



IBM IT Education Services

## E33 / E34

Charles E. Olsen

Problem Determination for CICS TS for VSE/ESA

**VSE Technical Conference**

November 10 - 12, 2003 | Hilton, Las Vegas, NV

© 2003 IBM Corporation



This page  
intentionally  
left blank

# Problem Determination for CICS TS

## Sessions E33 and E34

- CICS TS dump processing .....Page 5
- Transaction Abends ..... Page 11
  - ⇒ Transaction Dumps (DFHDU410) ..... Page 16
  - ⇒ Locate Failing Instruction in COBOL/VSE ..... Page 21
  - ⇒ On-line Problem Determination (OLPD) ..... Page 29
  - ⇒ System Dumps (DFHPD410)..... Page 33
- Combining VSE SYSDUMP files..... Page 47
- AKEB / ASRB Transaction Abends ..... Page 47
- Storage Violations..... Page 59
- Short on Storage .....Page 71
- Diagnosing a CICS TS Partition Hang .....Page 83
  - ⇒ CICS TS usage of VSE sub-tasking.....Page 88
  - ⇒ CICS TS Transaction Waits .....Page 94
- CICS TS Trace (DFHTU410)..... Page 107

VSE/ESA  
Technical Conference  
Las Vegas  
Nov 10th – 12th, 2003

Charles E. Olsen  
[colsen@us.ibm.com](mailto:colsen@us.ibm.com)

# Problem Determination for CICS TS Sessions E33 and E34

## Speaker Notes:

The material in this presentation has been assembled from externally available sources, including:

“CICS TS Data Areas” (LY33-6086-00)

“CICS TS Diagnostic Reference” (LY33-6085-00)

“CICS TS Problem Determination Guide” (GC33-1663-00)

“CICS TS Trace Entries” (SX33-6108-00)

CICS TS Formatted Dump

Nothing in this presentation should be interpreted as an official programming interface.

# CICS TS Dump Processing

**DFHAC2206 19:12:05 DEDCCICS Transaction PGCK has failed with abend ASRA.**

-Or-

**F2 0103 DFHAP0001 DEDCCICS An abend (code OC9/AKEA) has occurred at offset X'0000072E' in module CICSPGCK.**

ALL abends should be investigated

**abcode axxy**

**'a'** is either **'A'** (CICS) or **'D'** (DL/I)

**xx** is domain code

**y** is alphanumeric code

AKEA (ASRA) Termination due to program interrupt (progck)

AKEB (ASRB) Termination due to VSE system abend (invalid SVC, etc)

AKEC Termination due to run-away task

**'KE'** is "Kernel domain".

**'SR'** is "System Reliability Domain (DFHSRP)".

# CICS TS Dump Processing

## CICS Dump Datasets:

CICS maintains one or two dump datasets: DFHDMPA, DFHDMPB. Defining CICS dump datasets is described in “*CICS TS System Definition*” (SC33-1651). These dump datasets should not be confused with VSE SYSDUMP datasets. They are sequential files, and so may be managed by VSE/VSAM Space Management.

## The following types of dumps are taken by CICS:

### 1. Transaction Dumps

If a transaction terminates, CICS may write a transaction dump to the dump dataset (DFHDMPA or DFHDMPB), which is currently open. Depending on the abend code and dump settings, a system dump may also be taken (see below). Transaction dumps contain only CICS control blocks and transaction storage, and are printed/formatted using DFHDU410, see “*CICS TS Operations and Utilities Guide*” (SC33-1654-02). Since these dumps contain only application-specific control areas, they are generally of limited usefulness for system problems.

### 2. System Dumps (SDUMP)

are either IDUMPS or PDUMPS. IDUMPs are written to VSE SYSDUMP dataset, and are processed using INFOANAL. They can be formatted to show CICS control blocks using DFHPD410. PDUMPS are taken (re-directed) to SYSLST if an IDUMP request fails, unless VSE/ESA option SYSDUMPC is set, see “*CICS TS Operations and Utilities Guide*” (SC33-1654-02).

These dumps can be VERY LARGE (1.5 million lines, 25,000 pages, 200Meg)

# CICS TS Dump Processing

## A Transaction Dump is taken:

1. CICS detects an abend while a user transaction is in control
2. Explicit request from application

**EXEC CICS DUMP TRANSACTION**

## A System Dump is taken:

1. If CICS (including DL/I) abnormally terminates.
2. If systems operator requests **CEMT PERFORM SHUTDOWN ... DUMP**
3. If systems operator requests **CEMT PERFORM [SNAP | DUMP] TITLE('.....')**

This is *not* the same as an **AR DUMP**. The **CEMT SDUMP** “freezes” the CICS partition, whereas the **AR DUMP** allows CICS to continue to run during the dump. This may cause incompatibilities in the dumped data. The **CEMT snap dump** is also written to VSE SYSDUMP where it can be formatted using DFHPD410, instead of tape. In addition, under CICS TS, the sdump also includes relevant sections of the VSE Supervisor and System GETVIS.

4. If an application makes a specific dump request: **EXEC CICS PERFORM DUMP**  
See “*CICS Application Programming Reference*”, SC33-1658.
5. CICS TS allows system dumps to be requested for specified transaction abend codes.  
See “*CICS TS Problem Determination*” under 3.2 Using dumps in problem determination  
The CICS support group can also provide assistance with user exits.

# CICS TS Dump Processing

## Speaker Notes:

A **CEMT SNAP** dump is *not* the same as an **AR DUMP**. The **CEMT SDUMP** “freezes” the CICS partition, whereas the **AR DUMP** allows CICS to continue to run during the dump. This may cause incompatibilities in the dumped data. The **CEMT** snap dump is also written to VSE SYSDUMP where it can be formatted using DFHPD410, instead of tape. In addition, under CICS TS, the sdump also includes relevant sections of the VSE Supervisor and System GETVIS.

If the CICS TS partition is locked out, and does not respond to a **CEMT** command, an attention routine dump (**AR DUMP**) may be taken. However, the partition should first be taken out of the dispatch chain (suspended) to prevent the partition from continuing to run and producing a corrupted dump. Mount a scratch tape on 181, and use the following series of commands:

**SUSPEND F2**

**DUMP F2,181**

**DUMP SVA,181**

**DUMP SUP,181**

**RESUME F2**

File one of this dump tape (F2 partition) may be loaded into a VSE/ESA SYSDUMP library and formatted using DFHPD410.



# CICS TS Dump Processing

## Dump control for CICS TS:

Most of the following functions described for CEMT are also available via the CICS system programming interface (SPI) using “EXEC CICS”.

See “*CICS TS: System Programming Reference*” (SC33-1659-01)

1. System dumps (apart from CICS kernel domain dumps) can be suppressed or enabled:
  - **DUMP=YES | NO** (system initialization parameter)
  - **CEMT SET SYSTEM DUMPING | NOSYSDUMP**
  - Dump domain **XDUREQ** global user exit module.
2. Global Suppression - Inquire whether system dumping has been suppressed globally by using **CEMT INQUIRE SYSTEM DUMPING**. Control system dumping using **CEMT SET SYSTEM DUMPING (NOSYSDUMP)**.
3. System Dump Code Table - Allows suppression of system dumps for specific dump codes.



“NSY” and “NSH” are not displayed on the dump code line, only the actions, “SYS” and “SHU”.

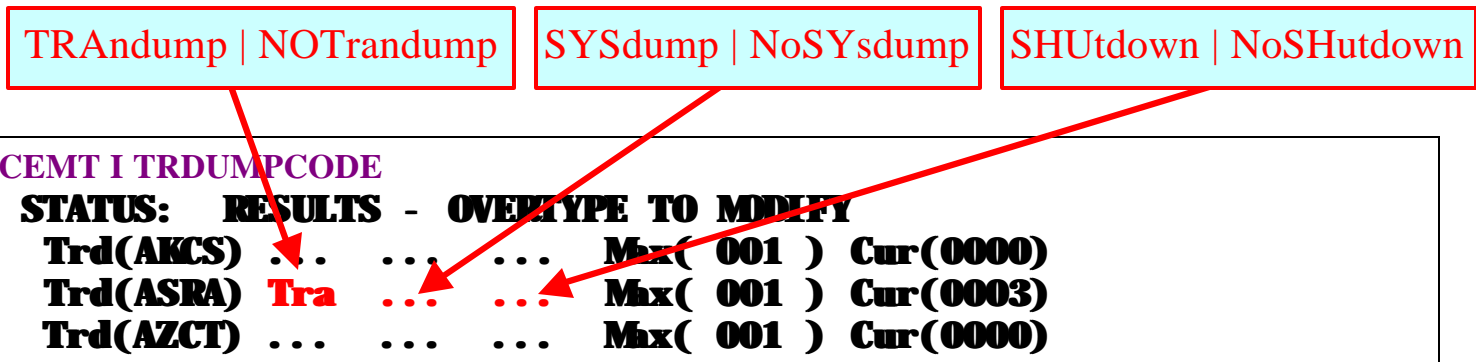
## CICS TS Dump Processing

A dump code is the CICS message number with the 'DFH' prefix removed.  
 For instance, 'AP0001' is the dump code for message DFHAP0001:

**F2 0103 DFHAP0001 DEDCCICS An abend (code 0C9/AKEA) has occurred at offset x'0000072E' in module CICSPGCK**

CEMT SET SYD(LD0107) MAX(1) SY ADD would add a new Dump Code to the table, and take a single system dump the next time message DFHLD0107I occurs.,,

4. Suppress or enable transaction dumps for individual transactions: using the DUMP attribute of the RDO TRANSACTION resource definition.
5. Suppress or enable transaction dumps for specific transaction abend codes: using Transaction Dump Code Table



“NOT”, “NSY” and “NSH” are not displayed on the dump code line, only “TRA”, “SYS” and “SHU”.

# CICS TS Transaction Abends (ASRA / AKEA)

## To Analyze a Transaction Dump:

1. Switch (or close) current dump dataset and print the transaction dump using DFHDU410.
2. **Abend details:** Printed at the front of the dump. There are two sets of PSW / Registers. The first set is that of the abend itself. The second is for the last EXEC CICS call from the transaction.
3. **Task Control Area (TCA):** This will also show failing transaction and program (phase).
4. **“\*EXC\*” entry in trace table:** Note what the transaction was doing just prior to abend.
5. **Compare TACB:** Follows the trace table. Compare with the PSW and registers printed at front of dump. If there are two TACBs, always use the second one.
6. **Locate failing code:** The PSW shows the location of the failure, and, the message **DFHAP0001** or **DFHSR0001** lists the displacement into the failing program (phase) or module. Scan for program name and **“PROGRAM STORAGE”**.
7. **Identify failing load module and displacement:** Use a Linkedit listing for the failing program.
8. **Identify the failing source code:** Use a current compile listing.

## **CICS TS Transaction Abends (ASRA / AKEA)**

The following sequence of events takes place when a transaction abend occurs:

1. Program Checks or VSE abends are detected by STXIT exits set by the CICS TS main task. The AP (Application) Domain is set ready-to-run.
2. AP Domain issues initial Program Check Recovery call to LE/VSE (if applicable)
3. AP Domain calls ME (Message) domain to issue message to SYSLOG
4. AP Domain calls DU (Dump) domain to save System Dump (if applicable)
5. AP Domain issues Rununit End Invocation call to LE/VSE (if applicable). LE/VSE trace output is written to SYSLST at this point. See entries written to DCT destination CESE.
6. AP Domain issues LINK-TO-USER-EXIT-PROGRAM IESOPIA. Saves OLPD data
7. AP Domain calls DU domain to save Transaction Dump (if applicable)
8. AP Domain releases transaction data areas and return to Kernel Domain.

## CICS TS Transaction Abends (ASRA / AKEA)

Four different methods are used to capture transaction abend data:

1. **System (full partition) dump** is taken first. Some control blocks (e.g. TACB) are not formatted yet, so this dump is more difficult to read, even when formatted by DFHPD410. It is large and slow, but offers the advantage that all data present in CICS TS at time of abend is saved. DFHPD410 can be run multiple times, extracting different kinds of data, or the entire dump can be printed using INFOANA.
2. **LE/VSE Trace** is written to destination CESE (defaults to SYSLST). Only contains LE/VSE control blocks. Traceback trace is useful to see sequence of LE/VSE calls leading up to the abend.
3. **On-line Problem Determination (OLPD)** is a very small snap shot of critical transaction abend information. It requires the Interactive Interface. The advantage is virtually no system overhead, and error data is presented in a very user-friendly, informative fashion. It is meant as a supplement, not a replacement for other error capture methods (transaction or system dump).
4. **Transaction dumps** are a small snapshot of AP, SM (Storage Management) and TR (Trace) control blocks. As such, the data is more oriented towards transaction rather than system diagnostics. The disadvantage is that it is a one-time shot; information not available in printout cannot be recovered.

## CICS TS Transaction Abends (ASRA / AKEA)

### Console Log:

```
F2 0103 DFHAP0001 DBDCCICS An abend (code 0C9/AKEA) has occurred at
offset x'0000072E' in module CICSPGCK.
F2 0103 DFHME0116 DBDCCICS
      (Module: DFHMEME) CICS symptom string for message DFHAP0001 is
      PIDS/564805400 LVLS/411 MS/DFHAP0001 RIDS/DFHSRP PTFS/VSE411
      AB/S00C9 AB/UAKEA RIDS/CICSPGCK ADRS/0000072E.
F2 0103 DFHDU0201 DBDCCICS ABOUT TO TAKE SDUMP. DUMPCODE: AP0001 ,
      DUMPID: 30/0001
F2 0103 OS24I AN SDUMP OR SDUMPX MACRO WAS ISSUED
F2 0103 OS29I DUMP STARTED
F2 0103 OS30I DUMP STARTED. MEMBER=DF200010. DUMP IN
F2 0103 1I51I DUMP COMPLETE
F2 0103 DFHDU0202 DBDCCICS SDUMPX COMPLETE. SDUMPX
```

**AKEA** is KE Domain transaction error code.  
**0C9** indicates a Fixed Point Divide Except.  
**ASRA** is the AP Domain error code.

# CICS TS Transaction Abends (ASRA / AKEA)

## 1. Print the transaction dump using DFHDU410:

```
CEMT SET DUMPDS SWITCH
STATUS: RESULTS - OVERTYPE TO MODIFY
Dum Cur(A) Ope Aut
```

At time of abend, dump dataset B was open. It is now closed, and A is open.

```
// JOB PRINT DUMP DATASET B
// LIBDEF *, SEARCH=(PRD1. BASE)
// DLBL DFHDUMP, ' CICS. DUMPB' , 0, VSAM, CAT=VSESPUC, DISP=(OLD, KEEP
// ASSGN SYS009, SYSLST (INDEX OUTPUT DESTINATION)
// EXEC DFHDU410, SIZE=2M, PARM=' TRANSLATE=LC, SINGLE, NOFULL'
// SELECT TYPE=SCAN
// END
/*
/ &
```

### INDEX OF DUMPS ON DATASET CICS. DUMPA

DUMPID	APPLID	DATE	TIME	TRANID	DUMP_CODE
30/0006	DBDCCICS	03182001	20:27	PGCK	<b>ASRA</b>

**AKEA** is KE Domain transaction error code.  
**0C9** indicates a Fixed Point Divide Except.  
**ASRA** is the AP Domain error code.

## CICS TS Transaction Abends (ASRA / AKEA)

The preceding job produced a list of all dumps in the selected dump dataset. They can then be selected based on Dump Code (ASRA), Transaction Id (PGCK), Dump Id (30/0006), or Time (20:27). Dump Id and Time may be specified as a range. We selected our dump based on Dump Code. Trace formatting is controlled via the execution parm (“NOABBREV” or “NOFULL”)

```
// JOB PRINT DUMP DATASET B
// LIBDEF *, SEARCH=(PRD1.BASE)
// DLBL DFHDUMP, 'CICS.DUMPB', 0, VSAM, CAT=VSESPUC, DISP=(OLD, KEEP
// ASSGN SYS009, SYSLST      (INDEX OUTPUT DESTINATION)
// EXEC DFHDU410, SIZE=2M, PARM='TRANSLATE=LC, SINGLE, NOFULL'
  SELECT TYPE=OR
  DUMPCODE=(ASRA)
  END
/*
/ &
```

See “[CICS TS Operations and Utilities Guide](#)” (SC33-1654-02)



# CICS TS Transaction Abends (ASRA / AKEA)

## 2. Abend details:

SYMPTOMS= AB **UASRA** PIDS/564805400 FLDS/DFHABAB RIDS/CICSPGCK  
CICS LEVEL = 0411

### PSW & REGISTERS AT TIME OF INTERRUPT

PSW	07BD2000	8241DEFE	00040009	00000000				
REGS 0- 7	0220A6BC	0220A730	00000000	000001F4	00000001	00000000	00614BBC	00000000
REGS 8- 15	006C10D0	0220A7C8	0241D91C	0241DBE0	0241D8E4	0220A578	8241DEEA	00000000

### REGISTERS AT LAST EXEC COMMAND

REGS 0- 7	00000000	0000006F	02209258	02374C7C	0220A578	00000000	02209A3C	0220D808
REGS 8- 15	0220CBA8	00000004	02209DD4	005F2750	022078B0	02209B98	00000001	02366EF0

Transaction environment for **transaction\_number(0000149)**  
**transaction\_id(PGCK)** orig\_transaction\_id(PGCK)  
initial\_program(CICSPGCK) **current\_program(CICSPGCK)**  
facility\_type(TERMINAL) facility\_name(A001) Start\_code(T0)  
netname(D0830001) profile\_name(DFHCICST)  
userid(SYSA ) cmdsec(N0) ressec(N0)

What kind of failure?

Which transaction failed?

Which transaction program?

# CICS/TS Transaction Abends (ASRA / AKEA)

## 3. Task Control Area (TCA):

### **TASK CONTROL AREA (at 005EB080)**

```

00000000 005BB180 00140001 021EA420 00589648 021623B0 005A25FC 00000000 00000060 * . $ . . . . . u . . . . . o . . . . . - *
00000020 00000000 0000149C 00000000 00000000 00000000 81C61614 021F2000 0000012C * . . . . . aF . . . . . *
.
.
.
00000140 02209B98 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . q . . . . . *
00000160 00000000 00000000 00C00000 00000000 00000000 00000000 80000000 C1E2D9C1 * . . . . . ASRA *
00000180 00000000 005BB4E4 00000000 00000000 005BB388 022078B0 006C1128 006C13E8 * . . . . . $ . U . . . . . $ . h . . . . . Y *
000001A0 00000000 80589080 00000000 00000000 D7C7C3D2 021EA420 00000000 00000000 * . . . . . PGCK . . . . . u . . . . . *
000001C0 00000000 D7C7C3D2 00000000 00000000 00000000 C1E2D9C1 00000000 021F9008 * . . . . . PGCK . . . . . ASRA . . . . . *
000001E0 00000000 00000000 00000000 00000000 B58DE2EE 70CEB042 00000000 00000000 * . . . . . S . . . . . *
00000200 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
00000220 00000000 00000000 00000000 00000000 80589400 00000000 00000000 005BB56C * . . . . . m . . . . . $ . . . . . *
00000240 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
00000260 00000000 00000000 170EE5E3 C1D4F14B C4C2C4C3 C3C9C3E2 8DE2EE70 CEB00001 * . . . . . VTAMI. DE DCCI CS. S. . . . . *
00000280 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
000002A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
000002C0 00000000 00000000 018A0023 006C1008 00000000 00000000 00000000 00000000 * . . . . . *
000002E0 00000000 00000000 C3C9C3E2 D7C7C3D2 F0C3F961 C1D2C5C1 0000072E 00030781 * . . . . . CICS/PGCK OC9/AKEA . . . . . a*
    
```

Which transaction failed?

