	0A61D2A0	YES	2	9/09/05 06:57:12
0A61C780	GGGFILE	VSAM	KSDS	BASE	NO	1	NO	YES	RUADB	OPEN	ENA	0	0A61D370	0A61D370	YES	2	9/09/05 06:57:12

**NOTE:** EDZFILE has SLG set to YES. This indicates all updates to this file are to use the System Log for backout purposes. This makes EDZFILE a recoverable file.

# Problem Two – FCPSWAIT - 'FCP=3' FCTE

```

FCTE.EDZFILE 0A61C510 FCT ENTRY

0000  C5C4E9C6 C9D3C540 00000000 00000000 00000000 0134BA0A 8001440A 84000000 *EDZFILE .....*
0020  00000000 40000000 00800000 00004000 00000000 00000000 00000000 001EC70F *.... .....G.*
0040  001EC5D3 00000000 0000057E 00000000 00000000 00000000 BD961923 E89B7640 00000000 *..EL.....=.....O...*
0060  0A61D1D0 0A61D1D0 0A4362E0 00000000 00000000 00000000 00000000 00000000 *./J}./J}...\.....*
0080  0801A804 01008000 000C001F 000F000C 0000001F 00000000 001F0001 0000046C *..y.....%*
00A0  00000000 0010000F 0A41FDF0 40000200 00000000 00000000 40404040 40404040 *.....0 .....*
00C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
00E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0100  00000000 40404040 40404040 40404040 40404040 00000000 00000000 00000000 *.....*
0120  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*

```

## FCTE OFFSETS

```

+00 FCTDSID - DDNAME
+60 FCTDSDP - DSNB Pointer
+88 FCTDSASC - Active String Count X '000C' = 12
+8A FCTDSCWC - Current String Wait Count X '001F' = 31
+8C FCTDSMSC - Upper Limit for String Count X '000F' = 15
+8E FCTDSPMS - Upper Limit for non-direct Read String Count X '000C' = 12
+90 FCTDSTSW - Total Tasks Waited for Strings X '001F' = 31

```

Note: All 12 strings for non-direct Reads are active. There are 31 tasks waiting for a string to file EDZFILE

## Problem Two – What We Know So Far

- Customer's access to file EDZFILE stalled
  - ▶ EDZFILE is a Recoverable file (use System Log = YES)
- Customer provided a dump taken at 20:07:36.996768 GMT
- CICS Dispatcher shows many tasks in FCPSWAIT for file EDZFILE
  - ▶ Earliest task in FCPSWAIT is 11794 at 19:54:29.698 GMT
  - ▶ DTIMOUT not set on the Transaction (no Timeout due)
- VERBX DFHPD660 'FCP=3' for file EDZFILE shows
  - ▶ 12 (x'C') Active Strings against the file
  - ▶ 31 (X'1F') Tasks waiting for a string
  - ▶ 15 (x'F') Strings defined for the file
  - ▶ 12 (x'C') Strings for non-direct reads

# Problem Two – FCPSWAIT - 'FCP=3' DSNB and VSWAs

DSNB 0A61D1D0 DATASET NAME BLOCK										
0000	C4E2D56D	C2D3D27A	<u>C5C4E9C6</u>	<u>C9D3C54B</u>	<u>D2E2C4E2</u>	<u>40404040</u>	<u>40404040</u>	<u>40404040</u>	*DSN_BLK: <u>EDZFILE.KSDS</u>	*
0020	40404040	40404040	40404040	40404040	40404040	0000016B	0000016B	00011DD2	*	.....K*
0040	80000000	00010001	F902000A	00000000	00004800	<u>0DBCC2C0</u>	00000000	00010000	*	.....9.....B{.....*
0060	01400000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	.....*
0080	00000000	0A61C510	0000082C	10404040	40404040	40404040	40404040	40404040	*	...../E.....*
00A0	40404040	40404000	00000000	00000000	00000000	0A61C510	00000000	00000000	*	...../E.....*
00C0	00000000	00000000	00000000						*	.....*
VSWA 0DBCC2C0 VSAM WORK AREA										
0000	8F400160	40404040	0000 <u>00</u> 4C	<u>00000000</u>	00000000	00000000	00010000	00000000	*	. .- .....<.....*
0020	0A41FDF0	00000000	00000000	09FF42E8	<u>4082</u> 0000	00000000	0000046C	0000046C	*	...0.....Y.e.....%...%*
0040	00000000	00000000	00000000	00000080	0DBCC3A0	0A61C510	0DBCC380	0DBCC390	*	.....C../E...C...C.*
0060	000E000A	00000000	00000000	8000000A	00000000	00000000	0DB9FB00	00000000	*	.....*
0080	00000000	00000000	00000000	00D00000	00000003	00000000	00000000	<u>0DBBF080</u>	*	.....}.....0.*
00A0	0DBB7300	00010000	00000000	00000000	00000000	C1C1C1C1	C1C1C1C1		*	.....AAAAAAA*

## DSNB Offsets

- +08 FCTDNAME - Dataset Name
- +54 FCTBCVSC - VSWA Anchor Chain

## VSWA Offsets

- +0A VSWAREQ - RPL Request Type
- +0C VSWAPLHP - PLH Address
- +30 VSWAOPTC - RPL Options
- +9C VSWASV12 - TCA Address

NOTE: PLH Address of 00000000 indicates this VSWA is not connected to a VSAM String.  
The Task associated to this VSWA would be in a FCPSWAIT waiting for a VSAM String.

# Problem Two – FCPSWAIT - VSWA Owning a String

```

VSWA 0B398570 VSAM WORK AREA

0000 8F000160 00000000 0000004C 0E8294AC 40000000 00000000 00010000 00000000 *...-.....<.bm. ....*
0020 0A41FDF0 00000000 0D4B34A0 0D4B34A0 40820000 00000000 0000046C 0000046C *...0.....e....%...%*
0040 00000000 00000000 00000000 00000080 0B398650 0A61C510 0B398630 0B398640 *.....f&.f...f *
0060 000E000A 00000000 00000000 B000000A 00000000 00000000 0B382830 0B398DB0 *.....*
0080 00000000 00000000 00000000 00C80001 00000000 00000000 26850000 097A5680 *.....H.....e...:..*
00A0 0B3910A8 00010000 00000000 00000000 00000000 C1C1C1C1 C1C1C1C1 *.....AAAAAAAA *

VSWA 0B382830 VSAM WORK AREA

0000 8F000160 00000000 0000004C 0E8290BC 40000000 00000000 00010000 00000000 *...-....b.. ....*
0020 0A41FDF0 00000000 0D443910 0D443910 40820000 00000000 0000046C 0000046C *...0.....e.....%...%*
0040 00000000 00000000 00000000 00000080 0B382910 0A61C510 0B3828F0 0B382900 *...../E...0...*
0060 000E000A 00000000 00000000 B000000A 00000000 00000000 0B382580 0B398570 *.....e.*
0080 00000000 00000000 00000000 00C80001 00000000 00000000 26850000 097AA080 *.....H.....e...:..*
00A0 0B35EF30 00010000 00000000 00000000 00000000 C1C1C1C1 C1C1C1C1 *.....BBBBBBBB *

```

## VSWA Offsets

- +0A VSWAREQ - RPL Request Type
- +0C VSWAPLHP - PLH Address
- +30 VSWAOPTC - RPL Options
- +9C VSWASV12 - TCA Address
- +B4 VSWAXKEY - RIDFLD (KEY)