What kind of failure?

Which transaction program?

## CICS TS Transaction Abends (ASRA / AKEA)

### 4. Locate “\*EXC\*” entry in trace table:

**= TRACE ENTRIES FOR DUMPING TRANSACTION =**

```

00149 1 US 0402 USXM EXIT INIT_TRANSACTION_USER/OK 01C4319F , 01C45050, 0           =000001=
00149 1 DS 0002 DSAT ENTRY SET_PRIORITY                1           =000002=
00149 1 DS 0003 DSAT EXIT SET_PRIORITY/OK             =000003=
00149 1 AP 05A0 APRM ENTRY TRANSACTION_INITIALIZATION =000004=
00149 1 AP 05A1 APRM EXIT TRANSACTION_INITIALIZATION/OK =000005=
.
.
.
00149 1 LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK      8241D7F0, 0241D7D0, 1C3A, 0, REUSABLE, ESDSA, OLD_COPY =000020=
.
.
.
00149 1 AP 00E1 EIP  ENTRY REWRITE                    0004, 0220A578 . . v. , 0B000606 . . . . =000093=
00149 1 AP 04E0 FCFR ENTRY REWRITE                    0218FA80 , 0000000C, 00000000, 0220A940, C8, NO, NO =000094=
00149 1 DS 0004 DSSR ENTRY WAIT_OLDW                  KSDS, FCIOWAIT, 0057E841, NO, IO =000095=
.
.
.
00149 1 AP 00E1 EIP  EXIT REWRITE                      OK                    00F4, 00000000 . . . . , 00000606 . . . . =000100=
00149 1 AP 1942 APLI *EXC* Program Check      START_PROGRAM CICSPGCK, CEIE, FULLAPI, EXEC, NO, 021F6030, 00000000 =000101=
00149 1 AP 1948 APLI EVENT CALL- TO- LE/VSE          Program_Check_Recovery CICSPGCK =000102=
00149 1 AP 1949 APLI EVENT RETURN- FROM LE/VSE      Program_Check_Recovery 00000004 CICSPGCK =000103=
00149 1 AP 0790 SRP  *EXC* PROGRAM_CHECK =000104=

```

# CICS TS Transaction Abends (ASRA / AKEA)

## 5. Compare the TACB:

== TRACE TABLE END ==

### STORAGE MANAGER TRANSACTION AREA (at 02161E18)

```
00000000 01BF718C 035DD880 836416D0 02161860 005BB34C 036403B8 005BB390 036403B0 *.....)Q.c.....-.$.<.....$......*
00000020 00000000 835FB080 0000050C 02161EC0 02161F08 *....c.....*
```

Compare this PSW and registers with those listed at front of dump

### TRANSACTION STORAGE-CICS31

ADDRESS 021F9000 TO 021F912F      LENGTH 00000130

```
00000000 C3F0F0F0 F0F1F4F9 01206EE3 C1C3C240 00000000 00001846 C4C6C8E3 C1C3C240 *C0000149..>TACB ..... DFHTACB *
00000020 00000000 C1E2D9C1 C3C9C3E2 D7C7C3E2 00000000 00000000 00000000 00000000 *... ASRA CICSPGCK ..... *
00000040 00000000 00000000 00000000 0000B000 00000000 0000072E D9C5C7E2 50D7E2E6 *..... REGS. PSW*
00000060 0220A6BC0 0220A7301 000000002 000001F43 000000014 000000005 00614BEC6 000000007 *..w...x.....4...../.....*
00000080 006C10D08 0220A7C89 0241D91CA 0241DEEDB 0241DEE4C 0220A578D 8241DEEAE 00000000F *....xH.R.....QU..v.b.....*
000000A0 07DE2000 8241DEEA 00040009 00000000 4931308E 88000000 4E000000 045D8AB5 *...b.....h...+....)..*
000000C0 4E000000 000254FB 40404040 40404040 00000000 00000000 00000000 00000000 *+..... ..*
000000E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
```

Failing PSW

Registers (0-15)

Abend Code

Transaction Program

# CICS TS Transaction Abends (ASRA / AKEA)

## 6. Locate failing code:

**CICS/PGCK**

**PROGRAM STORAGE ADDRESS 0241D7D0 TO 0241F409 LENGTH 00001C3A**

```

0241D7D0 -20 C406C8EB C94F1F1 58F00014 58F0F0B4 58F0F00C 58FF000C 07FF0000 00000000 *DFHI411 0...00...00.....
0241D7F0  0 47F0F028 00C3C5C5 00000000 00000014 47F0F001 4ACEAC00 0241D89C 00000000 *.00 CEE...
0241D810 20 00000000 00000000 90ECD00C 4110F038 98EFF04C 07FF0000 0241D7F0 0241D8E4 *.....
0241D830 40 0241E2A8 0241D850 0241D7F0 0241DC6A 0241E930 0241D8B0 00000000 00000007 *..Sy..Q...PC
...
0241D850 60 C306C2E3 C5E2E340 F2F0F0F1 F0F3F1F7 F1F8F5F8 F0F6F0F1 F0F1F0F1 00000000 *COBTEST 20010317185806010101.
0241D870 80 0000076C 60A87C4C 00000000 50800209 00000000 09000000 00800000 00000067 *....-y.<.....
...
0241D930 140 00000000 0050000A 00000000 00000000 00000000 00000000 00000000 *.....HShould not rece
0241D950 160 40A38889 A2408599 99999999 99999999 99999999 99999999 99999999 * this error (Illogic Error)
0241D970 180 81948540 9596A340 89999999 89999999 89999999 89999999 89999999 *ame not in Internet Table
...
0241DB60 370 96A49584 8199A840 A5899693 81A38584 E2404040 40404040 40404040 40E2E3E2 *oundary violated S
0241DB70 380 D6E4E340 40C306C2 E3C5E2E3 00000000 00000134 00000001 00000138 00000003 *OUT COBTEST.....
0241DB90 3A0 00000000 00000000 00000000 80000000 0409FF34 001C003C 0000005F 054A8132 *.....<....
...
0241DBD0 3E0 00300259 02C00007 08000030 02594040 5820D130 D2032000 A014D203 2008A014 *...
0241DBF0 400 D2032010 A014D207 2018A251 D2032020 A014920E 2028920F 2030D203 90C8C008 *K...
...
0241DEE0 6F0 4780B696 9180DOC4 4720B696 4110D12C 4100D144 58C0D080 05EF58C0 DOE850F0 *...
0241DEF0 700 D0785820 91188E20 002051E0 91205020 911058B0 C03447F0 B696D215 9128A1A4 *...
    
```

CICS TS Stub  
Call EIP to init  
apl control blks

LE/VSE  
Initialization  
Code

COBOL  
program name  
(Shows start of  
program)

Which machine  
instruction failed?  
PSW points past  
failing instruction

# CICS TS Transaction Abends (ASRA / AKEA)

Why did it fail?

```
TRANSACTION STORAGE-USER31      ADDRESS 02203A00 TO 0220C75F      LENGTH 00008C00
00000000 E4F0F0F0 F0F1F4F9 4C4CD9E4 E6D76E6E 00008C78 00000000 00008C78 00008C78 *U0000149<<RUWP>>..... *
00000020 00008C78 02203AE0 00010000 00000000 005A45671 125C005C 0000000C 00000000 *..... *
.
.
.
00006E00 99844B40 404040D7 9985A2A2 408595A3 859940A3 96408396 95A38995 A4854B40 *rd.   Press enter to continue. *
00006E20 60F44040 40000000 00000000 00000000 000001F4 00000000 00000000 00000000 *- 4   ..... *...4..... *
00006E40 D2C5E8F0 F0F0F0F0 F1F0F1F1 F1F1F1F1 40404040 40404040 40404040 40404040 *KEY00000101111111 *

```

Failing instruction is '5D209120' (Divide Reg2 by Reg9 +x'120'). From TACB, Reg9 = '0220A7C8'. The Transaction Dump printed this as transaction storage. Divisor is at '022A8E8', and contains zeros.

## CICS TS Transaction Abends (ASRA / AKEA)

Sometimes we need to locate another phase involved in the abend, but which may not have been included in the Transaction Dump printout.

### ----- MODULE INDEX -----

<u>LOAD PT.</u>	<u>NAME</u>	<u>ENTRY PT</u>	<u>LENGTH</u>	<u>LOAD PT.</u>	<u>NAME</u>	<u>ENTRY PT</u>	<u>LENGTH</u>
02056E60	DFHAPATT	02056E80	000002D2	02673750	DFHEDAD	02673770	0001D9B2
02057140	DFHZCSTP	02057160	0000027A	02692000	CEEMUEN3	02692000	00007C82
020573C0	DFHZCGRP	020573E0	00000412	02699C90	IGZCMGUE	02699C90	00004620
02058000	DFHPGRP	02058000	000032BA	035FB000	DFHSIP31	035FD090	000C36E0
.							
.							
.							
02402000	IESCONSL	02402000	0001B7C8				
<b>0241D7D0</b>	<b>CICSPGCK</b>	<b>0241D7F0</b>	<b>00001C3A</b>				
02500000	CEEEV003	025A3370	000EA7C8				
025EB000	DFHWBTC	025EB020	00013D30				

CICSPGCK is loaded at '**0241D7D0**'. The failing PSW was '**07BD2000 8241DEFE**'. Address portion is second word of PSW (x'80' indicates AMODE(31), so can be ignored). '**241DEFE**' – '**241D7D0**' = '**72E**' displacement into the phase, which was the same value as was listed in console message DFHAP0001.

# CICS TS Transaction Abends (ASRA / AKEA)

## 7. Identify the failing load module and displacement:

-----  
PHASE CICSPGCK, \*  
-----

INCLUDE DFHELII

\*\* MODULE DFHELII 2000-10-23 14.09 INCLUDED FROM PRD1 .BASE  
\*\* MODULE CEEBETBL 2000-10-23 14.01 AUTOLNKD FROM PRD2 .SCEEBASE  
\*\* MODULE CEEBINT 2000-10-23 14.01 AUTOLNKD FROM PRD2 .SCEEBASE

•  
•  
•

**03/19/2001 PHASE XFR-AD LOCORE HICORE CSECT/ LOADED RELOC. PARTT. PHASE**  
**ENTRY AT FACTOR OFFSET OFFSET**

-----  
CICSPGCK 500098 500078 501CB1  
-----

DFHELII 500078 500078 000000 000000  
\*DFHEPIN 500078  
\*DFHEXEC 500080  
+DFHEI 1 500080  
\*DLZEI 04 500080  
**COBTEST 500098 500098 000020 000020**  
CEEBETBL 500D70 500D70 000CF8 000CF8  
CEEBI NT 500D90 500D90 000D18 000D18

What load module failed?  
Displacement x'72E' is  
x'70E' into COBTEST.



# CICS TS Transaction Abends (ASRA / AKEA)

## 8. Identify the failing source code:

For this, you will need a COBOL compile listing with CBL options "LIST,XREF" specified.

```
// EXEC IGYCRCTL, SIZE=IGYCRCTL
```

```
PROCESS(CBL) STATEMENTS: CBL LIB, APOST, NOADV, NODYNAM, RENT, BUF(4096), LIST, XREF
```

```
000177      *EXEC CICS REWRITE FILE('KSDS') FROM(Console-Inarea)
000178      *          END-EXEC.
000179      MOVE 'ÊÊ\_____00074' TO DFHEIVO
000180      MOVE 'KSDS' TO DFHC0080
000181      MOVE LENGTH OF Console-Inarea TO DFHB0020
000182      CALL 'DFHEI1' USING DFHEIVO DFHC0080 Console-Inarea
000183                      DFHB0020.
000186      COMPUTE Result = Dividend / Divisor.
000188      GO TO EOJ.
```

000000		COBTEST	DS	0H	
			USING	*, 15	
000000	47F0 F028		B	40(, 15)	BYPASS CONSTANTS.
000004	00		DC	AL1(0)	ZERO NAME LENGTH FOR
<b>000005</b>	<b>C3C5C5</b>	LE/VSE Stub	<b>DC</b>	<b>CL3' CEE'</b>	<b>CEE EYE CATCHER</b>
000008	00000000		DC	F'0'	STACK FRAME SIZE
00000C	00000014		DC	A(@PPA1-COBTEST)	OFFSET TO PPA1 FROM Y
000010	47F0 F001		B	1(, 15)	RESERVED
000014		@PPA1	DS	0H	PPA1 STARTS HERE

## CICS TS Transaction Abends (ASRA / AKEA)

Address	@EPNAM	DS	OH	ENTRY POINT NAME
000060	<b>C3D6C2E3C5E2E340</b>	<b>DC</b>	<b>C' COBTEST '</b>	<b>PROGRAM NAME</b>
000068	F2F0F0F1	DC	CL4' 2001'	@TIMEVRS: YEAR OF COMPILATIO
00006C	F0F0F1F0	DC	CL4' 0319'	MONTH/DAY OF COMPILATI
000070		DC	CL4' 1612'	HOURS/MINUTES OF COMPI
000074	F4F5	DC	CL2' 45'	SECONDS FOR COMPILATIO
000076	F0F1F0F1F0F1	DC	CL6' 010101'	VERSION/RELEASE/MOD LE
0003F0		GN=2 EQU *		<b>Begin executable code</b>
<b>0003F0</b>	<b>5820 D130</b>	<b>L</b>	<b>2, 304(0, 13)</b>	<b>BL=1</b>
0003F4	D203 2000 A014	MVC	0(4, 2), 20(10)	SORT- CORE- SIZE
0003FA	D203 2008 A014	MVC	8(4, 2), 20(10)	SORT- FILE- SIZE
0006F8	05EF	BALR	14, 15	
0006FA	58C0 D0E8	L	12, 232(0, 13)	TCTFIXD+232
0006FE	50F0 D078	ST	1	TFIXD+120
000186	COMPUTE			
000702	5820 9118	L	2, 280(0, 9)	DI VI DEND
000706	8E20 0020	SRDA	2, 32(0)	
<b>00070A</b>	<b>5D20 9120</b>	<b>D</b>	<b>2, 288(0, 9)</b>	<b>DIVISOR</b>
00070E	5030 9110	ST	3, 272(0, 9)	RESULT
000188	GO			
000712	58B0 C034	L	11, 52(0, 12)	PBL=1
000716	47F0 B696	BC	15, 1686(0, 11)	E0J

Displacement x'70E' points just past divide instruction. COBOL source statement is #0186.

## CICS TS Transaction Abends (ASRA / AKEA)

### Summary:

1. Message DFHAP0001 identified an ASRA (program check) abend in CICSPGCK +x'072E'.
2. Transaction dump identified the current transaction as PGCK and transaction number as 0149.
3. From the trace table, prior to the failure, the program had just returned from File I/O, writing a record to VSAM file KSDS.
4. TACB matched the data in the front of the dump. Abend occurred at location x'00D7E2E6'. Reg9 is later found to be a pointer to program static storage at location x'0241D91C'.
5. From program storage, the failing instruction is identified as '5D209120' (Divide).
6. The program static storage addressed by register 9 is located in 31-bit transaction storage, and the divisor is determined to be zero, thus causing the program check.
7. Using a linkedit listing, and cobol compile listing, the failing source statement is identified.

## CICS TS Transaction Abends (ASRA / AKEA)

This page  
intentionally  
left blank

## CICS TS On-line Problem Determination (OLPD)

Immediately upon experiencing a program check the operator is presented with following screen:

**IESPRGDC1 ONLINE PROBLEM DETERMINATION DATA COLLECTION**

The transaction you were executing ended abnormally.

Information about this incident has been stored for later problem determination.

Specifics about this incident are:

Transaction ID:	<b>PGCK</b>	Abend Code:	<b>ASRA</b>
Task ID:	149	Abend Date:	3/18
Program ID:	<b>CICSPGCK</b>	Abend Time:	20:27:19

Press ENTER to return to your application.

===> PRESS ENTER TO CONTINUE.

What kind of failure?

Which program (phase) failed?

Which transaction failed?

Pressing enter returns to CICS screen:

DFHAC2206 20:28:01 DBDCCICS Transaction PGCK has failed with abend ASRA. Resource backout was successful.

# CICS TS On-line Problem Determination (OLPD)

Return to to IUI, and select '4.1 Online Problem Determination':

**IESPRBID1 ONLINE PROBLEM DETERMINATION INCIDENT REPORT**

P. NO. ENTER PARAGRAPH NUMBER TO GET ADDITIONAL INFORMATION

1 Transaction **PGCK** encountered a **ASRA** abend executing at **A001**. This occurred at offset X'00070E' in program **CICSPGCK**. Program 'CICSPGCK' starts at X'8241D7F0'. This incident occurred for task ID '00149' at 20:27:19 on 3/18.

2 The source language of the abending program is not defined correctly in the Processing Program Table, DFHPPT. Or the program has been link-edited incorrectly.

**3 This is a Fixed-Point Divide Exception program interrupt.**

PF1=HELP                      3=END                      4=RETURN                      6=PRINT  
8=FORWARD                      10=USER SCREEN

Why did it fail?

What kind of failure?

Which transaction failed?

Which program (phase) failed?

## CICS TS On-line Problem Determination (OLPD)

OLPD shows failing PSW and registers, excerpts from the dump around the abend location (but not the entire phase), and other problem related data (for instance, with a decimal divide exception, it shows the content of the divisor).

```

1      Program Check PSW ==> 07BD2000 8241DEFE

Registers in effect when the program check occurred:

REGS   0 - 3  0220A6BC 0220A730  00000000 000001F4
REGS   4 - 7  00000001 00000000  00614BBC 00000000
REGS   8 - 11 006C10D0 0220A7C8  0241D91C 0241DBE0
REGS  12 - 15 0241D8E4 0220A578  8241DEEA 00000000
    
```

```

1      The area of storage in the vicinity of the interrupt is
      shown below. This is at or near offset X'0000070A'
      within the application program. If the offset number
      shows question marks (?), the storage is not within the
      application program

      The failing instruction is at or near X'0241DEFA'.

Address  |<----- S T O R A G E ----->| Graphics
0241DED0 5030D1C4 9680D1C4 4110D1B8 58F0A000 &. JDo. JD.. J.. O..
0241DEE0 4100D144 58C0D080 05EF58C0 D0E850F0 .. J..... Y&O
0241DEF0 D0785820 91188E20 00205D20 91205030 ....j.....).j.&
0241DF00 911058B0 C03447F0 B696D215 9128A1A4 j..... O. oK. j.. u
    
```

Which machine instruction failed?

# CICS TS On-line Problem Determination (OLPD)

1 A Fixed-Point Divide Exception can occur during fixed-point division for either of two reasons.

**First check whether the divisor (the second operand) is zero. This is an invalid condition.**

The other reason for a Fixed-Point Divide Exception is that the quotient cannot be expressed as a 32-bit signed binary integer.

1 The storage area around the second operand of the failing instruction is shown below.

The second operand address is X'0220A8E8'.

Why did it fail?

Address	<----- S T O R A G E ----->	Graphics
0220A8D0	60F44040 40000000 00000000 00000000	-4 .....
0220A8E0	000001F4 00000000 <span style="border: 2px solid red; padding: 2px;">00000000</span> 00000000	... 4. ....
0220A8F0	D2C5E8F0 F0F0F0F0 F1F0F1F1 F1F1F1F1	KEY0000010111111
0220A900	E2C3C8D9 D6E3E340 404040D9 C5C3D6D9	SCHROTT RECOR



## CICS TS System Dump Processing (DFHPD410)

### Diagnose AKEA / ASRA Transaction Abends from System Dump:

1. **Format dump:** using `CALL DFHPD410 DATA DEF=0,KE=3,AP=3,TR=1`
2. **Locate failing transaction:** flagged with “**ERROR = \*YES\***” in KE Domain Task Summary and note the **KE NUM** (Kernel Number) and TCA Address. Then use **TRAN#** to find corresponding entry in AP Domain Transaction Summary and note the Terminal id (**FACILITY TYPE = “TC”**, **FACILITY ID = terminal id.**). **EIB** is also interesting, as it contains information on the last function requested by this transaction.
3. **KE Domain Error Table Summary:** locate entry matching task Kernel Number located in step 2.
4. **Kernel Error Data (KERRD):** Match sequence number of entry located in step 3.
5. **Locate failing instruction:** from PSW in KERRD and program dump in PSWDATA. Continue to isolate error to source statement, following step 6 on page 21.
6. **Register Data:** Depending on type of error, may either be found directly following the KERRD under REGDATA, or in USER31 / USER24 storage.
7. **“\*EXC\*” entry in trace table:** Note what the transaction was doing just prior to abend. If trace data table is long and complicated, you may print only the trace entries pertaining to this task (task id from the AP Domain Transaction summary) using:  
`CALL DFHPD410 DATA DEF=0,TR=1,TRS=<TASKID=(0269)>.`

# CICS TS System Dump Processing (DFHPD410)

## 1. Format Dump:

```
// JOB PRINT DUMP FROM SYSDUMP LIBRARY
// LIBDEF *, SEARCH=(PRD1. BASE)
// EXEC INFOANA, SIZE=300K
  DUMP NAME SYSDUMP. F2. DF200010
  SELECT DUMP SYMPTOMS
    PRINT DATA
    RETURN
  SELECT DUMP VIEWING
    CALL DFHPD410 DATA DEF=0, KE=3, AP=3, TR=1
    RETURN
  SELECT END
/*
/ &
```

## CICS TS System Dump Processing (DFHPD410)

DFHPD410 formats all control blocks required for problem determination, so the output can be quite large and complex. The control blocks are divided into **DOMAINS**, or discrete CICS functions. It is important to limit the amount of output by carefully selecting the domains (and level) to be formatted.

**DEF=0** Skip all domains not explicitly included

**AP={0|1|2|3}** Application Domain

**DS={0|1|2|3}** Dispatcher Domain

**FCP={0|2}** File Control Domain

**KE={0|1|2|3}** Kernel Domain

**LD={0|1|2|3}** (program) Loader Domain

**PCT={0|2}** Program Control Table

**PG={0|1|2|3}** Program Manager Domain

**IND={0|1|2|3}** Index of formatted control blocks

**SM={0|1|2|3}** Storage Manager Domain

**TCP={0|1|2|3}** Terminal Control Domain

**TR={0|1|2|3}** Trace Domain

**TRS={<PARMS>}** Select trace entries to be formatted

**TRS=<ENTRY\_NUM=(NN)>** Select one trace entry

**XM={0|1|2|3}** Transaction manager

**US={0|1}** User domain

# CICS TS System Dump Processing (DFHPD410)

## Interactive Interface to CICS TS Dump Analysis:

Select 4.3:

<b>PRB\$IDH1</b>		<b>STORAGE DUMP MANAGEMENT</b>					
<b>LIST OF SYSTEM MANAGED DUMPS</b>							
<b>OPTIONS: 2 = PRINT SYMPTOMS 3 = PRINT DUMP 4 = ANALYZE SA DUMP</b>							
<b>5 = DELETE DUMP 8 = ON/OFFLOAD DUMP 9 = ANALYZE CICS DUMP</b>							
<u>OPT</u>	<u>----- DUMP NAME-----</u>	<u>RELATED DUMP</u>	<u>ON-LINE</u>	<u>DATE</u>	<u>TIME</u>	<u>TAPE LABEL</u>	<u>DATA SPACE</u>
_	SYSDUMP. BG. DBG00002	NONE	X	03/14/01	18:13:13		
<u>9</u>	SYSDUMP. F2. DF200004	NONE	X	03/13/01	15:14:14		
_							
<b>PF1=HELP</b>		<b>2=REDISPLAY</b>	<b>3=END</b>	<b>5=PROCESS</b>		<b>6=ADD DUMP</b>	

## CICS TS System Dump Processing (DFHPD410)

In coexistence mode (if CICS/VSE is also installed), a second panel is presented to allow the selection of CICS Dump Formatting Tool: DFHPD410 (1) or DFHDAP (2):

**PRB\$IDHA**

**SPECIFY CICS VERSION**

Enter the required data and press ENTER.

Dump- Name: SYSDUMP. F3. DF300008

You have a coexistence system with CICS 2.3 and CICS/TS installed.

Please specify the CICS version with which the selected dump was produced.

CICS VERSION..... 1

Enter 1 if the dump was produced by CICS/TS, Enter 2 if the dump was produced by CICS/VSE.

PF2=REDISPLAY 3=END

# CICS TS System Dump Processing (DFHPD410)

~~PRBSIDH9~~

## ANALYZE CICS DUMP: SPECIFY DUMP LEVEL

Enter the required data and press ENTER.

Dump-Name: SYSDUMP. F2. DF200004

Enter the dump level for the following CICS component identifiers.

Enter 0, 1, 2 or 3

DS..... 3

Dispatcher domain

KE..... 3

CICS kernel

LD..... 3

Loader domain

TR..... 3

Trace domain

OTHER DUMP OPTIONS..... 2

Enter 1 to activate any other dump option, otherwise enter 2.

PF1=HELP

2=REDISPLAY

3=END

# CICS TS System Dump Processing (DFHPD410)

## 2. Locate failing transaction:

**CALL DFHPD410 DATA DEF=0, KE=3, AP=3, TR=1**

===KE: Kernel Domain KE\_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN #	TRANSID	DS	TASK	KE	KTCB	ERROR
0001	035F3880	KTCB Step	00000000				00000000	035D70B0		
0002	035F3500	KTCB QR	00000000				03505000	035D9020		
0003	035F3180	KTCB RO	00000000				03503000	035D8068		
...										
001F	02152B00	Unused								
<b>001F</b>	<b>0215F080</b>	<b>***Running**</b>	<b>005EB080</b>	<b>00149</b>	<b>PGCK</b>		<b>01BE8580</b>	<b>035D9020</b>	<b>*YES*</b>	
0020	0215F400	Unused								

Use KE\_NUM to locate entry in Domain Error table

TCA Address

===AP: AP DOMAIN TRANSACTION SUMMARY

Tran No	Tran Id	Orig Tran	TCA Addr	TVA Addr	EIB Addr	SEIB Addr	EIS Addr	EIUS Addr	Facility Type	Facility Id...
TCP	CSTP	CSTP	01C74680	0210A438	0210A0D0	01C74A8C	01C74988	0210A008		
00006	CSSY	CSSY	00596080	0053A000	0058E0D0	0059648C	00596388	0058E008		
00007	CSSY	CSSY	00596680	0053A000	0059E0D0	00596A8C	00596988	0059E008		
00020	CSSY	CSSY	005BB680	0053A000	005A80D0	005BBA8C	005BB988	005A8008		
00021	CSSY	CSSY	00597680	0053A000	005A60D0	00597A8C	00597988	005A6008		
00022	CSNE	CSNE	01C75080	0053A000	021190D0	01C7548C	01C75388	02119008		
00023	CXPB	CXPB	01C76680	0053A000	021830D0	01C76A8C	01C76988	02183008		
00025	IESO	IESO	00595080	006C0438	006C00D0	0059548C	00595388	006C0008		
<b>00149</b>	<b>PGCK</b>	<b>PGCK</b>	<b>005EB080</b>	<b>0053A000</b>	<b>006C10D0</b>	<b>005EB48C</b>	<b>005EB388</b>	<b>006C1008</b>	<b>TC</b>	<b>A001</b>

# CICS TS System Dump Processing (DFHPD410)

## 3. KE Domain Error Table Summary

**==KE: KE Domain Error Table Summary**

<u>ERR NUM</u>	<u>ERR TIME</u>	<u>KE NUM</u>	<u>ERROR TYPE</u>	<u>ERR CODE</u>	<u>MODULE</u>	<u>OFFSET</u>
00000001	20:11:24	0022	PROGRAM_CHECK	0C9/AKEA	DFHYI411	0000072E
00000002	20:11:45	0022	TRAN_ABEND_PERCOLATE	---/ASRA	DFHSR1	00000380
00000003	20:20:39	0022	PROGRAM_CHECK	0C9/AKEA	DFHYI411	0000072E
00000004	20:20:39	0022	TRAN_ABEND_PERCOLATE	---/ASRA	DFHSR1	00000380
<b>00000005</b>	<b>20:27:19</b>	<b>001F</b>	<b>PROGRAM_CHECK</b>	<b>0C9/AKEA</b>	<b>DFHYI411</b>	<b>0000072E</b>

## 4. Locate Kernel Error Data (KERRD):

**=KE: Error Number: 00000005**

KERRD 035F43C8 KERNEL ERROR DATA

TCA

ILC, Interrupt Code

000	F0C3F961	C1D2C5C1	010400C9	0000FFFF	C4C6C8C1	D7D3C9F1	01DA6D90	01BE8580	*OC9/AKEA...I...DFHAPLI1...e.*	
020	<b>0051B080</b>	0215F080	00000005	00000009	07BD0009	A0000000	<b>07BD2000</b>	<b>8241DEFE</b>	*\$....0.....b...*	
040	<b>0004</b>	<b>0009</b>	00000000	8241DEFE	B0000000	0220A6BC	0220A730	10000000	200001F4	*.....b.....w...x.....4*
060	00000001	00000000	00614BBC	00000000	006C10D0	0220A7C8	0241D91C	0241DBE0	*...../.....%...xH.R...*	
080	0241D8E4	0220A578	8241DEEA	00000000	00000000	00000000	00000000	00000000	*..QU..v.b.....*	
0A0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*	
0C0	00000000	00000000	00000000	00000000	07BD0009	A0000000	07BD2000	8241DEFE	*.....b...*	
0E0	00040009	00000000	8241DEFE	B0000000	0220A6BC	0220A730	00000000	000001F4	*.....b.....w...x.....4*	
100	00000001	00000000	00614BBC	00000000	006C10D0	0220A7C8	0241D91C	0241DBE0	*...../.....%...xH.R...*	
120	0241D8E4	0220A578	8241DEEA	00000000	00000000	00000000	00000000	00000000	*..QU..v.b.....*	
140	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*	
160	00000000	00000000	00000000	00000000	B58DE2F0	F9479E07	4931308E	88000000	*.....S09.....h...*	
180	4E000000	045D8AB5	4E000000	000254FB	40404040	40404040	00000000	00000000	*+....).+.....*	
1A0	00000000	00000000							*.....*	

PSW



# CICS TS System Dump Processing (DFHPD410)

## 5. Locate Failing Instruction:

### PSWATA 0241D7D0

```
0000 C4C6C8E8 C9F4F1F1 58F00014 58F0F0B4 58F0F00C 58F0000C 07FF0000 00000000 *DFHYI41I. 0...00. 00..... * 0241D7D0
0020 47F0F028 00C3C5C5 00000000 00000014 47F0F001 47CEAC00 0241D89C 00000000 *. 00. . CEE. .... 00. >..... * 0241D7F0
0040 00000000 00000000 90ECD00C 4110F038 98EFF04C 07FF0000 0241D7F0 0241D8E4 *..... 0. q. 0<..... * 0241D810
0060 0241E2A8 0241D850 0241D7F0 0241DC6A 0241E930 0241D8B0 00000000 00000007 *.. Sy.. Q& PO..... Z.. Q... * 0241D830
0080 C3D6C2E3 C5E2E340 F2F0F0F1 F0F3F1F7 F1F8F5F8 F0F6F0F1 F0F1F0F1 00000000 *COBTEST 200103171858060101* 0241D850
00A0 0000076C 60A87C4C 00000000 50800209 00000000 09000000 00800000 00000067 *... %-y@<.... &..... * 0241D870
.
.
.
06E0 92E8A020 412095C8 41309340 5020D1B8 5030D1BC 41209178 413092E8 5020D1C0 *kY... nH.. l & J. & J... j... * 0241DEB0
0700 5030D1C4 9680D1C4 4110D1B8 58F0A000 4100D144 58C0D080 05EF58C0 D0E850F0 *& JD. JD.. J.. 0... J..... * 0241DED0
0720 D0785820 91188E20 0020 51E0 9120 5030 9110B0 C03447F0 B696D215 9128A1A4 *.... j..... ). j. & j..... 0. o* 0241DF00
0740 9240913E D238913F 913E9208 D10CD160 4120B34C 5020D160 47F0B592 D203D160 *k j. K. j. j. K. J%J-... <& J-. 0* 0241DF10
```

## 6. Locate Register Data:

### REG 9 0220A7C8

31-bit data follows:

### REGDATA 0220A7C8

```
- 0080 0220A578 0220A790 0220A7C8 8220AAC0 C8C1D5C3 02205900 02205900 00000000 *.. v... x... xHb... HANC..... * 0220A7C8
- 0060 8220A758 0220ADE8 00001000 00000970 0220A758 00000670 00000668 00000000 *b. x... Y..... x..... * 0220A7E8
- 0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... * 0220A808
- 0020 E2E8E2D6 E4E34040 00000000 00000000 0E000000 00000000 0F000000 00000000 *SYSOUT ..... * 0220A828
0000 D2C5E8F0 F0F0F0F0 F1F0F1F1 F1F1F1F1 E2C3C8D9 D6E3E340 404040D9 C5C3D6D9 *KEY0000010111111SCHROTT * 0220A848
0020 C440F0F0 F0F04040 40404040 40404040 40404040 40404040 40404040 40404040 *D 0000 * 0220A868
0040 40404040 40404040 40404040 40404040 00000000 00000000 00000000 00000000 * ..... * 0220A888
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... * 0220A8A8
0080 00BF LINES SAME AS ABOVE *..... * 0220A8C8
00C0 00000000 00000000 00000001 00000000 D9858184 A840A396 40998581 84408689 *..... Ready to re* 0220A8E8
00E0 99A2A340 99858396 99844B40 404040D7 9985A2A2 408595A3 859940A3 96408396 *rst record. Press enter * 0220A908
```

# CICS TS System Dump Processing (DFHPD410)

## USER31.00149 02203AB0 USER storage above 16MB

```

0000 E4F0F0F0 F0F1F4F9 4C4CD9E4 E6D76E6E 00008C78 00000000 00008C78 00008C78 *U0000149<<RUWP>>..... * 02203AB0
0020 00008C78 02203AE0 00010000 00000000 4C4CD9E4 E6C26E6E E4F0F0F0 F0F1F1F1 *..... <<RUWB>>U0000111 * 02203AD0
0040 E4F0F0F0 F0F1F1F1 00000000 00000000 00000000 00000000 00000000 00000000 *U0000111..... * 02203AF0
.
.
.
6D00 00000000 00000000 0E000000 00000000 0F000000 00000000 D2C5E8F0 F0F0F0F0 *.....KEY00000* 0220A7B0
6D20 F1F0F1F1 F1F1F1F1 E2C3C8D9 D6E3E340 404040D9 C5C3D6D9 C440F0F0 F0F04040 *10111111SCHROTT RECORD 0* 0220A7D0
6D40 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * * 0220A7F0
6D60 40404040 40404040 00000000 00000000 00000000 00000000 00000000 00000000 *..... * 0220A810
6D80 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... * 0220A830
6DA0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... * 0220A850
6DC0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... * 0220A870
6DE0 00000001 00000000 D9858184 A840A396 40998581 84408689 99A2A340 99858396 *.....Ready to read first * 0220A890
6E00 99844B40 404040D7 9985A2A2 408595A3 859940A3 96408396 95A38995 A4854B40 *rd. Press enter to conti * 0220A8B0
6E20 60F44040 40000000 00000000 00000000 000001F4 00000000 00000000 *-4 .....4..... * 0220A8D0
6E40 D2C5E8F0 F0F0F0F0 F1F0F1F1 F1F1F1F1 E2C3C8D9 D6E3E340 404040D9 C5C3D6D9 *KEY0000010111111SCHROTT * 0220A8F0
6E60 C440F0F0 F0F04040 40404040 40404040 40404040 40404040 40404040 40404040 *D 0000 * 0220A910

```

Divisor (Reg9 +x'120')

DFHPD410 dumps x'180' bytes for each registers (x'80 prior and x'100' after address in register)  
Unfortunately, in this case, we need Reg9 + x'120'. For this we look for this address in User31 storage. Search for the address in the far-right column.

# CICS TS System Dump Processing (DFHPD410)

## ==AP: AP DIMAIN TRANSACTION CONTROL BLOCKS

### TCA.00149 005EB080 Task Control Area (User Area)

```

0000 005EB180 00140001 021EA420 00589648 02161010 005A25FC 00000000 00000008 *$. . . . . u. . . . . !. . . . . *
0020 00000000 00001490 00000000 00000000 00000000 81F84278 021F2000 0000012C * . . . . . a8. . . . . *
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
0060 00000081 01004000 00000040 021EA420 00000000 00000000 00000000 00000000 * . . . . . u. . . . . *
0080 00000019 021EA420 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . u. . . . . *
00A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
00C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 80589DC0 * . . . . . *
00E0 00000000 00000000 00000000 00000000 0053A000 00000000 00000000 00000000 02203AB8 * . . . . . *
    
```

### SYS-TCA.00149 005EB180 Task Control Area (System Area)

```

0000 00000000 00000000 00000000 00000000 0000149C 01BF050C 0000004A 00000000 * . . . . . > . . . . . *
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
0040 0220A578 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * v. . . . . *
0060 00000000 00000000 00C00000 00000000 00000000 00000000 80000000 00000000 * . . . . . *
0080 00000000 005BB4E4 00000000 00000000 005BB388 022078B0 006C1128 006C13E8 * . . . . $ U. . . . . S. h. . . . % . . . % Y*
00A0 00000000 80589080 00000000 00000000 D7C7C3D2 021EA420 00000000 00000000 * . . . . . PGCK. u. . . . . *
00C0 00000000 D7C7C3D2 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . PGCK. . . . . *
00E0 00000000 00000000 00000000 00000000 B58DE2EE 70CEB042 00000000 00000000 * . . . . . S. . . . . *
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
0120 00000000 00000000 00000000 00000000 80589400 00000000 00000000 005BB56C * . . . . . m . . . . . $ %*
0140 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
0160 00000000 00000000 170EE5E3 C1D4F14B C4C2C4C3 C3C9C3E2 8DE2EE70 CEB00001 * . . . . . VTAM. DBDCCI. S. . . . . *
0180 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
01A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * . . . . . *
01C0 00000000 00000000 018A0023 006C1008 00000000 00000000 00000000 00000000 * . . . . . % . . . . . *
01E0 00000000 00000000 C3C9C3E2 D7C7C3D2 F0C3F961 C1DE5C1 0000072E 08030781 * . . . . . CICS PGCK. OC9/AREA . . . . a*
0200 00000000 00000000 . . . . . *
    
```

## CICS TS System Dump Processing (DFHPD410)

In a Formatted Transaction Dump, the TCA was presented as a single block, followed by a repeat of the System Area. Actually, the User Area, TCA(U), is only x'100' bytes long, followed by the system area, TCA(S). The first full word in TCA(U) points at TCA(S), but since, under CICS TS, TCA(U) is always x'100' long, the TCA(S) isn't hard to find.