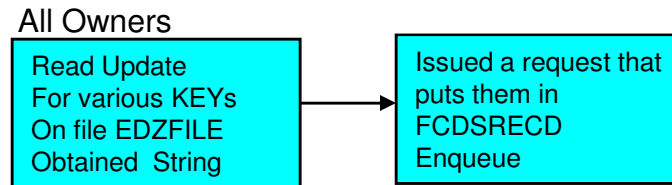
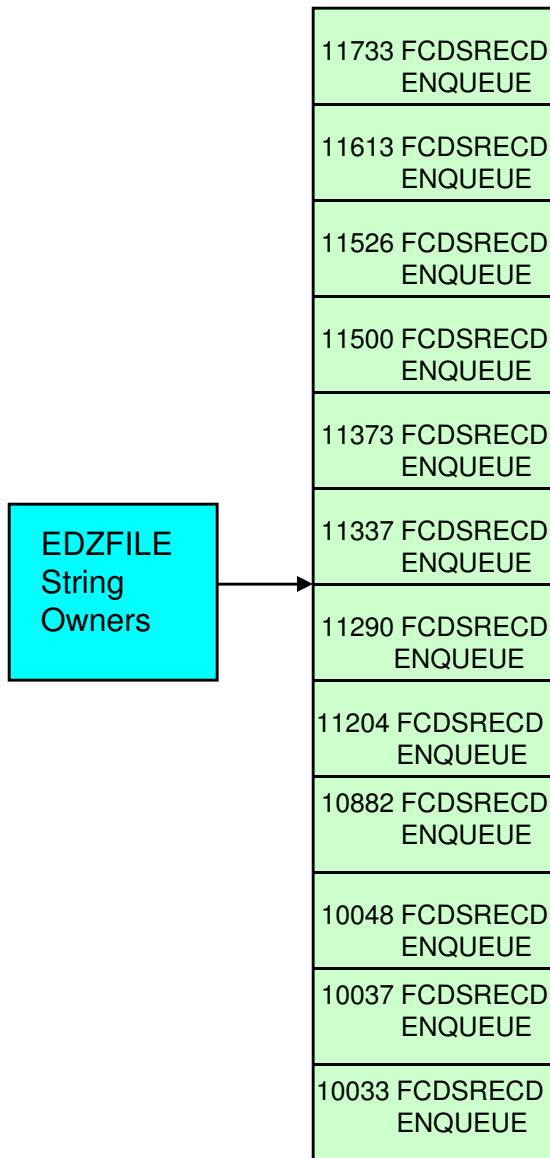
**Note:** PLH of non-zero indicates this VSWA owns a VSAM String on File EDZFILE. The 00 for VSWAREQ indicates this is a READ command. VSWAOPTC has Update Access set on, so this is a READ UPDATE command. The VSAM String will not be released until a REWRITE or Unlock is issued. The RIDFLD for VSWA1 is AAAAAAAAAA, VSWA2 is BBBBBBBB

# Problem Two – FCPSWAIT - EDZFILE String Owners

<b>TCA Address</b>	<b>Task Number</b>	<b>Wait Type</b>	<b>Wait Time</b>
097A5680	- 11733	- ENQUEUE FCDSRECD	19:54:11.273
097AA080	- 11613	- ENQUEUE FCDSRECD	19:53:35.267
097AF080	- 11526	- ENQUEUE FCDSRECD	19:53:08.475
097AF680	- 11500	- ENQUEUE FCDSRECD	19:52:57.186
097A8080	- 11373	- ENQUEUE FCDSRECD	19:52:06.948
097A7680	- 11337	- ENQUEUE FCDSRECD	19:51:46.592
097A8680	- 11290	- ENQUEUE FCDSRECD	19:51:38.836
097A9680	- 11204	- ENQUEUE FCDSRECD	19:51:10.696
097A9080	- 10882	- ENQUEUE FCDSRECD	19:49:34.307
097AC080	- 10048	- ENQUEUE FCDSRECD	19:44:57.451
097AC680	- 10037	- ENQUEUE FCDSRECD	19:44:53.997
097AB680	- 10033	- ENQUEUE FCDSRECD	19:44:53.971

Note:

1. Retrieve the TCA Address from all VSWAs owning a String.
2. Issue CICS VERBX with 'KE=3' and find the TCA Address to get the Task Number.
3. Issue CICS VERBX with 'DS=3' and find the Task Number to see the wait type of the String owners.