In the Formatted Transaction Dump we saw earlier, TCA(S) +x'D4' contained the transaction abend code ('ASRA'). Here we see evidence that the System dump is taken prior to the Transaction dump, since this field is not yet filled in. However, TCA(S) +x'1F0' contains the Kernel Domain trans abend code ('0C9/AKEA').

### 7. Locate Exception (\*EXC\*) Trace Entry:

**CALL DFHPD410 DATA DEF=0, TR=1, TRS=<TASKID=(0149)>**

<b>00149</b>	<b>1</b>	<b>AP</b>	<b>00E1</b>	<b>EIP</b>	<b>ENTRY REWRITE</b>	<b>0004, 0220A578 .. v. , 0E000606</b>	<b>=000645=</b>
00149	1	AP	04E0	FCFR	ENTRY REWRITE	0218FA80 , 0000000C, 00000000, 0220A940, C8, NO, NO	=000646=
00149	1	DS	0004	DSSR	ENTRY WAIT_OLDW	KSDS, FCIOWAIT, 0057E841, NO, IO	=000647=
00149	1	DS	0005	DSSR	EXIT WAIT_OLDW/OK		=000648=
00149	1	AP	FO0A	XCPC	ENTRY DEQ		=000649=
00149	1	AP	FO0B	XCPC	EXIT DEQ		=000650=
00149	1	AP	04E1	FCFR	EXIT REWRITE/OK	00000000, LENGTH_OK	=000651=
<b>00149</b>	<b>1</b>	<b>AP</b>	<b>00E1</b>	<b>EIP</b>	<b>EXIT REWRITE</b>	<b>OK</b>	<b>00F4, 00000000 .. . , 00000606 =000652=</b>
<b>00149</b>	<b>1</b>	<b>AP</b>	<b>1942</b>	<b>APLI</b>	<b>*EXC* Program Check</b>	<b>START PROGRAM, CICSPGCK, CEIE, FULLAPI, EXEC, NO, 021F6030, 0000, =000653=</b>	
00149	1	AP	1948	APLI	EVENT CALL- TO- LE/VSE	Program_Check_Recovery CICSPGCK	=000654=
00149	1	AP	1949	APLI	EVENT RETURN- FROM LE/VSE	Program_Check_Recovery 00000004 CICSPGCK	=000655=
00149	1	AP	0790	SRP	*EXC* PROGRAM_CHECK		=000656=

# CICS TS System Dump Processing (DFHPD410)

Now, let's say we wanted more detail on the **EIP ENTRY REWRITE** and the Exception Entry (Trace entries 645 and 653):

**CALL DFHPD410 DATA DEF=0, TR=3, TRS=<ENTRY\_NUM=(645, 653)**

```

AP 00E1 EIP ENTRY REWRITE                                REQ(0004) FIELD-A(0220A578 .. v.) FIELD-B(0B000606 ....)
                TASK-00149 KE_NUM-001F TCB-00342000 RET-8241DEEA TIME-20:27:19.3500194704 INTERVAL-00.0000021093 =000645=

AP 1942 APLI *EXC* - Program Check  FUNCTION(START_PROGRAM) PROGRAM(CICSPGCK) CEDF_STATUS(CEDF) EXECUTION_SET(FULL
                ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(021F6030) COMMAREA(00000000 , 00000000)
                LINK_LEVEL(1) SYSEIB_REQUEST(NO)
                TASK-00149 KE_NUM-001F TCB-00342000 RET-81D3FC8C TIME-20:27:19.3514947204 INTERVAL-00.0001222187 =000653=
    
```

Program Block

```

1-0000 00580000 000000DA 00000000 00000000 B81B5D40 00000000 02680100 C3C9C3E2 *.....) .. CICS*
0020 D7C7C3D2 01CB3AB0 B5000000 01CB3AB0 01000001 01410202 0241D7D0 021F6030 *PGCK..... P...-*
0040 00001C3A 00000000 00000000 00000000 00010002 02020000 *.....*
    
```

KERRD

```

2-0000 FOC3F961 C1DE5C1 010400C9 0000FFFF C4C6C8C1 D7D3C9F1 01DA6D90 01BE8580 *OC9/AMEA... I... DFHAPLI1.*
0020 005BB080 0215F080 00000005 00000009 07BD0009 A0000000 07BD2000 8241DEFE *$. . . 0. . . . . b...*
0040 0004 0009 00000000 8241DEFE B0000000 0220A6BC 0220A730 00000000 000001F43 *..... b..... w... x... 4*
0060 00000001 40000000 500614BBC 60000000 7006C10D0 80220A7C8 80241D91C A0241DBE0 B ..... /..... PSW H.
0080 0241D8E4 C0220A578 D8241DEEA E0000000 F 00000000 00000000 00000000 00000000 *.. QU.. v. b.....*
00A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
00C0 00000000 00000000 00000000 00000000 07BD0009 A0000000 07BD2000 8241DEFE *..... b...*
    
```

# CICS TS System Dump Processing (DFHPD410)

This page  
intentionally  
left blank

## CICS TS Transaction Abends (AKEB)

### To diagnose AKEB Transaction Abends:

```
L1 0107 DFHSR0606 A01CICS5 Abend (code 2C5/AKEB) has been detected.
L1 0107 DFHME0116 A01CICS5
      (Module: DFHMEME) CICS symptom string for message DFHSR0606 is
      PIDS/564805400 LVLS/410 MS/DFHSR0606 RIDS/DFHSRP PTFS/VSE410
      AB/S02C5 AB/UAKEB.
L1 0107 DFHDU0201 A01CICS5 ABOUT TO TAKE SDUMP. DUMPCODE: SR0606 ,
      DUMPID: 1/0002
L1 0107 OS24I AN SDUMP OR SDUMPX MACRO WAS ISSUED
L1 0107 OS29I DUMP STARTED
L1 0107 OS30I DUMP STARTED. MEMBER=DL100008. DUMP IN SUBLIB=SYSDUMP. L1
L1 0107 1I51I DUMP COMPLETE
L1 0107 DFHDU0202 A01CICS5 SDUMPX COMPLETE. SDUMPX RETURN CODE X'00'
```

1. Format dump: Use “CALL DFHPD410 DATA DEF=0,KE=3,AP=3,TR=1”.
2. KE Domain Error Table Summary :
3. Kernel Error Data (KERRD):
4. PSW:
5. Register Data:
6. “\*EXC\*” entry in trace table:
7. Exec Interface Block (EIB):
8. User Savearea:

# CICS TS Transaction Abends (AKEB)

## Speaker Notes:

1. Format dump: Use “CALL DFHPD410 DATA DEF=0,KE=3,AP=3,TR=1”. Add “FCP=3”, if an error in file access. Check Cancel Code in error message on console. ‘2C5’ is a special return code for OS/390 simulation mode, and give us a reason code, which is saved in the KERRD+x’2C’. We’ll check this later when we get to the KERRD. Locate failing transaction in KE Domain Task Summary.
2. KE Domain Error Table Summary : Locate entry matching task located in step 1.
3. Kernel Error Data (KERRD): Locate entry matching sequence number of entry from step 2.
4. PSW: Locate failing instruction from PSW in KERRD and program dump in PSWDATA. Usually with AKEB abends, the PSW is not pointing at a user program. To identify the failing module, use INFOANA to print off the segment of the dump surrounding the PSW and look for an eyecatcher (see page 78). In this case, the failing module was in the SVA, so is not included in the CICS TS System Dump.
5. Register Data: x’100’ bytes of register data may be found directly following the KERRD under REGDATA. If we need to see more than this, register data is also usually found in USER31 / USER24 storage.
6. “\*EXC\*” entry in trace table: Helps us understand what the transaction was doing just prior to abend. Unfortunately, in this case, the user did not have INTTR turned on, so all we have is the \*EXC\* entry. Requesting a full trace entry didn’t show us much more. Trace data is the KERRD.

Since we don’t have a trace table, we don’t know what the transaction was doing which precipitated the AKEB abend. The System EIB will tell us what CICS TS command was requested, and the User Savearea will help us locate the failing source statement.

7. Exec Interface Block (EIB): Displacement x’1B’ contains the two-digit Function Code (EIBFN) which tells us what function the failing application requested. For a failed request, displacement x’1D’ contains a six-digit response code (EIBRCODE). See “*CICS TS User’s Guide*” (SX33-6101-00). In this case, x’0602’ indicates “Read File”. For File Access Request, displacement x’23’ contains the dataset name (EIBDS).
8. User Savearea: x’40’ into System TCA is a pointer to the User Savearea. This points to the last CICS TS Command Level Language call. This is probably the call which resulted in the AKEB. This will not be formatted, but, since the address is <16Meg, it can be located in a UDSA storage area. Search for “USER24.NNNNN”, where “nnnnn” is the transaction id, then locate the address in the right-hand column. Continue to isolate error to source statement, following step 6 on page 21.



# CICS TS Transaction Abends (AKEB)

## 1. Locate failing transaction in KE Domain Task Summary.

**CALL DFHPD410 DATA DEF=0, KE=3, AP=3, FCP=3, TR=1**

===**KE: Kernel Domain KE\_TASK Summary**

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB	ERROR
0001	036F6880	KTCB Step	00000000			00000000	036DA0B0	
.	.	.	.	.	.	.	.	.
003C	0265A080	Unused						
<b>003D</b>	<b>0265A400</b>	<b>***Running**</b>	<b>00722080</b>	<b>00035</b>	<b>PRNT</b>	<b>03629180</b>	<b>036DC020</b>	<b>*YES*</b>
003E	0265A780	Unused						
0040	022AF080	Not Running	00721080	00006	CSSY	0363E680	036DC020	
0041	022AF400	Not Running	00721680	00007	CSSY	0363EB80	036DC020	
0042	022AF780	Not Running	007D7080	00022	CSSY	0363ED80	036DC020	

===**AP: AP DOMAIN TRANSACTION SUMMARY**

Tran No	Tran Id	Orig Tran	TCA Addr	TWA Addr	EIB Addr	SEIB Addr	Facility Type	Facility Id
TCP	CSTP	CSTP	02268680	022AE438	022AE0D0	02268A8C		
00006	CSSY	CSSY	00721080	006BA000	0072C0D0	0072148C		
00007	CSSY	CSSY	00721680	006BA000	0072D0D0	00721A8C		
JBS	CSSY	CSSY	007D7680	006BA000	007D60D0	007D7A8C		
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
<b>00035</b>	<b>PRNT</b>	<b>PRNT</b>	<b>00722080</b>	<b>006BA000</b>	<b>008010D0</b>	<b>0072248C</b>	<b>C</b>	<b>T01A</b>

# CICS TS Transaction Abends (AKEB)

## 2. Locate KE Domain Error Table Summary:

==KE: KE Domain Error Table Summary

ERR_NUM	ERR_TIME	KE_NUM	ERROR TYPE	ERR_CODE	MODULE	OFFSET
00000001	17: 43: 23	0004	TRAN_ABEND_PERCOLATE	--- /AEI 1	DFHPCP	000004EC
00000002	17: 43: 23	0004	TRAN_ABEND_PERCOLATE	--- /AEI 1	DFHEIP	000015D6
<b>00000003</b>	<b>17: 44: 22</b>	<b>003D</b>	<b>ABEND</b>	<b>2C5/AKEB</b>	<b>UNKNOWN</b>	<b>UNKNOWN</b>

## 3. Locate KE Error Data (KERRD):

==KE: Error Number: **00000003**

ILC, Interrupt Code

Reason Code

PSW

VSE Cancel Code ('2C5')

KERRD 0265A5D8 KERNEL ERROR DATA

0000	F2C3F561	C1D2C5C2	020402C5	0000FFFF	C4C6C8C6	C3C2C440	00767420	03629180	*2C5/AKEB...E...DFHFCBD.....j.*
0020	00722080	0265A400	00000003	<b>42000000</b>	07DD0000	4017367E	<b>07DD0000</b>	<b>0017367E</b>	*.....u.....=.*=
0040	<b>0002</b>	<b>0000</b>	00000000	0017367E	D0000000	00000005	0E7DA000	006F0608	*.....PSW is pointing outside CICS TS partition.*
0060	006F0728	077DA108	007DA078	006F0610	A0173656	FFFFFFFF	0000000B	00000001	*?...*
0080	0000000F	006F06B8	006F0600	00173500	00000000	00000000	00000000	00000000	*.....*
00A0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
00C0	00000000	00000000	00000000	00000000	07DD0000	4017367E	07DD0000	0017367E	*.....*
00E0	00020000	00000000	0017367E	D0000000	00000005	0E7DA000	006F0608	00000000	*.....*
0100	006F0728	077DA108	007DA078	006F0610	A0173656	FFFFFFFF	0000000B	00000001	*.....*
0120	0000000F	006F06B8	006F0600	00173500	00000000	00000000	00000000	00000000	*.....*
0140	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
0160	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
0180	40404040	40404040	40404040	40404040	40404040	40404040	00000000	00000000	*.....*
01A0	00000000	00000000							*.....*

ILC = 2. Usually indicates an SVC (0Axx)  
 Interrupt Code = '0000' would indicate SVC 00 (EXCP)

## CICS TS Transaction Abends (AKEB)

Error Code: 2C5/AKEB    Error Type: ABEND    Timestamp: B592C5FADC5D7600

Date (GMT) : 22/03/01    Time (GMT) : 17:44:22.474199  
Date (LOCAL) : 22/03/01    Time (LOCAL) : 17:44:22.474199

KE\_NUM 003D    KE\_TASK: 0265A400    TCA\_ADDR: 00722080    DS\_TASK: 03629180

**Program DFHCED was in control, but the PSW was elsewhere.**

Registers and PSW

PSW 07DD0000 0017367E    Instruction Length: 2    Interrupt Code: 00    Exception Address: 00000000

Execution key at Program Check/Abend: D

## CICS TS Transaction Abends (AKEB)

### 17.3 OS/390 API Abend Codes

<u>Abend Code</u>	<u>Reason Code</u>	<u>Subreason Code</u>	<u>Explanation</u>
⋮			
2C5			This ABEND code is reserved for VSE specific error situations. VSE cancel codes are reflected by reason codes in the range from X'01000000' to X'FFFFFFFF'.
⋮			
	cc00000		cc is the VSE cancel code as described in VSE/ESA Messages and Codes Volume 1.

### 17.2 VSE/Advanced Functions Cancel Codes

<u>Cancel Code</u>	<u>Associated Message</u>	<u>Reason for Cancellation</u>
⋮		
42	OP86	DASD file-protection exception: the failing program attempted to access a location beyond extent limits.

# CICS TS Transaction Abends (AKEB)

## 4. Locate Failing instruction:

Data at PSW 0017367E    Module: UNKNOW    Offset: UNKNOW

PSWATA 0017367E

- 0080	4710F106	41440001	9602102C	4580F298	1B441B33	4330102D	4343E174	4A40F7A4	*..1.....o.....2q.....>7u*
- 0060	1A4E9500	102D4780	F1309502	102D4770	F1529140	102047E0	F152BFOC	1076BF02	*.+n.....1.n.....1.j....1.....*
- 0040	106F587E	0000BF01	7006BF18	101D0A4B	BE01E102	4580F298	9104102D	4710F336	*.?.=.....2qj.....3.*
- 0020	91101010	4780F17C	9102102D	4710F17C	9104102D	4710F17C	0A2347F0	F152E040	*j.....1@j.....1@j.....1@...01=..*
0000	91101010	4780F190	9505102D	4770F190	0A249101	102C4710	F40C981E	D00007FE	*j.....1.n.....1..j.....4.q.....*
0020	1B33D201	108EF6EC	47F0F0DA	95001028	4780F27C	1B664360	10288960	00011A61	*..K..6..00.n.....2@...-..i-.../*
0040	58206010	91021015	4780F1E6	F2776018	20004FA0	6018F271	60182008	4FC06018	*...j.....1W2.-... .-.2.-... .-.*
0060	42C0600F	47F0F1F6	D2026019	2000D200	600F2003	58A06018	1BCC4120	602018BA	*...016K.-...K.-.....-.....*

## CICS TS Transaction Abends (AKEB)

The amount of storage printed as “PSW DATA” does not include an eyecatcher to identify the failing module. Attempt to print a range of addresses around the PSW address from the dump the CICS TS System Dump.

```

// EXEC INFOANA, SIZE=300K
DUMP NAME SYSDUMP. F2. CICSTSAB
SELECT DUMP VIEWING
PRINT 173500 174000
RETURN
SELECT END
    
```

```

00173500 47FOF6FC 47FOF3E8 9701102D 9707102D 9701102D 9703102D 9701102D 47000000 *.06.03Yp...p...p...p.....*
00173520 47FOF080 C9D1C7E7 C4C1C640 47FOF6F4 F1F5C340 F5F6F8F6 60FOF6F6 404DC35D *IJGXDAF.06415C 5686-066 (C)*
00173540 40C3D6D7 E8D9C9C7 C8E340C9 C2D440C3 D6D9D740 F1F9F8F0 6BF1F9F9 F540D3C9 *COPYRIGHT IBM CORP 1980,1995 LI*
00173560 C3C5D5E2 C5C440D4 C1E3C5D9 C9C1D3E2 60D7D9D6 D7C5D9E3 E840D6C6 40C9C2D4 *CENSED MATERIALS-PROPERTY OF IBM*
00173580 58110008 50D100E8 41D100B8 58110184 50ED0034 58E10008 901CD000 4870102E *...&J. Y. J....d&.....*
001735A0 4A70F7A6 1A715877 00005077 C9400 10589504 102D4780 *..7w.....&=...'..m..m..n....*
001735C0 FOEA9101 10154710 F19E4833 E4030 108E4133 00089110 *0.j.....1....<.....>.....j.*
001735E0 10154780 FOEA4030 10AE4580 E4140 F79F9140 10204710 *....0. ....1.j.....1..7.j....*
00173600 F1064144 00019602 102C4580 102D4343 E1744A40 F7A41A4E *1....o.....2q.....7u.+*
00173620 9500102D 4780F130 9502102D 4770F152 91401020 47E0F152 BF0C1076 BF02106F *n....1.n....1.j ....1.....?*
00173640 587E0000 BF017006 BF18101D 0A4BBE01 E1024580 F2989104 102D4710 F3369110 *.=.....2qj.....3.j.*
00173660 10104780 F17C9102 102D4710 F17C9104 102D4710 F17C0A23 47FOF17E 0A00110 *....1@j.....1@j.....1@...01=..j.*
00173680 10104780 F1909505 102D4770 F1900A24 9101102C 4710F40C 981ED000 07FE1B33 *....1.n....1...j.....4.q.....*
001736A0 D201108E F6EC47F0 F0DA9500 10284780 F27C1B66 43601028 89600001 1A615820 *K..6..00.n....2@...-..i-.../.*
    
```

Failing Instruction  
IJGXDAF+x'15C'

# CICS TS Transaction Abends (AKEB)

## 5. Examine Register Data:

REG 1 0E7DA000

31-bit data cannot be accessed \*\*

24-bit data follows:

REGDATA 007DA000

```
-0080 03628D7C 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *...@.....*
```

```
-0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
```

```
-0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
```

```
0000 00008000 0000134 006F0600 00000000 00173500 220AC4D9 D7D9E3C4 40E0134 *...?.....DRPRTD*
```

```
0020 41000000 00000000 00707032 00000078 00008800 00000000 007DA040 00000000 *...h.....*
```

```
0040 077DA109 00000006 97602024 0FB28F00 93000000 0000003D 00000000 00000000 *...'p...l.....*
```

```
0060 317DA0EB 40000005 00000000 00000000 06707080 40000FB2 127DA103 40000005 *...'.....'g.p.g.j.....*
```

```
0080 397DA0EB 40000004 0E707080 40000FB2 8718979E 8718919E 077DA109 40000006 *...'.....'g.p.g.j.....*
```

```
00A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
```

```
00C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
```

```
00E0 00000000 00000000 00000000 01000B07 00707028 00707028 00000000 0000007F *.....**
```

For SVC 00 errors, check out register 1.  
In this case, it is pointing to a DTF: **DRPRTD**

The CCB (first 16 bytes of DTF) does not show a two-byte I/O (CSW) completion code (displacement x'04'). This implies that the error was recognized by the supervisor, not the I/O device. If "DASDFP=YES" is specified, this is probably a DASD File Protect error, which prohibits programs from accessing or changing data outside the defined extents for the file.

# CICS TS Transaction Abends (AKEB)

## 6. Locate \*EXC\* entry in trace table:

Unfortunately, INTTR=OFF  
Detail for SRP \*EXC\* is the  
KERRD (see page 40)

### INTERNAL TRACE TABLE

DSTCB	1	KE	0201	KEDD	ENTRY SET_GLOBAL_TRACE	OFF	=000001=
DSTCB	1	KE	0202	KEDD	EXIT SET_GLOBAL_TRACE/OK		=000002=
00001	2	TR	0230	TRSR	EVENT TIME-BASE-RESET		=000003=
00014	2	ST	0059	STST	*EXC* DMF	ERROR.	=000004=
III	2	ST	0059	STST	*EXC* DMF	ERROR.	=000005=
00033	2	AP	EA01	TMP	EXIT LOCATE	DSN, PRY2, VERSION2, YEARLY, PRINT, CAPTURE, 0280A490, NORMAL	=000116=
00033	2	AP	EA01	TMP	EXIT UNLOCK	DSN, PRY2, VERSION2, YEARLY, PRINT, CAPTURE, NORMAL	=000117=
<b>00035</b>	<b>2</b>	<b>AP</b>	<b>0791</b>	<b>SRP</b>	<b>*EXC* VSE ABEND</b>		<b>=000118=</b>

## 6. Locate EIB and extract EIBFN:

EIBFN = '0602': File Read from DRPRTD

### SYSEIB.00035 0072248C System EXEC Interface Block

-0008						5C E2EE2 C5C9240	*		*SYSEIB*
0000	0174422F	0101081F	D7D9D5E3	0000035C	E3F0F1C1	000000C9	00C87D0602000000	*	..... PRNT... *T01A... I. H' .....
0020	000000C4	D9D7D9E3	C4404000	00000000	000000C4	D9D7D9E3	C4404000	00000000	*... DRPRTD .....
0040	00000000	00000000	00000000	00000000	00000000	00		*	..... *



# CICS TS Transaction Abends (AKEB)

## 7. Locate last EXEC CICS call from Application Program:

### SYS\_TCA.00035 00722180 Task Control Area (SystemArea)

```

0000 00000000 00000000 00000000 00000000 0000035C 03636020 00000000 ..... *
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
0040 00805088 00000000 00000000 00000000 00000000 00000000 00000000 ..... *
0060 00000000 00000000 00000000 00000000 00000000 00000000 80000000 ..... *
0080 00000000 007224E4 00000000 00000000 00722388 00000000 00801128 008013E8 *..... U..... h..... Y*
    
```

TCA(S) + x'40' ==> User Savearea

## 8. User Savearea: (From TCA(S) + x'40')

### USER24.00035 00801520 USER storage below 16MB

```

0000 C2F0F0F0 F0F0F3F5 D7D9D5E3 F0F0F5C1 00000000 00000000 00000000 00000000 *B0000035PRNT005A..... 00801520
0020 00000000 00000000 00000000 00000000 00000000 00000000 E900C800 00000000 *..... Z. 00801540
0040 116C0000 119C0000 00000000 116C0000 0000010C 010C0000 1C000CC4 D9D7D9E3 *.%.....%. DRPRT 00801560
0060 C44040D7 D9D5E300 00000000 00000000 00000000 00000000 00000000 00000000 *D PRNT..... 00801580
.
.
.
3B60 40404040 40000000 00000000 0080105C 00000000 5078841A 00000000 FFFFFFFD *..... *.... & d... 00805080
3B80 00805390 00788354 007871C8 00801BEE 40788B64 00801528 00802528 00803528 *..... c... H... 008050A0
3BA0 00804528 00783240 00783240 007871C0 2012004B 00000000 00000000 80768AD0 *..... 008050C0
3BC0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..... 008050E0
    
```

R14 points at last application program call.

```

000917      005700      MOVE FILENAME TO NAMEFILE.
000918      *EXEC CICS READ DATASET(FILENAME) RIDFLD(DAM KEY)
000919      *      INTO(PRTD-RECORD) END- EXEC.
000920      005710      MOVE '.. 0..... 00769 ' TO DFHEIV0                                510
000921      MOVE LENGTH OF PRTD-RECORD TO DFHB0020                                IMP 430 476
    
```

000922 CALL 'DFHEI1' USING DFHEIV0 FILENAME PRTD-RECORD DFHB0020 DAM KEY

## CICS TS Transaction Abends (AKEB)

### To Summarize:

1. External symptom indicated a generic VSE cancel code in an unknown module.
2. Failing task was KE\_Num 003D, AP Transaction Number 00035.
3. Failing transaction was PRNT from terminal T01A.
4. KERRD showed a VSE Abend at location x'0017367E' caused by a DASD File Protect violation. Probably an illegal SVC 00. This address is in the 24-bit System GETVIS. Further analysis revealed that the failing instruction was indeed an SVC 00 at displacement x'15C' into IJGXDAF, Direct Access (DAM) Logic Mod.
5. Register 1 was pointing at a DTFDA, but it was not flagged as I/O Complete, nor were there any Channel or Device Errors posted. This indicated an error detected by the I/O Supervisor, consistent with DASD File Protect violation.
6. Internal Trace was not running, but the EIB revealed that the last transaction program EXEC CICS request was for a Read Record from a DAM file (DRPRTD)
7. This error was caused by IBM System code. It was a problem with the initial setup of the DTF by CICS TS File Control. The Seek Address is pointing at the wrong location in the DTF.

# CICS TS Storage Violation

## What is a Storage Violation:

1. Data written outside transaction storage
2. Storage Isolation:

## Storage Check Zones:

1. Replace SAAs:
2. Storage chain protected:

## CICS TS detection of storage violations:

### System Initialization (SIT) parameters:

**CHKSTSK = NONE | CURRENT | ALL**

**CHKSTRM = NONE | CURRENT:**

**RENTPGM = PROTECT:**

**STGPROT=YES:**

**STGRVCY = YES:**

# CICS TS Storage Violation

## Speaker Notes:

### What is a Storage Violation:

1. **Data written outside transaction storage:** A storage violation occurs when data is written outside of the storage assigned to the current transaction. Since task storage is usually acquired contiguously, these overlays usually only damage storage for this transaction. However, sometimes, particularly if uninitialized or bad pointers are involved, the overlay can occur anywhere in the CICS TS partition.
2. **Storage Isolation:** To reduce the impact of storage violations, CICS TS isolates user storage in different partition getvis locations. Thus, storage violations by a user transaction are more likely to impact itself or another user transaction than CICS system areas. In addition, the user can request subsystem storage protection, which isolates CICS storage in a different, inaccessible, storage protect key (**STGPROT=YES** in SIT).

### Storage Check Zones:

1. **Replace SAAs:** In CICS TS, user storage areas are no longer chained together via Storage Accounting Areas (SAAs). In their place, task data in UDSA (24-bit) and EUDSA (31-bit) subpools are framed with Check Zones” (or “Crumple Zones”) which are checked by CICS during FREEMAIN and end-of-task. **CHKSTRM={CURRENT|NONE}** and **CHKSTSK={ALL|CURRENT|NONE}**. TIOAs still use the old SAA storage format.
2. **Storage chain protected:** Task data is chained at a higher level, so overlaid data no longer destroys the storage chain. If authorized to do so, CICS will attempt to repair storage violations and continue with the task. Otherwise the task is abended but its storage is placed out-of-service (not FREEMAINed). **STGRVCY={NO|YES}**. In either case, CICS TS control system attempts to continue after a storage violation.

### CICS TS detection of storage violations:

CICS detects storage violations involving TIOAs by checking the SAA chains when it receives a command to FREEMAIN an individual element of TIOA storage, at least as far as the target element. It also checks the chains when it FREEMAINs the storage belonging to a TCTTE after the last output has taken place. CICS detects storage violations involving user-task storage by checking the storage check zones of an element of user-task storage when it receives a command to FREEMAIN that element of storage. It also checks the chains when it FREEMAINs all the storage belonging to a task when the task ends.

The storage violation is detected not at the time it occurs, but only when the SAA chain or the storage check zones are checked.

# CICS TS Storage Violation

## System Initialization (SIT) parameters:

**CHKSTK = NONE | CURRENT | ALL** Set scope of CICS TS checks for Task Storage Check Zones. It will not change how often CICS TS will perform storage checks (see previous page):

- NONE** – Task storage-violation checking is deactivated.
- CURRENT** – Only storage for the current task is checked.
- ALL** – Storage for all tasks is checked.

**CHKSTRM = NONE | CURRENT:** Set scope of CICS TS checks for Terminal I/O storage (TIOAs)

**RENTPGM = PROTECT:** Read-only DSAs, RDSA and ERDSA, will be allocated from read-only key-0 protected storage.

**STGPROT=YES:** Sets Storage Protection in the CICS region. CICS will acquire user transaction storage in key 9 (normally reserved for partition F8), CICS subsystem storage (CICS Key) in the normal partition key. (key 3 for F2). When CICS switches back and forth between applications (transactions) and subsystem, the SPKA instruction sets the proper storage protect key. This prevents user applications from overlaying CICS storage. If “NO” is specified, all storage runs in partition key.

**STGRVCY = YES:** Corrupted SAAs or check zones are repaired and the transaction continues.

**CICS TS storage protection:** Attempting to write into the wrong storage area with CICS TS storage protection enabled will cause an application program check (ASRA) with Program Interruption Code = x'04' (Protection Exception). CICS TS also checks all application-supplied addresses, and will cancel the application with an AEYD abend.

Storage Protection options implemented	Execution key of program	Writing to this type of storage will cause ASRA or AEYD abends
Read-only storage (RENTPGM=PROTECT)	CICS-key and user-key.	CICS key 0 read-only storage (RDSA and ERDSA)
Subsystem storage protection (STGPROT=YES)	User-key	All CICS-key storage (CDSA and ECDSA)
Base CICS (all storage is CICS key storage) (RENTPGM=NOPROTECT and STGPROT=NO)	CICS-key and user-key	VSE/ESA (operating system) storage only

## CICS TS Storage Violation

### To diagnose Storage Violation Problems:

**F2 0103 DFHSD102 DBDCCICS A storage violation (code x'0FOC') has been detected by module DFHSMAR .**

F2 0103 DFHME0116 (Module: DFHMEME) CICS symptom string for message DFHSD102 is PIDS/564805400 LVLS/411 MS/DFHSD102 RIDS/DFHSMAR PTFS/VSE411 RCS/0000FOC.

F2 0103 DFHDU0201 DBDCCICS ABOUT TO TAKE SDUMP. DUMPCODE: SMD102 ,  
DUMPID: 37/0002

F2 0103 OS24I AN SDUMP OR SDUMPX MACRO WAS ISSUED

F2 0103 OS29I DUMP STARTED

F2 0103 OS30I DUMP STARTED. MEMBER=**DF200001.DUMP** IN SUBLIB=SYSDUMP. F2

F2 0103 1I51I DUMP COMPLETE

F2 0103 DFHDU0202 DBDCCICS SDUMPX COMPLETE. SDUMPX RETURN CODE X'00'

## CICS TS Storage Violation

Random storage overlays are among the most difficult problems to resolve. The best than can be done with an non-recurring problem is to locate the storage which was overlaid, and attempt to identify the data which overlaid it.

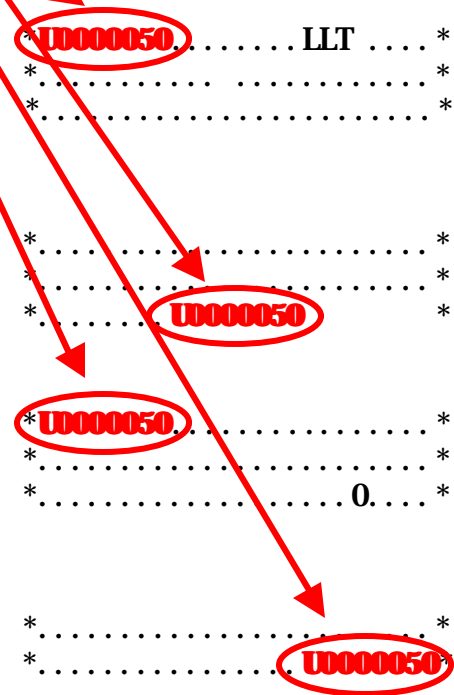
1. **Format System Dump:** Use `CALL DFHPD410 DATA DEF=0,KE=1,AP=1,SM=1,TR=1`. Locate the failing task in the Kernel Task Summary. It will no longer be visible in the AP Domain Transaction Summary.
2. Locate the exception trace entry (“\*EXC\*”) in the trace table.
3. Print detail (level 2) for the exception trace entry:  
Use `CALL DFHPD410 DATA DEF=0,TR=3,TRS=<ENTRY_NUM=(648)>`.  
Look up violation code (x'0F0C') in “CICS TS Trace Entries” to see what information is included in the trace entry.
4. Often, you cannot locate the source of these problems from a single occurrence.  
See additional hints on page 70

# CICS TS Storage Violation

## Prior to Overlay:

USER31.00050 02200040 USER storage above 16MB											
0000	E4FOFOFO	FOFOF5FO	02203218	00000000	D3D3E340	061C0001	00000000	00000000	* UD000050	LLT .... *	02200040
0020	00000000	00000000	00000040	00000000	00000000	00000000	00000000	00000000	*	..... *	02200060
0040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	02200080
.	.	.	.	.	.	.	.	.	.	.	.
3A60	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	02203AA0
3A80	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	02203AC0
3AA0	00000000	00000000	E4FOFOFO	FOFOF5FO					*	..... *	02203AE0
USER31.00050 02203AF0 USER storage above 16MB											
0000	E4FOFOFO	FOFOF5FO	00000000	00000000	00000000	00000000	00000000	00000000	* UD000050	..... *	02203AF0
0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	02203B10
0040	00000000	00000000	00000000	00000000	022054F0	00000000	00000000	00000000	*	..... 0. ... *	02203B30
.	.	.	.	.	.	.	.	.	.	.	.
8C40	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	0220C730
8C60	00000000	00000000	00000000	00000000	00000000	00000000	E4FOFOFO	FOFOF5FO	*	..... UD000050	0220C750

'U' (EUDSA) plus task id



## After Overlay: (I did this myself, so I knew where it was. Now, let's see if we can find it using CICS TS tools)

3A80	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	02203AC0
3AA0	01020304	05060708	11121314	15161718					*	..... *	02203AE0
USER31.00050 02203AF0 USER storage above 16MB											
0000	21222324	25262728	31323334	35363738	00000000	00000000	00000000	00000000	*	..... *	02203AF0
0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	..... *	02203B10



# CICS TS Storage Violation

## 1. Format System Dump:

**CALL DFHPD410 DATA DEF=0, KE=1, AP=1, SM=1, TR=1**

=== DUMP SUMMARY

DUMPID: 37/0002

DUMPCODE: SMD102

DATE/TIME: 01/03/21 11:37:46 (LOCAL)

MESSAGE: **DFHSM102 DEDCCICS A storage violation (code X'0FOC') has been detected by module DFHSMAR .**

SYMPTOMS: PIDS/564805400 LVLS/411 MS/DFHSM102 RIDS/DFHSMAR PTFS/VSE411 PRCS/0000FOC

# CICS TS Storage Violation

## ===AP: AP DOMAIN TRANSACTION SUMMARY

<u>Tran No</u>	<u>Tran Id</u>	<u>Orig Tran</u>	<u>TCA Addr</u>	<u>TWA Addr</u>	<u>EIB Addr</u>	<u>SEIB Addr</u>	<u>EIS Addr</u>	<u>EIUS Addr</u>	<u>Facility Type</u>
TCP	CSTP	CSTP	01C74680	0210A438	0210A0D0	01C74A8C	01C74988	0210A008	
00006	CSSY	CSSY	00596080	0053A000	0058E0D0	0059648C	00596388	0058E008	
00007	CSSY	CSSY	00596680	0053A000	0059E0				
00020	CSSY	CSSY	005BB680	0053A000	0059F0				
00021	CSSY	CSSY	<del>005BB080</del>	<del>0053A000</del>	<del>005B40</del>				
00022	CSNE	CSNE	01C75080	0053A000	021130				
00023	CXPB	CXPB	01C76080	0053A000	021810				
00025	IESO	IESO	00595080	006C0438	006C00				

We have a problem here. The failing transaction is no longer visible in the AP Domain summary. It is still in the Kernel Task list, but with no transaction number or id.