## Problem Two – What We Know So Far

- Customer's access to file EDZFILE stalled
  - ▶ EDZFILE is a Recoverable file (use System Log = YES)
- Customer provided a dump taken at 20:07:36.996768 GMT
- CICS Dispatcher shows many tasks in FCPSWAIT for file EDZFILE
  - ▶ Earliest task in FCPSWAIT is 11794 at 19:54:29.698 GMT
  - ▶ DTIMOUT not set on the Transaction (no Timeout due)
- VERBX DFHPD660 'FCP=3' for file EDZFILE shows
  - ▶ 12 (x'C') Active Strings against the file - 31 (X'1F') Tasks waiting for a string
  - ▶ 15 (x'F') Strings defined for the file - 12 (x'C') Strings for non-direct reads
- String Owners for EDZFILE have all issued Read Update commands for various Ridflds (obtaining a string ) and are now in FCDSRECD ENQUEUE suspends

# FCDSRECD ENQUEUE

- A resource name of FCDSRECD indicates a wait for a record lock in a VSAM file
- When a transaction updates a record in a VSAM file locking occurs at two levels:
  - ▶ VSAM locks the CI when the record has been read, and CICS locks the record
- The CI lock is released as soon as the REWRITE (or UNLOCK) request is completed. However, if the file is recoverable, the record is not unlocked by CICS until the updating transaction has reached a syncpoint
  - ▶ This is to ensure that data integrity is maintained if the transaction fails before the syncpoint and the record has to be backed out
- If a transaction attempts to update a record that is locked by another transaction, it is suspended on resource type ENQUEUE until the lock is released.
  - ▶ This can be a long wait since the owner of the ENQUEUE may itself be waiting

## Problem Two – FCDSRECD Enqueue – ‘NQ=3’

OWNER / WAITER	Enqueue Name	Len	Sta	NQEA Address	Tran id	Tran Num	Local Uowid	Lifetime Uow	Hash Tsk	Indx
X' <u>0A61AD30</u> .....		14	Act	09748600	BTCH	<u>08957</u>	BD96C4BB6ED08820	1	0	19
X' <u>C1C1C1C1C1C1C1</u> ' <u>Waiter</u> :				097483C0	EAA2	<u>10033</u>	BD96C4BA46EF8680	1	0	19
X' <u>0A61AD30</u> .....		14	Act	09748600	EAA2	<u>08957</u>	BD96C4BB6ED08820	1	0	19
X' <u>C2C2C2C2C2C2C2</u> ' <u>Waiter</u> :				097483C0	EAA2	<u>10037</u>	BD96C4BA46EF8680	1	0	19
X' <u>0A61AD30</u> .....		14	Act	09748600	EAA2	<u>08957</u>	BD96C4BB6ED08820	1	0	19
X' <u>C3C3C3C3C3C3C3</u> ' <u>Waiter</u> :				097483C0	EAA2	<u>10048</u>	BD96C4BA46EF8680	1	0	19
.										
.										
.										
X' <u>0A61AD30</u> .....		14	Act	09748600	EAA2	<u>08957</u>	BD96C4BB6ED08820	1	0	19
X' <u>D3D3D3D3D3D3D3</u> ' <u>Waiter</u> :				097483C0	EAA2	<u>11773</u>	BD96C4BA46EF8680	1	0	19

NOTE: An FCDSRECD ENQUEUE name will always be the HEX address of the DSNB (0A61AD30) followed by the RIDFLD that is locked (C1C1C1C1C1C1C1 or AAAAAAAA). In this case, the owner of all the record locks that the string holders for EDZFILE need is task number 08957. This happened to be the running task identified on slide 54.

The DSNB that all the enqueues are for ( 0A61AD30 ) is recoverable file BATCHFIL. You can verify this using slide 58. These enqueues will not be released until task 08957 either syncpoints or terminates.

## Problem Two – What We Know

- Customer's access to file EDZFILE stalled
  - ▶ EDZFILE is a Recoverable file (use System Log = YES)
- Customer provided a dump taken at 20:07:36.996768 GMT
- CICS Dispatcher shows many tasks in FCPSWAIT for file EDZFILE
  - ▶ Earliest task in FCPSWAIT is 11794 at 19:54:29.698 GMT
  - ▶ DTIMOUT not set on the Transaction (no Timeout due)
- VERBX DFHPD630 'FCP=3' for file EDZFILE shows
  - ▶ 12 (x'C') Active Strings against the file - 31 (X'1F') Tasks waiting for a string
  - ▶ 15 (x'F') Strings defined for the file - 12 (x'C') Strings for non-direct reads
- String Owners for EDZFILE have all issued Read Update commands for various Ridflds (obtaining a string ) and are now in FCDSRECD ENQUEUE suspends
- Owner of the FCDSRECD Enqueues for file EDZFILE is Task 08957 which is currently running and updating BATCHFIL

## Problem Two – What can be done???

- Code all applications to access files in the same order
  - ▶ In this case, the owner of FCDSRECD Enqueue was a new transaction that updated all records in the recoverable dataset BATCHFIL
  - ▶ Application logic was moved from batch environment since the company went global and the files had to be available 24X7
- Code Syncpoint commands to release record locks
- Code DTIMOUT value for the transactions
  - ▶ Transactions waiting would abend AFCY

# End Of Problem Two: FCPSWAIT FCDSRECD Enqueue

# Problem Three ABEND0C1

# ABEND0C1 – SYSLOG

```
DFHAP0001 EDZAOR An abend (code 0C1/AKEA) has occurred at offset X'FFFFFFFF' in module EDZPROG
DFHME0116 EDZAOR (Module:DFHMEME) CICS symptom string for message DFHAP0001 is PIDS/5655M1500 LVLS/650
MS/DFHAP0001 RIDS/DFHSRP
PTFS/HCI6500 AB/S00C1 AB/UAKEA RIDS/EDZPROG ADRS/FFFFFFFF
DFHDU0201 EDZAOR ABOUT TO TAKE SDUMP. DUMPCODE: AP0001 , DUMPID: 1/0002
IEA045I AN SVC DUMP HAS STARTED
IEA794I SVC DUMP HAS CAPTURED:
DUMPID=001 REQUESTED BY JOB (DFHSTART)
DUMP TITLE=CICS DUMP: SYSTEM=EDZAOR CODE=AP0001
DFHDU0202 EDZAOR SDUMPX COMPLETE. SDUMPX RETURN CODE X'00'
IEA611I COMPLETE DUMP ON SYS2.DUMP.EAA.D090929.T162421.S00001
```



# VERBX DFHPD650 'KE'

```
* * * * * CICS 6.5.0 - IPCS EXIT * * * * *
```

```
CICS650 OPERANDS:
```

```
KE
```

```
=== SUMMARY OF ACTIVE ADDRESS SPACES
```

```
ASID(hex):      JOBNAME:  
006C            EDZAOR
```

```
-- DFHPD0121I FORMATTING CONTROL BLOCKS FOR JOB EDZAOR
```

```
=== DUMP SUMMARY
```

```
DUMPCODE: AP0001
```

```
DATE/TIME: 29/09/09 12:34:24 (LOCAL)
```

```
MESSAGE: DFHAP0001 EDZAOR An abend (code 0C1/AKEA) has occurred at offset X'FFFFFFFF' in module EDZPROG.
```

```
SYMPTOMS: PIDS/5655M1500 LVLS/650 MS/DFHAP0001 RIDS/DFHSRP PTFS/HCI6500 AB/S00C1 AB/UAKEA RIDS/EDZPROG  
ADRS/FFFFFFFF
```

```
TITLE:      (None)
```

```
CALLER:     (None)
```

```
ASID:       X'006C'
```

# VERBX DFHPD650 'KE' Continued

## Find \*running

===KE: Kernel Domain KE\_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB	ERROR
0033	2B1FF900	***Running**	0005D080	00048	EDZZ	2A4D8500	2A2CAFF8	*YES*