## ===KE: Kernel Domain KE\_TASK Summary

<u>KE NUM</u>	<u>KE TASK</u>	<u>STATUS</u>	<u>TCA ADDR</u>	<u>TRAN #</u>	<u>TRANSID</u>	<u>DS TASK</u>	<u>KE KTCB</u>	<u>ERROR</u>
001E	02152B00	Unused						
<b>001F</b>	<b>0215F080</b>	<b>***Running**</b>	<b>00000000</b>				<b>01BE8480</b>	<b>035D9020</b>
0020	0215F400	Unused						
0021	0215F780	Not Running	00595080	00025	IESO		01BE8280	035D9020
0022	0215FB00	Unused						
0023	0216C080	Not Running	01C76080	00023	CXPB		01BE8080	035D9020
0027	02115780	Not Running	005BB680	00020	CSSY		03501480	035D9020

# CICS TS Storage Violation

## 2. Locate “\*EXC\*” entry in trace table:

00050	1	AP	00E1	EIP	EXIT	REWRITE	OK	00F4, 00000000 .....	00000606 .....	=000593=	
00050	1	AP	1949	APLI	EVENT	RETURN-FROM LE/VSE	Rununit_Initialization	OK	CICSREAD	=000594=	
00050	1	AP	1948	APLI	EVENT	CALL-TO-LE/VSE	Rununit_End_Invocation		CICSREAD	=000595=	
00050	1	AP	1949	APLI	EVENT	RETURN-FROM LE/VSE	Rununit_End_Invocation	OK	CICSREAD	=000596=	
00050	1	AP	1948	APLI	EVENT	CALL-TO-LE/VSE	Rununit_Termination		CICSREAD	=000597=	
00050	1	AP	1949	APLI	EVENT	RETURN-FROM LE/VSE	Rununit_Termination	OK	CICSREAD	=000598=	
00050	1	AP	1948	APLI	EVENT	CALL-TO-LE/VSE	Thread_Termination			=000599=	
00050	1	AP	1949	APLI	EVENT	RETURN-FROM LE/VSE	Thread_Termination	OK		=000600=	
00050	1	PG	0700	PGHM	ENTRY	FREE_HANDLE_TABLES				000601	
00050	1	PG	0701	PGHM	EXIT	FREE_HANDLE_TABLES/OK					
00050	1	AP	1941	APLI	EXIT	START_PROGRAM/OK	....., NO,		CICSREAD		
.											
.											
.											
XM	1	PG	0801	PGXM	ENTRY	TERMINATE_TRANSACTION					
XM	1	PG	0802	PGXM	EXIT	TERMINATE_TRANSACTION/OK					
XM	1	SM	0F01	SMAR	ENTRY	RELEASE_TRANSACTION_STG				=000642=	
XM	1	SM	0F0D	SMAR	EVENT	Storage_released	USER24	storage at	006C1008	=000643=	
XM	1	XM	1001	XMIQ	ENTRY	SET_TRANSACTION	INCREMENT			=000644=	
XM	1	XM	1002	XMIQ	EXIT	SET_TRANSACTION/OK				=000645=	
XM	1	AP	1700	TFIQ	ENTRY	SET_TERMINAL_FACILITY	YES			=000646=	
XM	1	AP	1701	TFIQ	EXIT	SET_TERMINAL_FACILITY/EXCEPTION	NO	TERMINAL		=000647=	
<b>XM</b>	<b>1</b>	<b>SM</b>	<b>0F0C</b>	<b>SMAR</b>	<b>*EXC*</b>	<b>Storage check failed at address 02203AFD</b>	<b>RELEASE_TRANSACTION_STG</b>			<b>=000648=</b>	
XM	1	ME	0301	MEME	ENTRY	SEND_MESSAGE	66, SMD10Z,	036848FE ,	00000002, 036848E0 ,	00000008, SM	=000649=

Hopefully, the failing transaction is still in the trace table. Search on “\*EXC\*”  
Backing up from this “exception entry”, we find the transaction termination for task 00050, program CICSREAD.

# CICS TS Storage Violation

## 3. Print detail for the exception trace entry (number 000648):

**CALL DFHPD410 DATA DEF=0, TR=3, TRS=<ENTRY\_NUM(648)>**

SM OFOC SMAR *EXC*	- Storage_check_failed_at_address - 02203AF0 FUNCTION(RELEASE_TRANSACTION_STG)											
TASK_XM	KE_NUM001F	TCB_00342000	RET_836A84A4	TIME_11:37:46.9870023767	INTERVAL_00.0000019531	=000648=						
1-0000	00280000	000000D1	00000000	00000000	B0000000	00000000	02000100	00000000	00000000	00000000	*	*
	0020	00000000	00000000									
2-0000	02203AF0											
3-0000	00008C80											
4-0000	<b>21222324 25262728 31323334 35363738</b>										*	*
	0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	0040	00000000	00000000	00000000	00000000	022054F0	00000000	00000000	00000000	00000000		
	01C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	01E0	00000000	00000000	00000000	00000000	00000000	00000000	00000030	00000000	00000000	*	*
5-0000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	0040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	01C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	01E0	00000000	00000000	00000000	00000000	00000000	00000000	<b>EAF0F0F0 F0F0</b>	00000000	00000000		
6-0000	00000000	00000000	00000650	0220A790	00000000	00000000	00000000	00000000	00000000	00000000		
	0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	0040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	03C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	03E0	00000000	00000000	00000000	00000000	<b>01020304 05060708 11121314 15161718</b>						
7-0000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		
	0040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000		

- Parm 1: SMAR parameter list
- Parm 2: Address of corrupted User Task Storage Area
- Parm 3: Length of corrupted User Task Storage Area
- Parm 4: Corrupted User Task Storage Area (first 512 bytes)
- Parm 5: Corrupted User Task Storage Area (last 512 bytes)
- Parm 6: 1024 bytes preceding corrupted storage area
- Parm 7: 1024 bytes following corrupted storage area

# CICS TS Storage Violation

## 4. Look up violation code (x'0F0C') in "CICS TS Trace Entries"

### 1.1.3.13 Storage manager domain trace points

Table 96. Storage manager domain trace points

<u>Point ID</u>	<u>Module</u>	<u>Lvl</u>	<u>Type</u>	<u>Data included in level2 Trace entry</u>
SM 0F0C	DFHSMAR	Exc	Storage check failure	<ol style="list-style-type: none"><li>1 SMAR parameter list</li><li>2 Storage address where the storage violation was detected.</li><li>3 Length of User Transaction Storage</li><li>* 4 First x'200' bytes of User Transaction Storage</li><li>* 5 Last x'200' bytes of User Transaction Storage</li><li>* 6 x'400' bytes preceding violation.</li><li>* 7 x'400' bytes following User Transaction Storage</li></ol>

\* Entries invalid in manual.

# CICS TS Storage Violation

So let's summarize what we know about this error:

1. **Location overlaid:** (32 bytes, centered around x'02203AF0')
2. **Data which overlaid the storage:** (x'01020304 05060708 11121314 15161718 21222324 25262728 31323334 35363738')
3. **Transactions (programs) which were running at the time of failure.**

See **AP DOMAIN TRANSACTION SUMMARY**.

**Caution:** The overlay may have occurred any time during the life of this transaction. So, for a comprehensive "suspects list", we would have to go back through the trace and identify all programs which were active at any point during the life of this transaction).

Most of the time, storage violations are caused by the failing transaction, usually by exceeding the size of the requested user storage block. If you do not recognize the data, or cannot find an obvious error in the failing program, it may be coming from another program. In any case, we often have to wait for it to fail again. But this time, we can be prepared:

1. **Turn on Aux Trace:** (if it is not already on): **AUXTR=ON** in DFHSIT. Be sure to define two Auxilliary trace datasets, and specify **AUXTRSW=YES** to automatically switch when one is full.
2. **Enable Internal Tracing:** **INTTR=YES** in DFHSIT. Consider increasing the size of the internal trace table beyond the 80K default with VSE/ESA (**TRTABSZ=NNN**, in K)
3. **Start collecting incidents:** Make a list of the type of overlay, location, and the programs active at any time during the failing transaction lifetime. Eventually, you will see trends allowing you to narrow down the problem scope.
4. **Contact CICS Level2 for specialized trace assistance.**

# CICS TS Short on Storage

## To diagnose Short on Storage Problems:

```
F2 0091 DFHSD133 FJC2CICS CICS is under stress (short on storage above 16MB).  
F2 0091 DFHDU0201 FJC2CICS ABOUT TO TAKE SDUMP. DUMPCODE: SMD133 ,  
DUMPID: 1/0001  
F2 0091 OS24I AN SDUMP OR SDUMPX MACRO WAS ISSUED  
F2 0091 OS29I DUMP STARTED
```

1. **Format System Dump:** Use DFHPD410 DEF=0,KE=3,AP=3,SM=3,TR=1
2. **Locate failing task:** Search for “SUSPEND QUEUE”.
3. **Check Space Management Summary:**
4. **Locate Kernel Linkage Stack:**
5. **Check exception trace entry:**
6. **Request trace detail for the exception entry:**
7. **Locate the failing code:**

## CICS TS Short on Storage

CICS TS manages eight separate storage areas ( DSAs) as follows:

- CDSA** The CICS Key DSA:
- RDSA** The read-only DSA:
- SDSA** The shared DSA:
- UDSA** The user DSA:
- ECDSA** The extended CICS-key DSA:
- ERDSA** The extended read-only DSA:
- ESDSA** The extended shared DSA:
- EUDSA** The extended user DSA:



# CICS TS Short on Storage

## Speaker Notes:

3. **Check Space Management Summary:** for the failing DSA. How many tasks are waiting? How often have we come up short? Has the DSA been extended into multiple extents?
4. **Locate Kernel Linkage Stack:** for the failing transaction (task). Contains sequence of calls by CICS TS system routines which preceded the error. This will show us the CICS TS system functions performed just prior to detecting the error, and often helps us ascertain how we got into this fix in the first place.
5. **Check exception trace entry:** in the trace table and compare with results from step 4.
6. **Request trace detail for the exception entry:** and others, as required, by re-running DFHPD410, The detailed trace will often show an explanation for the parameters in a level 1 trace entry.
7. **Locate the failing code:** if you suspect that the GETMAIN request is invalid, (See page 21). Unfortunately, the program storage is not formatted for this kind of a problem, so we have to find the program load address from the trace table, and use INFOANA to print that section of the dump.

CICS TS manages eight separate storage areas ( DSAs) as follows:

- CDSA** The CICS Key DSA: allocated below the 16MB boundary, always from CICS-key storage
- RDSA** The read-only DSA: allocated below the 16MB boundary from either read-only storage or CICS-key storage depending on the **RENTPGM** system initialization parameter
- SDSA** The shared DSA: allocated below the 16MB boundary from either user-key or CICS-key storage depending on the **STGPROT** system initialization parameter.
- UDSA** The user DSA: allocated below the 16MB boundary from either user-key or CICS-key storage depending on the **STGPROT** system initialization parameter
- ECDSA** The extended CICS-key DSA: allocated above the 16MB boundary, always from CICS-key storage
- ERDSA** The extended read-only DSA: allocated above the 16MB boundary, either from read-only storage or CICS-key storage, depending on the **RENTPGM** system initialization parameter
- ESDSA** The extended shared DSA: allocated above the 16MB boundary from either user-key or CICS-key storage depending on the **STGPROT** system initialization parameter.
- EUDSA** The extended user DSA: allocated above the 16MB boundary, from either user-key or CICS-key storage depending on the **STGPROT** system initialization parameter

# CICS TS Short on Storage

## 1. Format System Dump

**CALL DFHPD410 DATA DEF=0, KE=3, AP=3, SM=3, TR=1**

DUMPID: 1/0001

DUMPCODE: SMD133

DATE/TIME: .....:00 15:17:09 (LOCAL)

MESSAGE: **DFHSM133 FJC2CICS CICS is under stress  
(short on storage above 16MB).**

This task is asking for 96 million bytes. At first blush, that would appear to be the problem. Let's see who the culprit is.

## 2. Locate failing task

**==SM Suspend queue summary**

<u>KE Task</u>	<u>Tran #</u>	<u>Susptok</u>	<u>Subpool</u>	<u>DSA</u>	<u>Request</u>
<b>01A4E080</b>	0000038	0286000B	U0000038	EUDSA	<b>96066416</b>

Task Address (Use this to locate task in Kernel Task Summary)

## CICS TS Short on Storage

### 3. Check SM Summary for the failing DSA:

#### ==SM EUDSA Summary

```
Size: 1024K
Cushion size: 0K
Current free space: 960K (93%)
* Lwm free space: 960K (93%)
* Hwm free space: 1024K (100%)
Largest free area: 960K
* Times nostg returned: 0
* Times request suspended: 1
Current suspended: 1
* Hwm suspended: 1
* Times cushion released: 0
Currently SOS: YES
* Times went SOS: 1
* Time at SOS: 00:00:00.000
* Storage violations: 0
Access: CICS
* Extents added: 1
* Extents released: 0
Number of extents: 1
```

Hmmm ... Only one task,  
only one occurrence.  
I think we are getting closer.

```
Extent list:      Start      End      Size      Free
                  01B00000  01BFFFFF 1024K     960K
```

\* NOTE: these values reset at 18:33:29 GMT (the last statistics collection interval)

# CICS TS Short on Storage

## 4. Locate Kernel Linkage Stack

Locate the failing task, by matching the Task Address from the previous page.

===**KE: Kernel Domain KE\_TASK Summary**

<u>KE NUM</u>	<u>KE TASK</u>	<u>STATUS</u>	<u>TCA ADDR</u>	<u>TRAN #</u>	<u>TRANSID</u>	<u>DS TASK</u>	<u>KE_KTCB</u>	<u>ERROR</u>
0001	035F5880	KTCB Step	00000000			00000000	035D90B0	
0002	035F5500	KTCB QR	00000000			035F5000	035DB020	
0003	035F5180	KTCB R0	00000000			035D9000	035DA068	
.								
.								
0040	01A41B00	Unused						
<b>0041</b>	<b>01A4E080</b>	<b>Not Running</b>	<b>00594080</b>	<b>00038</b>	<b>QXXX</b>	<b>03507380</b>	<b>035DB020</b>	
0042	01A4E400	Unused						
0043	01A4E780	Not Running	00592680	00024	IES0	03507180	035DB020	

Now, locate the Linkage Stack that matches this Kernel No.

## CICS TS Short on Storage

KE NUM	@STACK	LEN	TYPE	ADDRESS	LINK REG	OFFS	ERROR NAME
<b>0041</b>	01A4F020	0120	Bot	83609BA0	83609E70	02D0	DFHKETA
0041	01A4F140	01F0	Dom	8362E480	8362E56E	00EE	DFHDSKE
0041	01A4F330	0370	Dom	836A8590	836A8D18	0788	DFHXMFA
0041	01A4F6A0	03A0	Dom	81820CF8	81821600	0908	DFHPGPG
			Int	+00D4	81820D86	008E	INITIAL_LINK
<b>0041</b>	<b>01A4FA40</b>	<b>04E0</b>	<b>Dom</b>	<b>818D9250</b>	<b>818DAF42</b>	<b>1CF2</b>	<b>DFHAPLI1</b>
			<b>Int</b>	<b>+2128</b>	<b>818D972E</b>	<b>04DE</b>	<b>LE370_INTERFACE</b>
			<b>Int</b>	<b>+1C94</b>	<b>818DB6CC</b>	<b>247C</b>	<b>GETMAIN</b>
0041	01A4FF20	0240	Dom	836778E8	83679CBE	23D6	DFHSMGF
			Int	+236C	836779F0	0108	SUSPEND_REQUEST
0041	01A50160	01C0	Sub	83685CF8	83685F80	0288	DFHSMSQ
			Int	+07EA	83685DBE	00C6	SUSPEND_REQUEST
			Int	+011E	836865F4	08FC	CHECK_NOTIFY_REQUIRED
0041	01A50320	0470	Dom	836245C0	83625CFE	173E	DFHDSAT
			Int	+0FD4	836246A6	00E6	CHANGE_PRIORITY
			Int	+1042	836		

Linkage Stack Address.

The culprit appears to be DFHAPLI1 (CICS TS – LE/VSE interface), well perhaps between it and LE/VSE.

### KERNSTCK 01A4FA40 STACK ENTRY KERNEL LINKAGE STORAGE

0000	000004E0	01A4F6A0	0050AA08	818DAF42 <sup>E</sup>	835FD080 <sup>F</sup>	0000008A <sup>0</sup>	01A4FE38 <sup>1</sup>	00509000 <sup>2</sup>	*.....*	01A4FA40
0020	818D9250 <sup>3</sup>	01A4FA40 <sup>4</sup>	018DA24F <sup>5</sup>	018D924E <sup>6</sup>	018DC24D <sup>7</sup>	018DD24C <sup>8</sup>	01A4FE38 <sup>9</sup>	01A4FE38 <sup>A</sup>	*.....*	01A4FA40
0040	0179FCD8 <sup>B</sup>	00594080 <sup>C</sup>	01A4FA40	00000000	01A4E080	00509000	0050A208	82FFFC4	*.....*	01A4FA40
0060	01A4FF20	818D9250	01A4F6A0	017180D0	01716080	00000001	00000004	80000000	*.....*	01A4FA40
0080	01A4FDE1	018DC808	018DC			64C25E	018DC79D	018DC808	*.....*	01A4FA40
00A0	018DC7A5	018DC8D8	01A4F			A4FDE1	018DC808	018DC7A5	*.....*	01A4FA40

Registers (14-12) when DFHAPLI1 was called.

# CICS TS Short on Storage

## 5. Check entries in the trace table:

```

00038 1 LD 0002 LDLD EXIT ACQUIRE_PROGRAM/OK 82100020, 02100000, 2612, 190, REUSABLE, ESDSA, NEW_COPY      =000667=
00038 1 AP 1940 APLI ENTRY ESTABLISH_LANGUAGE QXXX01, 02100000, 82100020, 00002612, USER, BELOW COBOL,    =000668=
.
.
.
00038 1 AP 1940 APLI ENTRY START_PROGRAM QXXX01, CEDF, FULLAPI, EXEC, NO, 0179FCD8,                    =000676=
00038 1 SM 0301 SMGF ENTRY GETMAIN 3A8C, YES, 00, LE_TWA, TASK31 =000677=
00038 1 SM 0302 SMGF EXIT GETMAIN/OK 01B00008 01B00008 =000678=
00038 1 AP 1948 APLI EVENT CALL-TO-LE/VSE Thread Initialization QXXX01 =000679=
00038 1 AP 1949 APLI EVENT RETURN-FROM-LE/VSE Thread Initialization OK QXXX01 =000680=
00038 1 SM 0301 SMGF ENTRY GETMAIN 5F220A58, YES, LE_RUWA, TASK31 =000681=
00038 1 SM 1206 SMPQ *EXC* Insufficient storage to satisfy request ALLOCATE PAGEPOOL STORAGE, 0352E9F8, 5F220A70, YES =000682=
00038 1 SM 1001 SMSQ ENTRY SUSPEND_REQUEST 5F220A70, 0352E9F8, NO =000683=

```

## 6. Re-run DFHPD410, requesting trace detail:

**CALL DFHPD410 DATA DEF=0, TR=3, TRS=<ENTRY\_NUM-(681)**

Invalid GETVIS length

SM 0301 SMGF ENTRY - FUNCTION(GETMAIN) **GET\_LENGTH(5F220A58)** SUSPEND(YES) REMARK(LE\_RUWA)  
STORAGE\_CLASS(TASK31)

TASK- 00038 KE\_NUM- 0041 TCB- 003B3000 RET- 818DAF42 TIME- 13: 36: 04. 6478565012  
INTERVAL- 00. 0000036250=000681=  
1- 0000 00400000 0000000E 00000000 00000000 B6580000 00000000 015401B0 01A4FFC0  
0020 0354C064 19490008 5F220A58 00000004 006C2148 0100D3C5 6DD9E4E6 C1400708

# CICS TS Short on Storage

## 7. Print Program Storage:

```
// JOB INFOANAL PRINT Portion of Dump
// EXEC PROC=DTRINFOA
// EXEC INFOANA, SIZE=300K
DUMP NAME SYSDUMP. F2. PMR72960
SELECT DUMP VIEWING
PRINT 02100000 02002612
RETURN
SELECT END
/*
/ &
```

**PRINT 02100000 02002612**

```
02100000 C4C6C8E8 C3F4F1F0 58F00014 58F0F0B4 58F0F00C 58FF000C 07FF0000 00000000 * IFHC410. 0... 00.. 00..... *
02100020 47F0F028 00C3C5C5 00000000 00000014 47F0F001 4ACEAC00 021000CC 00000000 * . 00.. CEE ..... 00..... *
02100040 00000000 00000000 90ECD00C 4110F038 98EFF04C 07FF0000 02100020 02100114 * ..... 0... 0<..... *
02100060 02100FE0 02100080 02100020 021004F8 02101B38 021000E0 00000000 00000006 * ..... 8..... *
02100080 C4C5D4D6 F0F14040 F2F0F0F1 F0F3F1F3 F1F2F5F7 F4F9F0F1 F0F1F0F0 00000000 * IEM001 20010313125749010100... *
021000A0 00000000 E08A3C44 00000000 50800B09 00200000 00000000 00808000 0000025C * ..... &..... **
021000C0 0000006A 00000800 40404040 05000001 02101AB8 00000000 FFFFFFFBC 02100020 * ..... *
021000E0 02100058 00000008 00000006 02100058 02101AB8 021016D0 00000005 00000000 * ..... *

02100FC0 58C0D080 05EF58C0 D0E850F0 D0785820 D05C58F0 22244110 938E05EF 40404040 * ..... Y&0..... * 0..... *
02100FE0 00108001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
02101000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
02101020 00000000 00000000 F3E3C7E3 00000000 03000000 00010020 00000000 00000000 * ..... STGT ..... *
02101040 00000000 00000000 000004F5 00000000 00000000 00000000 00000000 00000000 * ..... 5..... *
```

## CICS TS Short on Storage

“IDBA” missing between end of code and TGT3 indicates program was compiled nonreentrant (without "RENT" option). This caused LE/VSE to request an invalid amount for 31-bit working storage. When the customer recompiled his program with “RENT”, the problem went away.