## Find '0033'

KE_NUM	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFSET	ERR	NAME
0033	2B266020	0120	Bot	AA101C00	AA101FBC		0003BC		DFHKETA
0033	2B266140	0320	Dom	AA11A5F8	AA11A810		000218		DFHDSKE
0033	2B266460	0820	Dom	AA142A48	AA143C08		0011C0		DFHXMETA
0033	2B266C80	05D0	Dom	AA80B970	AA80C92A		000FBA		DFHPGPG
			Int	+0002DC	AA80BB02		000192		INITIAL_LINK
0033	2B267250	0AD0	Dom	AAA0E900	AA5A0876		000000	*Y*	DFHAPLI1
			Int	+002FCA	AAA0F3A4		000AA4		LE370_INTERFACE
			Int	+00267C	AAA12A70		004170		DO_LE370_RUNUNIT_END_INVOCATION
			Int	+002BAA	AAA11010		002710		INVOKE
0033	2B267D20	04F0	Sub	AA59E738	AA59F802		0010CA		DFHSRP
0033	2B268210	0E50	Dom	AA18CDB0	AA190A14		003C64		DFHMEME
			Int	+003222	AA18D016		000266		SEND
			Int	+00146E	AA1900A8		0032F8		CONTINUE_SEND
			Int	+003B98	AA18E328		001578		TAKE_A_DUMP_FOR_CALLER
0033	2B269060	0620	Dom	AA22E960	AA22F5E4		000C84		DFHDUDU
			Int	+000B26	AA22EB60		000200		SYSTEM_DUMP
			Int	+001934	AA22F8B4		000F54		TAKE_SYSTEM_DUMP

# VERBX DFHPD650 'KE' Continued

## Find Table

==KE: KE Domain Error Table Summary

ERR_NUM	ERR_TIME	KE_NUM	ERROR TYPE	ERR_CODE	MODULE	OFFSET
=====	=====	=====	=====	=====	=====	=====
00000001	12:24:21	003B	PROGRAM_CHECK	0C4AKEA	UNKNOWN	UNKNOWN
00000002	12:24:25	003B	TRAN_ABEND_PERCOLATE	---/ASRA	DFHSR1	00000598
<b>00000003</b>	12:34:22	0033	<b>PROGRAM_CHECK</b>	<b>0C1/AKEA</b>	UNKNOWN	UNKNOWN

## Find 'Error Number: 00000003'

=KE: **Error Number: 00000003**

KERRD 2A27E880 KERNEL ERROR DATA

0000	F0C3F161	C1D2C5C1	018400C1	0000FFFF	C4C6C8C1	D7D3C9F1	2AA0E900	2A4D8500	* <b>0C1/AKEA</b> .d.A....DFHAPLI1..Z..(e.*
0020	0005D080	2B1FF900	00000003	00000001	FF850001	00000000	078D2000	80000002	*..}...9.....e.....*
0040	00020001	7F537000	80000002	802B7A46	00000000	7FFFF000	7FFFF000	7FFFF000	*....":.....".0.".0.".0.*
0060	7FFFF000	7FFFF000	7FFFF000	00000000	00000000	7FFFF000	7FFFF000	00000000	*".0.".0.".0.....".0.".0.....*
0080	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*

# VERBX DFHPD650 'KE' Continued

**Error Code: 0C1/AKEA** Error Type: PROGRAM\_CHECK Timestamp: C4DCA455280B9CC6

Date (GMT) : 29/09/09 Time (GMT) : 16:34:22.295225  
Date (LOCAL) : 29/09/09 Time (LOCAL) : 12:34:22.295224

**KE\_NUM: 0033** KE\_TASK: 2B1FF900 TCA\_ADDR: 0005D080 DS\_TASK: 2A4D8500

Program DFHAPLI1 was in control, but the PSW was elsewhere.

Error happened under the CICS RB.

## **CICS Registers and PSW.**

**PSW: 078D2000 80000002** Instruction Length: 2 Interrupt Code: 01

Execution key at Program Check/Abend: 8

Space at Program Check/Abend: Basespace **Branch Event Address: 2B7A46E4**

## **REGISTERS 0-15**

REGS 2A27E8D0

0000	00000000	7FFFF000	7FFFF000	7FFFF000	7FFFF000	7FFFF000	7FFFF000	7FFFF000	00000000
0020	00000000	7FFFF000	7FFFF000	00000000	00000000	00000000	00000000	00000000	00000000

Data at PSW: 80000002 Module: UNKNOWN Offset: UNKNOWN

# ABEND0C1 What we know so far

- Transaction 00048 EDZZ received ABEND0C1 at x'FFFFFFFF' in EDZPROG
- PSW points to 80000002
- Registers do not show any clue as to how the PSW points to low core

# Introduction to the BEAR

- BEAR – Breaking Event Address Register
- The BEAR is a hardware enhancement in z9-109 (z/Architecture mode)
- The BEAR contains the address of the last instruction that caused a break in sequential instruction - the last successful branch address
- The minimum operating system level to take advantage of the BEAR is z/OS V1.7
- The BEAR is in the base product beginning with CICS TS V4.1
  - ▶ If you are running CICS TS V3.1 or V3.2, apply the following PTFs to use the BEAR:
    - APAR PK47015, PTF UK28041 - CICS TS V3.1
    - APAR PK47715, PTF UK28429 - CICS TS V3.2
- When the Kernel domain is formatted in a CICS dump with the added BEAR support, the new BEAR data will be formatted in the Kernel error information with the title 'Branch Event Address'

# Browsing BEAR address

L 2B7A46E4

2B7A46E4		<b>07FE</b> 0000	D7C1E3C3	C840C1D9		....PATCH AR	
2B7A46F0	C5C14060	40404040	40404040	40F2F0F0		EA - EDZPROG 200	
2B7A4700	F74BF0F8	F740B13E	B140B142	B144B146		7.087 ... ..	

**NOTE:** 07FE is an Unconditional Branch to what's in R14. R14 at the time of the ABEND0C1 was 00000000. This is an Unconditional Branch to location 0. This is why there is ABEND0C1 PSW of 80000002

Backing up from PSW Address

2B7A46C0	41302004	5030D088	58F0B178	4110D088		....&}.}h.0....}h	
2B7A46D0	05EF5850	C2E0D247	<b>41400000</b>	<b>18D498EC</b>		...&B\K.&}.}..Mq.	
2B7A46E0	<b>D00C1FFF</b>	<b>07FE</b> 0000	D7C1E3C3	C840C1D9		}.....PATCH AR	
2B7A46F0	C5C14060	40C3C5C5	C3C5D5C3	40F2F0F0		EA - EDZPROG 200	
2B7A4700	F74BF0F8	F740B13E	B140B142	B144B146		7.087 ... ..	

**41400000** LOAD ADDRESS R4, with 00000000 (R4=00000000)

**18D4** LOAD REGISTER R13 with R4 (R13=00000000)

**98ECD00C** LOAD MULTIPLE R14 through REG12 with what is in REG13 PLUS OFFSET 12

Load all registers from lowcore

**1FFF** SUBTRACT LOGICAL REGISTER R15 with R15 (R15=00000000)

**07FE** UNCONDITIONAL BRANCH TO R14

# Using INSTR to list Assembler Instructions

## Raw storage at 2B7A46D0

```
2B7A46D0  05EF5850  C2E0D247  41400000  18D498EC  | ...&B\K.&..}..Mq. |
2B7A46E0  D00C1FFF  07FE0000
```

## ip list 2B7A46D8 instr len(14)

```
LIST 2B7A46D8. ASID(X'006C') LENGTH(X'30') INSTRUCTION
2B7A46D8 | 4140 0000      | LA    R4,X'00'
2B7A46DC | 18D4          | LR    R13,R4
2B7A46DE | 98EC D00C     | LM    R14,R12,X'C' (R13)
2B7A46E2 | 1FFF         | SLR   R15,R15
2B7A46E4 | 07FE         | BCR   X'F',R14
```



# ABEND0C1 What we know

- Transaction 00048 EDZZ received ABEND0C1 at x'FFFFFFFF' in EDZPROG
- PSW points to 80000002
- Registers do not show any clue as to how the PSW points to low core
- BEAR pointed to last successful branch address
- EDZPROG loaded registers from low core and branched to location 00000000

# Summary

- **Problem One – ABEND0C4 / AKEA / AFCZ**
- **Problem Two – FCPSWAIT / FCDSRECD Enqueue**
- **Problem Three – ABEND0C1**