Sample from another program:

```
0241E280 B6B05820 D05C58F0 21CC4110 A29405EF 947FD1A4 5820D05C 58F02224 4110A284 *.....*.0....sm.m'Ju*.0....sd*
0241E2A0 05EF4040 40404040 0241E2B4 0241E42C 0241E2D8 C9C4C2C1 00030000 F400012C *.. ..S...U...SQIDBA...4...*
0241E2C0 0003F000 03F00000 00400000 01D00000 00003100 00000615 10002000 30000010 *..0..0... ..*
0241E2E0 80010000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0241E300 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0241E320 00000000 0000F3E3 C7E30000 00000300 00000001 00200000 00000000 00000000 *.....TGT.....*
```



# CICS TS Short on Storage

## Summary:

1. Initial console message DFHSM0133 indicated a short on storage condition in EUDSA. This was confirmed by the preface in the DFHPD410-formatted dump.
2. The Storage Management Suspend Queue Summary pointed to a single task waiting for storage. This task is requesting 96Meg of transaction storage.
3. The EUDSA summary showed a single instance of short on storage, increasing the possibility that the condition was due to a single aberrant request, and not gradual storage consumption.
4. The transaction currently waiting for 96meg of storage was Kernel task number 0041, transaction id QXXX.
5. The Kernel Linkage stack shows the storage request immediately followed a call to LE/VSE by DFHAPLI1 (CICS TS – LE/VSE interface module).
6. The sequence from step 5 was confirmed in the trace table. A detailed (FULL) GETMAIN trace entry showed that the storage requested was actually x'5F220A58' bytes.
7. Program storage for transaction QXXX was not formatted by DFHPD410 for this type of dump. Printing the storage using INFOANA showed that the program was compiled without the RENT option. Changing the program options to include RENT and re-compiling, corrected the error.

## CICS TS Short on Storage

### DISPLAY CICS TS STORAGE (Fast Path 364) (IEDC Command)

<b>IESADMCST</b>		<b>DISPLAY CICS TS STORAGE</b>				<b>Time: 09:40:19</b>	
<b>Applid: DBDCCICS</b>	<b>Sysid: CIC1</b>	<b>Jobname: CICSICCF</b>				<b>CICS TS Level: 110</b>	
Storage Protection . . . . .		INACTIVE		Reentrant Programs . . . . .		PROTECT	
				CICS Trace Table size..		80	
Extended DSA:		(All sizes in kbyte)				LIMIT	25600
		ECDSA	EUDSA	ESDSA	ERDSA	Totals	
Current DSA Size . . . . .		2048	1024	1024	6144	10240	
Current DSA used . . . . .		1876	64	8	5220	7168	
*Peak DSA used . . . . .		1884	64	8	5220		
Peak DSA Size . . . . .		2048	1024	1024	6144	10240	
Largest free area/Free Storage		0.95	1.00	1.00	0.94		
Times short-on-storage (SOS)..		0	0	0	0	0	
DSA:						LIMIT	5120
		CDSA	UDSA	SDSA	RDSA	Totals	
Current DSA Size . . . . .		512	256	512	512	1792	
Current DSA used . . . . .		344	8	456	344	1152	
*Peak DSA used . . . . .		352	28	456	344		
Peak DSA Size . . . . .		512	256	512	512	1792	
Largest free area/Free Storage.		0.88	1.00	0.86	0.86		
Times short-on-storage (SOS)...		0	0	0	0	0	
PF1=HELP		2=REFRESH		3=END		4=RETURN	

# CICS TS is not Responding

## It could have several reasons:

1. VSE/ESA System Problem
2. CICS TS System Abend
3. CICS TS in a partition wait state or CICS TS System Code in a tight loop
4. CICS TS transaction(s) in a wait state

# CICS TS is not Responding

## 1. VSE/ESA System Problem

a. Is Attention Routine responding?

```
REPLID  
AR 0015 1188I NO REPLIES OUTSTANDING
```

No ... investigate possible system problem.

b. Are higher priority partitions running?

```
PRTY  
AR 0015 PRTY Z, Y, P, C, BG, FB, FA, F9, F8, F7, F6, F5, F4, F2, F3, F1  
  
D T  
AR 0015 1C39I COMMAND PASSED TO VSE/POWER  
F1 0001 1R46I TIME IS 17:28:44, DATE IS 04/17/1997  
F1 0001 1R46I 013 PAGES FIXED, 014 CURRENT TASKS
```

No ... higher priority partition (i.e. POWER) or VSE System task may be in a loop.  
You are in the wrong class.  
You should be attending “Problem Solving in VSE/ESA, (part 1)”

Yes ... Go to 2a

# CICS TS is not Responding

## 2. CICS TS System Abend

- a. What was the last console message from CICS?  
Enter "F2" on console input line and press F7  
Cancellation Msg? ... See previous sections.

## 3. CICS TS in a Partition Wait State

- a. Does CICS respond to External Interrupt?

```
MSG F2,DATA='CEMT I TA'
```

```
Tas(0000025) Tra(IES0) Sus Tas Pri ( 020 )  
Sta(S ) Use(CICSUSER) Rec(X' B59A0E8A80867884' ) Hty(EKCWAIT )  
Tas(0000072) Tra(TST2) Fac(A001) Sus Ter Pri ( 001 )  
Sta(T0) Use(SYSA ) Rec(X' B59A29ABBC9EB5CB' ) Hty(ZCIOWAIT)  
Tas(0000081) Tra(TST2) Fac(A002) Sus Ter Pri ( 001 )  
Sta(T0) Use(SYSB ) Rec(X' B59A2AD6DC5C528A' ) Hty(FCXCWAIT)  
Tas(0000090) Tra(CEMT) Fac(CNSL) Run Ter Pri ( 255 )  
Sta(T0) Use(CICSUSER) Rec(X' B59A2B6535A2A387' )
```

```
.  
. .  
.
```

Yes ... goto 4a (Transaction Waits)

# CICS TS is not Responding

This page  
intentionally  
left blank

# CICS TS is not Responding

b. Is CICS TS being dispatched? Check CSATODP (CSA+x'50'):

```
LOCATE F2,'AICA
AR+0015 MATCH FOUND AT 0058ADB0
V0058ADB0 C1C9C3C1 1C000000 E2E3D6D9 C1C7C540 B6 *AICA... STORAGE * R02078DB0
V0058ADC0 00000248 0050E8A0 0050B780 81394060 B6 *... &Y..&.. a. - * R02078DC0
15 E
AR 0015 1I40I READY

SHOW F2,58ADB0.70
V0058ADB0 C1C9C3C1 1C000000 E2E3D6D9 C1C7C540 B6 *AICA... STORAGE * R02078DB0
V0058ADC0 00000248 0050E8A0 0050B780 81394060 B6 *... &Y..&.. a. - * R02078DC0
V0058ADD0 00000000 02213080 013940D0 00509000 B6 *... .. }.&.. * R02078DD0
V0058ADE0 81393E40 017A0700 0000000B 0194D000 B6 *a.. ..:..... m}. * R02078DE0
V0058ADF0 01966458 019664E2 01966587 8058ADC0 B6 *.o...o.S.o.g...{* R02078DF0
V0058AE00 01280080 01280080 0010020C 0127E680 B6 *.....W.* R02078E00
V0058AE10 1307070F 017AB2D0 00000100 018B8200 B6 *.....:}...... b.* R02078E10

SHOW F2,58ADB0.70
V0058ADB0 C1C9C3C1 1C000000 E2E3D6D9 C1C7C540 B6 *AICA... STORAGE * R02078DB0
V0058ADC0 00000248 0050E8A0 0050B780 81394060 B6 *... &Y..&.. a. - * R02078DC0
V0058ADD0 00000000 02213080 013940D0 00509000 B6 *... .. }.&.. * R02078DD0
V0058ADE0 81393E40 017A0700 0000000B 0194D000 B6 *a.. ..:..... m}. * R02078DE0
V0058ADF0 01966458 019664E2 01966587 8058ADC0 B6 *.o...o.S.o.g...{* R02078DF0
V0058AE00 01280080 01280080 0010020C 0127E680 B6 *.....W.* R02078E00
V0058AE10 1307112F 017AB2D0 00000100 018B8200 B6 *.....:}...... b.* R02078E10
```

Yes ... goto 4a (Transaction Wait)

# CICS TS is not Responding

## Speaker Notes:

You can locate the CSA by searching for the printable string “AICA STORAGE”. Note, there is no terminating quote in the command. The first quote is a special symbol to notify the LOCATE command that you are searching for printable data (as opposed to hex data). Don’t forget to enter “15 e” to terminate the locate command. Otherwise it will lock the attention routine and reject all further console commands.

The CSA starts at +x’10’ after this eyecatcher. Last dispatched time-of-day (HHMMSSST) is a packed field (ends with “F”), and is located at CSA+x’50’. This value changes each time CICS TS goes through task dispatching (Dispatcher Domain). If this does not change, CICS is either in a partition wait, a loop in CICS TS system code, or a transaction has “seized” the system.

The CSA+x’4C’ points at the user area of the current TCA. The first word points at the system area of the TCA, which will be a x’100’ higher address. This System TCA will contain printable data fields for the current transaction, containing the current transaction and transaction status, in the case of an abend.



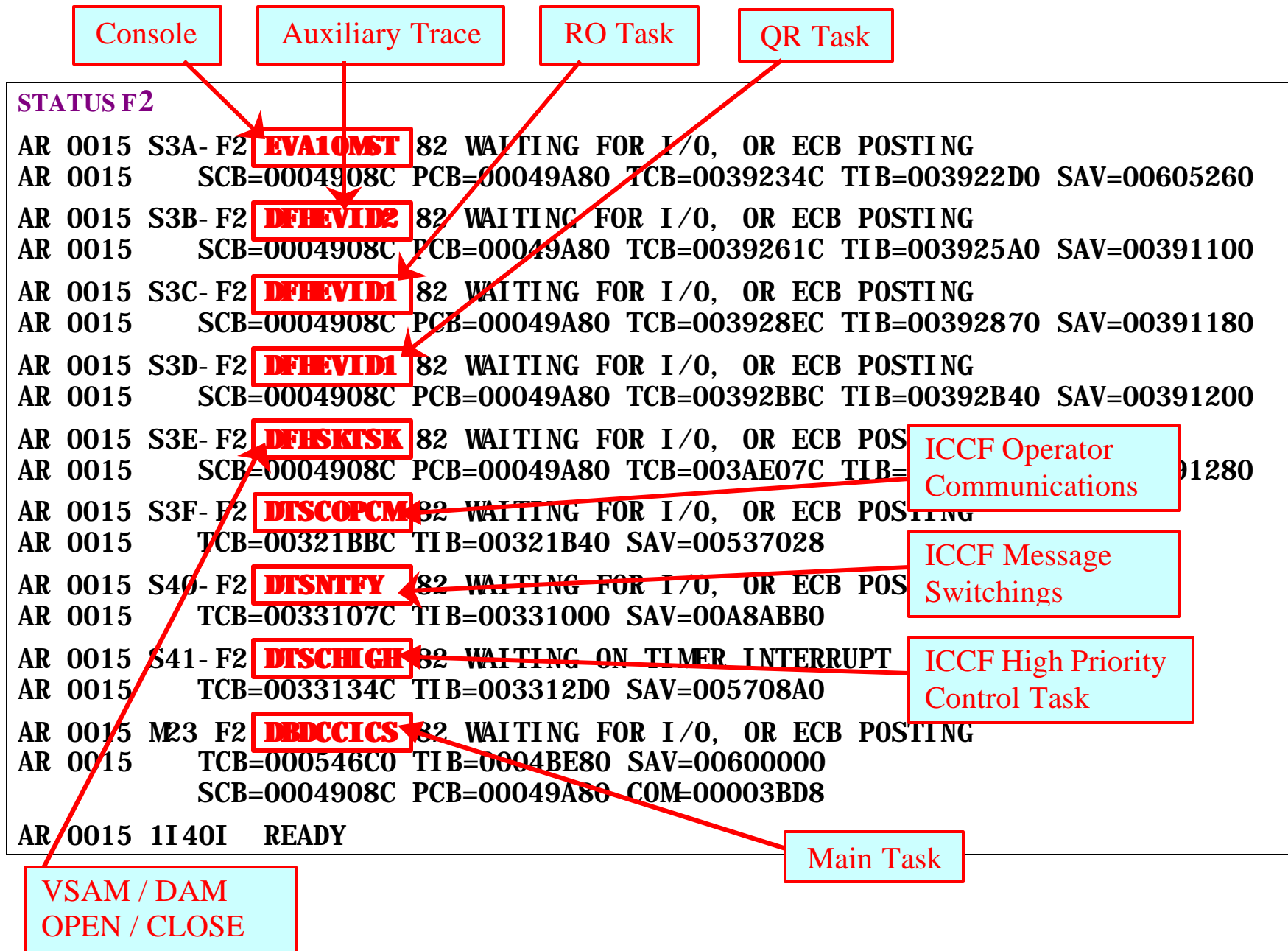
## Usage of VSE/ESA Sub-tasks

CICS TS uses multiple VSE/ESA subtasks, and the main task is dormant until CICS is terminated. Up to five VSE/ESA tasks are used to run CICS TS user transactions, and have the same name (**DFHEVID1**):

- **Quasi-Reentrant (QR) task:** this runs "normal" task code
- **Resource-Owning (RO) task:** CICS code will switch to this task when a long VSE/ESA wait will occur that would stop other CICS tasks from being dispatched via the QR subtask e.g. OPEN and CLOSE TD files, CICS-managed files such as dump datasets, load a program, journal archive submit, security SVC interface etc.
- **Secondary LU Usage (SZ) Task:** Front End Programming Interface (FEPI)
- **Sockets (SO) Task:** CICS Web Support
- **Sockets Listener (SL) Task:** CICS Web Support

In the VSE/ESA STATUS command display, the order from top is: RO, QR, then SZ, SO, and SL, if active

## Usage of VSE/ESA Sub-tasks



# Usage of VSE/ESA Sub-tasks

## VSE Sub-task 4 (QR) (DFHEVID1, TASKID 3D)

(Normally)

```
AR 0015 S3D- F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004908C PCB=00049A80 TCB=00392BBC TIB=00392B40 SAV=00391200
```

SHOW 391200.60

```
AR 0015 DATA FOUND AT 00391200
V00391200 C4C6C8C5 E5C9CAF1 07BD1000 81F498CC 06 *DFHEVID1... a4q. * R00256400
V00391210 01E2D2689 01E280D4A 00000001B 01E2D268C 06 *. SK.. S. M.... SK. * R00256410
V00391220 01F0BAC0D 00000006E 2B8BEC74F 000000010 06 *. 0. {..... * R00256420
V00391230 FE1D80001 01E2D268E 81F490E03 01E2D0004 06 *..... SK. a4. \. S}. * R00256430
V00391240 01F4A0DF5 000000006 806053B07 01EF60008 06 *..... * R00256440
```

Base Register

PSW

## Usage of VSE/ESA Sub-tasks

Base Register

SHOW 1F490E0.60

```

V01F490E0 00486ED4 D6C4C8C5 C1C403D7 00F4F1F0 B4 *.. >MODHEAD. P. 410* R001380E0
V01F490F0 C4C6C8C4 E2C4E2F3 F0F161F2 F061F9F9 B4 *DFHSDS301/20/99* R001380F0
V01F49100 7CF1F14B F4F10005 81F49B20 D1E5E2C7 B4 *@11. 41.. a4.. JVSG* R00138100
V01F49110 40404040 1E380218 00000002 00001E48 B4 * ..... * R00138110
V01F49120 00000360 00000000 183F4150 3FFF1840 B4 *... - ..... &... * R00138120
.
.
V01F498C0 41101000 13110700 070 0A84 07000700 B4 *..... d.... * R001388C0
V01F498D0 0A01 205 803098EF 80301824 1E269801 B4 *..... q..... q. * R001388D0
V01F498E0 22A81FE0 1FF147B0 38104100 00011FE0 B4 *.Y. \. 1..... * R001388E0
.
.
V01F4AB20 92687001 41000028 50007004 18175890 B4 *k..... &..... * R00139B20
V01F4AB30 D05498F0 90300DEF 94FDD219 58E0D238 B4 *} . q0..... m K. . \K. * R00139B30
V01F4AB40 07FED7C1 E3C3C840 C1D9C5C1 406040C4 B4 *.. PATCH AREA - D* R00139B40
V01F4AB50 C6C8C4E2 C4E2E340 F9F94BF0 F2F05A7F B4 *FHSDS3 99. 020! " * R00139B50
V01F4AB60 5A815A83 5A855A87 5A895A8B 5A8D5A8F B4 *!a!c!e!g!i! !.!. * R00139B60
    
```

PSW

**0A840700 07000A01:** SIMSVC instruction format.  
**0A84** (SVC 132)            Calls VSE/ESA OS/390 emulation.  
**07000700:**                Filler  
**0A01:**                    OS/390 SVC for Execute Channel program (EXCP)

## Usage of VSE/ESA Sub-tasks

Normally, the save area for the quasi-reentrant task will be pointing as shown on previous page, but ...

```
AR 0015 S3D-F2 DFHEVID1 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004908C PCB=00049A80 TCB=00392BBC TIB=00392B40 SAV=00391200
```

SHOW 00391200.60

```
V00391200 C4C6C8C5 E5C9C4F1 07BD0000 801D23E0 06 *DFHEVID1.....\* R00256200
V00391210 00667BE09 00628310A 006678505 001D2318C 06 *..#\..c...&... * R00256210
V00391220 011FAA18D 801D39A6E 001D457F7 801D408E0 06 *.....w...". * R00256220
V00391230 00667B801 000000002 01E13D843 00667B804 06 *..... * R00256230
V00391240 000004205 011FA1986 006284187 011FA1988 06 *.....q * R00256240
V00391250 0000B1FF 003D54EF 40404040 40404040 06 *..... * R00256250
AI CCB Ptr (R1) READY
V001D2310 00000000 00000000 47F0C018 C9D2D8C9 04 *.....0{.IKQI * R005EA310
V001D2320 D6C44040 F4F5C340 C4E8F4F4 F7F7F800 04 *OD 45C DY44778. * R005EA320
V001D2330 4560C8A0 5860D108 58106030 12114780 04 *.-H...J... * R005EA330
V001D2340 C0D49110 10024710 C0CA1B55 BF571009 04 *{Mj.....{..... * R005EA340
V001D2350 951B5000 4770C044 92075000 58501020 04 *n.&...{.k.&..&.. * R005EA350
V001D2360 4A55003A 4B50C900 947F5001 9238500B 04 *c...&I.m"&k.& * R005EA360
.
.
V001D23C0 0A4112FF 4780C0B4 1B000A06 5010B084 04 *.....{.....&..d * R005EA3C0
V001D23D0 181518E6 47F0C086 05EF18E6 47F0C0CA 04 *...W.0{f...W.0{. * R005EA3D0
V001D23E0 91801002 471000D2 0A07C02A 4560C804 04 *j.....{K..{..-H. * R004A23E0
V001D23F0 58C0D14C 07FE4BC0 C826180E 5880D108 04 *.{J<...{H....J. * R004A23F0
```

# Usage of VSE/ESA Sub-tasks

## Speaker Notes

This task is waiting in VSE/VSAM I/O routine IKQIOD. Register one points at a CCB (see following) which indicates that VSE/VSAM is currently reading from cylinder 19, head 1, record 3 on the device assigned to SYS199 (x'05C7' in CCB+x'0A', drop off the '05' and convert x'C7' to decimal).

The actual device can be identified by releasing a pause job and issuing the following command:

**0 LISTIO F2**

In this case, the customer was experiencing a corrupt file index, which was causing VSE/VSAM to loop reading the same index records over and over. See "Design and Tuning of VSE/VSAM" to identify which cluster was corrupted.

### Reg1 (CCB):

```
V00667B80 00001400 000005C7 00667840 01000000 B0 *.....G. .... *
V00667B90 00000000 00000000 00000000 00667BC0 B0 *.....#. *

```

First CCW in string

```
V00667840 63400010 00667BC0 47400010 00667BD0 B0 *. ....#. ....#. *
V00667850 85002000 011CCA00 03000038 00000000 B0 *..... *
V00667860 00000000 00000000 00000000 00000000 B0 *..... *

```

Define Extent Block

Extent definition (cchh – cchh):  
From Cyl x'13', hd 0, to Cyl x'1C' Hd 14

```
V00667BC0 98C02000 00000040 00130000 001C000E B0 *..... *

```

Locate Record Block

Record being read (cchhr):  
Cyl x'13', head 1, Record 3

```
V00667BD0 01800001 00130001 00130001 03482000 B0 *..... *
V00667BE0 00000000 13000104 00000100 00000000 B0 *..... *

```

# CICS TS Transaction Waits (Hangs)

## 4. CICS TS transaction(s) in a wait state

For Transaction status, use CEMT:

```
CEMT I TA
STATUS: RESULTS - OVERTYPE TO MODIFY
  Tas(0000025) Tra(IES0) Sus Tas Pri ( 020 )
    Sta(S ) Use(CICSUSER) Rec(X' B59A0E8A80867884' ) Hty(EKCWAIT )
  Tas(0000072) Tra(TST2) Fac(A001) Sus Ter Pri ( 001 )
    Sta(T0) Use(SYSA ) Rec(X' B59A29ABBC9EB5CB' ) Hty(ZCIOWAIT)
? Tas(0000081) Tra(TST2) Fac(A002) Sus Ter Pri ( 001 )
Sta(T0) Use(SYSB ) Rec(X' B59A2AD6DC5C528A' ) Hty(FCXCWAIT)
  Tas(0000089) Tra(CEMT) Fac(A003) Run Ter Pri ( 255 )
    Sta(T0) Use(OPER ) Rec(X' B59A2B0E2DCE7EC2' )

RESULT - OVERTYPE TO MODIFY
Task(0000081)
  Tranid(TST2)
  Facility(A002)
Runstatus(Suspended)
  Userid(SYSB)
  Recunitid(X' B59A2AD6DC5C528A' )
Htype(FCXCWAIT)
Hvalue(KSDS)
```

Task waiting on exclusive access to a VSAM record. File: KSDS

## CICS TS Transaction Waits (Hangs)

Or, from the VSE console:

```
MSG F2,DATA='CEMT I TA'
```

```
Tas(0000025) Tra(IES0) Sus Tas Pri ( 020 )  
Sta(S ) Use(CICSUSER) Rec(X' B59A0E8A80867884') Hty(EKCWAIT )  
Tas(0000072) Tra(TST2) Fac(A001) Sus Ter Pri ( 001 )  
Sta(T0) Use(SYSA ) Rec(X' B59A29ABBC9EB5CB') Hty(ZCIOWAIT)  
Tas(0000081) Tra(TST2) Fac(A002) Sus Ter Pri ( 001 )  
Sta(T0) Use(SYSB ) Rec(X' B59A2AD6DC5C528A') Hty(FCXCWAIT)  
Tas(0000090) Tra(CEMT) Fac(CNSL) Run Ter Pri ( 255 )  
Sta(T0) Use(CICSUSER) Rec(X' B59A2B6535A2A387')
```

*“If your task is waiting on resource type FCXCWAIT, it means that it cannot get exclusive control of a VSAM control interval at the present time. Another task already has shared or exclusive control of the control interval, so your task is suspended pending the release of that control interval. An exclusive control wait on resource type FCXCWAIT occurs within CICS, unlike the similar wait on FCIOWAIT, which occurs within VSAM.” “CICS TS Problem Determination Guide”*



# CICS TS Transaction Waits (Hangs)

## To diagnose Transaction Wait Problems:

1. Format System Dump: Use **DFHPD410 DEF=0,KE=3,AP=3,DS=1,FCP=3,TR=1. WAIT\_OLDC**".
2. Locate the task in the Kernel Domain KE\_Task Summary:
3. Locate TCA:
4. Locate User Savearea:
5. Locate the System Exec Interface Block (EIB):

If transaction is waiting on VSE/VSAM File access:

6. Identify transaction holding the resource we are requesting:
7. Locate FRTE for failing transaction:
8. Locate RPL Message Area:
9. If this transaction is waiting because there are no more strings available:
10. Locate program storage:

# CICS TS Transaction Waits (Hangs)

## Speaker Notes: (for following page):

1. **Format System Dump:** Use **DFHPD410 DEF=0,KE=3,AP=3,DS=1,FCP=3,TR=1**. Locate waiting task in the Dispatcher Domain Summary, either by the “Resource Type / Name” or via “C” in the “W” column. Most transactions waiting on ECBs are handled by CICS TS as “C=WAIT\_OLDC”.
2. **Locate the task in the Kernel Domain KE\_Task Summary:** Use the KE\_Task (address). Note the Transaction Number and TCA address. The Transaction Number will match the “CEMT I TA” display.
3. **Locate TCA:** Either by using the address or by searching for “TCA.NNNNN”, where “nnnnn” is the Transaction Number.
4. **Locate User Savearea:** Pointer at x’40’ into System TCA. Since this time, this is a 31-bit address, it will be located in a EUDSA storage area. Search for “USER31.NNNNN”, then locate the address in the right-hand column.

At this point we know why the transaction is waiting (which we already knew from the “CEMT I TA”), and we know the address in the user program (but not the statement number). Before we go after the user program storage, let’s check to see what command the failing transaction issued, and also extract the transaction history from the trace file.

5. **Locate the System Exec Interface Block (EIB):** Check EIBFN (displacement x’1B’). In this case, x’0602’ indicates “Read File”, which we kinda figured already. For File Access Request, displacement x’23’ contains the dataset name (EIBDS).

If transaction is waiting on VSE/VSAM File access:

6. **Identify transaction holding the resource we are requesting:** All long-running requests against a file are tracked via File Request Elements (FRTEs), which are chained from the File Control Table Element (FCTE).
7. **Locate FRTE for failing transaction:** FRTEs are not formatted in the same section as the FCTE. To locate the FRTE for the failing transaction, search on “FRTE”, and scan for transaction id. Otherwise, search for the FRTE matching the address in the FCTE +x’54’.. FRTE + x’30’ points at the VSWA, which contains the RPL starting at displacement x’08’. The VSWAs are formatted, but not associated with the transaction Search for “VSWA” and compare the address from the FRTE.
8. **Locate RPL Message Area:** When VSE/VSAM refuses a request because the resource is already owned (exclusive control conflict), it returns an area which points to the transaction owning this request. VSWA+x’3C’ points to this RPL Message Area. This is translated by CICS TS into a pointer to the VSWA for which I am waiting (VSWA +x’80’).
9. **If this transaction is waiting because there are no more strings available:** FCTE +x’7E’ (Number of tasks waiting on strings) will be non-zero, and the VSAM RPL Feedback will be zero. In addition, the VSWA +x’80’ will be zero, indicating that this request is not waiting on another specific request.
10. **Locate program storage:** The program which is waiting is not formatted by DFHPD410, and must be printed using INFOANA. See example on page 78. You can then locate the source statement using the procedure on page 21.

# CICS TS Transaction Waits (Hangs)

## 1. Locate waiting task in the Dispatcher Domain Summary:

**DEF=0, KE=3, AP=3, DS=1, FCP=3, TR=1**

**==DS: DISPATCHER DOMAIN - SUMMARY**

DS	TOKEN	KE	TASK	T	S	F	P	TT	RESOURCE	RESOURCE_NAME	W	TIME OF SUSPEND	TIMEOUT DUE	DIA (DTSK)	AD	ATTACHER	M	SUSPAREA
	00000003		035FAC80	S	S	N	N	-	ZC	DFHZNAC1	S	14: 53: 27. 050	-	03501080	XM	01C06700	Q	03501080
	00020003		035FA900	S	S	N	N	-	TLEXPIRY	DS_NUDGE	S	15: 46: 17. 114	-	03501180	TI	00470003	Q	03500710
	00940001		035FA580	S	S	N	N	IN	SMSYSTEM		S	15: 50: 37. 161	15: 55: 37. 161	01BDF80	SM	00000002	Q	03500680
	01800001		0215E080	N	S	P	N	-	OPEN_ANY	DFHPSPIO	W	12: 46: 16. 816	-	01BE8080	XM	01C06300	Q	021C90F9
	01840001		02151780	N	S	N	N	-	EKCWAIT		W	12: 46: 18. 168	-	01BE8280	XM	01C07700	Q	00614511
	01860035		02151400	N	R									01BE8380	XM	01C07100	Q	
	01880017		02151800	N	S	P	N	-	ZCLOWAIT	DFHZAR01	S	14: 47: 43. 119	-	01BE8480	XM	01C06B00	Q	01BE8480
	018C0017		02151080	N	S	P	N	-	FCXWAIT	KSDS	C	14: 52: 55. 130	-	01BE8680	XM	01C07300	Q	005A26B4

- T = TYPE OF TASK
- S = STATE OF TASK
- F = PURGEABILITY FLAG
- P = PURGE STATUS
- TT = TIMEOUT TYPE
- W = WAIT/SUSPEND TYPE**
- DIA= DISPATCHER TASK AREA
- AD = ATTACHING DOMAIN
- M = TASK MODE
- S=SYSTEM N=NON-SYSTEM
- D=DISPATCHABLE S=SUSPENDED R=RUNNING E=RESUMED EARLY
- P=PURGEABLE N=NOT PURGEABLE
- N=NO PURGE X=PURGED P=PURGE PENDING
- IN=INTERVAL DD=DEADLOCK DELAYED DI=DEADLOCK IMMEDIATE
- X=WAIT\_EXTERNAL S=SUSPEND C=WAIT\_OLDW**
- Q=QUASI-REENTRANT R=RESOURCE OWNING S=FEPI OWNING O=SOCKET OWNING L=LISTENER

# CICS TS Transaction Waits (Hangs)

## 2. Locate task in the Kernel Domain Task Summary:

==KE: Kernel Domain KE\_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KICB	ERRR
0001	035F3380	KTCB Step	00000000			00000000	035D70B0	
0002	035F3500	KTCB QR	00000000			03505000	035D9020	
.								
.								
.								
001F	02151080	Not Running	00597080	00081	TST2	01BE8680	035D9020	
0020	02151400	***Runni ng**	00597680	00100	CEM	01BE8380	035D9020	
0021	02151780	Not Runni ng	00595080	00025	IES0	01BE8280	035D9020	
0022	02151B00	Not Runni ng	00595680	00072	TST2	01BE8480	035D9020	

# CICS TS Transaction Waits (Hangs)

## 3. Locate TCA:

### TCA 00081 00597080 Task Control Area (User Area)

0000	00597180	00000001	021E9618	00589648	02153990	00000000	00000000	00000000	*.....o...o.....*
0020	00000000	0000081C	00000000	00000000	00000000	81F85A1E	021F1150	0000012C	*.....a8!...&...*
0040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
0060	00000081	01004000	00000040	021E9618	00000000	00000000	00000000	00000000	*...a...o.....*
0080	00000000	021E9618	00000000	00000000	00000000	00000000	00000000	00000000	*.....o.....*
00A0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
00C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	80589DC0	*.....*
00E0	00000000	00000000	00000000	0053A000	00000000	00000000	00000000	02213AB8	*.....*

### SYS\_TCA 00081 00597180 Task Control Area (SystemArea)

0000	00000000	00000000	00000000	00000000	0000081C	01BF050C	0000004A	00000000	*.....>.....*
0020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
0040	0221A588	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*..vh.....*
0060	00000000	80000000	00C00000	00000000	00000000	00000000	80000000	00000000	*.....*
0080	00000000	005974E4	00000000	00000000	00597388	022178C0	006C2128	006C23E8	*.....U.....h...%...% Y*
00A0	00000000	80589080	00000000	00000000	E3E2E3F2	021E9618	00000000	00000000	*.....TST2..o.....*
00C0	00000000	E3E2E3F2	00000000	00000080	00000000	00000000	00000000	00000000	*...TST2.....*
00E0	00000000	00000000	00000000	00000000	B59A2AD6	DC5C528A	00000000	00000000	*.....0.*.....*
0100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*
0120	00000000	00000000	00000000	00000000	80589400	00000000	00000000	0059756C	*.....m.....%*

# CICS TS Transaction Waits (Hangs)

## 4. Locate User Savearea:

**USER31.00081 02213AF0 USER storage above 16MB**

```

0000  EAF0F0F0 F0F0F0F1 00000000 00000000 00000000 00000000 00000000 *U0000061..... 02213AF0
0020  00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... 02213B10
0040  00000000 00000000 00000000 00000000 022154F0 00000000 00000000 *.....0..... 02213B30
0060  00000000 00000000 00000000 00000000 00000000 00000000 00000000 *..... 02213B50
.
.
.
6A80  00000000 00000000 00000000 00000000 00000000 00000000 00108001 02219418 *..... 0221A570
6AA0  00000000 8240268CE00000000F0221A6CCD 0221A74010221AA0020221AAA83000000014 *... b ... w... x ... 0221A590
6AC0  00000000500614BBC6000000007006C20D08 0221A7D890240214CA0240240CB022178C0C *... /... %... xQ. .<. 0221A5B0
6AE0  F3E3C7E3 02219828 03000000 61030220 0221A250 006151FC 0221A758 00000000 *3TGT..q.... /... s& /... 0221A5D0
    
```

## 5. Locate System Exec Interface Block (EIB):

**SYSEIB.00081 0059748C System EXEC Interface Block**

```

-0008  5CF2E8E2 C5C9C240 * *SYSEIB *
0000  0145252F 0101087F E3E2E3F2 0000081C C1F0F0F2 00000000 00007006 02000000 *... "TST2"... A002... '... *
0020  000000D2 E2C4E240 40404000 00000000 000000D2 E2C4E240 40404000 00000000 *... KSDS ... KSDS ... *
0040  00000000 00000000 00000000 00000000 00000000 00 *... *
    
```

EIBFN  
'0602' = Read

EIBDS  
'KSDS'

EIBTR

# CICS TS Transaction Waits (Hangs)

## 6. Locate File Control Table Entry:

**FCTE KSDS 0218FA90 FCT ENTRY**

Address	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8	Field 9	Field 10	Field 11	Field 12
0000	D2E2C4E2	40404040	0219A160	00000000	00000000	00EABA0A	80004400	10000000	*KSDS	.....	.....	*
0020	40400000	00020002	00000000	00000000	00000000	00000000	00000000	00000004	*	.....	.....	*
0040	00000006	00000002	00000000	B59A29AF	41F72AC8	0219D030	0219B670	0219B670	*	.....	7.....	*
0060	02184204	00000000	00000000	00000000	00000000	00008044	00000000	00010000	*	.....	.....	*
0080	000A0008	00000000	00000000	00000000	00C80000	000B000A	005B4170	40000000	*	.....	H.....\$..	*
00A0	00000000	00000000	C9D1E2E8	E2C3E300	40404040	40404040	00000000	00000000	*	.....	IJSYSCT.	*
00C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*	.....	.....	*
00E0	00000000	00000000	0000						*	.....	.....	*

Maximum number of strings

Maximum number of update strings

FRTE Chain Pointer

Number of active strings

Number of transactions waiting on strings.

**ACB KSDS 005B4170 VSAMACB** (FCTE +x'98' points to ACB)

0000	A040004C	00D7F718	001C45A8	0000000B	000ADA20	28110A00	00000000	D2E2C4E2	*	..<.P7....y.....	<b>KSDS</b> *
0020	40404040	00000000	00000000	01B29400	005A21AC	00000000	0A080000	00000000	*	.....m.!	*
0040	00000000	01000000	00000000						*		*

If transaction is waiting on FCTE strings, FCTE +x'7E' will be non-zero, and the VSAM RPL Feedback will be zero. In addition, the VSWA +x'80' will be zero, indicating that this request is not waiting on another specific request. (See next page.)

# CICS TS Transaction Waits (Hangs)

## 7. Locate FRTE and VSWA:

**FC FRAB AND FREE CHAIN FOR TRANSACTION NUMBER: 0000061**

**DEFRAB 0219C060 FC FRAB STORAGE**

0000 0219D030 005A2630 00597080 00000000 00000000 \*.....!.....\*

**DEFRTPS 0219D030 FC FREE STORAGE**

0000 00000000 00000000 0219D080 00000000 03000000 00000000 00000000 \*.....\*

0020 00000000 0219C060 0218FA80 0219B670 **005A2630** 00000000 00000000 \*.....\*

0040 00000000 00000000 00000000 \*.....\*

VSAM RPL Feedback  
x'080014' = Exclusive control Conflict

**VSWA 005A2630 VSAM WORK AREA**

0000 8F0000E4 00000000 0011003C 00000000 **0221AA00** 0221A7D8 000000C8 000000C8 \*...U.....xQ...H..H\*

0020 005B4170 021C000A 99C00000 **00080014** 00000000 00000000 00000000 **0221C778** \*\$. . . . . r. . . . . G.\*

0040 00800000 00000000 0218FA80 005A26EC 005A2700 00140010 00000000 00000000 \*.....!..!.....\*

0060 00000000 00000000 00000000 00000000 005A2540 00000000 00000000 00000000 \*.....!.....\*

0080 **005A2540** 00900000 0000001C 00000000 00000000 00597080 0219D030 00008000 \*!. . . . . \*.....\*

00A0 00000000 00000000 00000000 00000000 00000000 00000000 0219B670 \*.....\*

00C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 \*.....\*

00E0 00000000 \*.....\*

Pointer to VSWA /  
RPL we are waiting  
for

Pointer to RPL  
Message Area



# CICS TS Transaction Waits (Hangs)

## RPL Message Area:

**USER31.00081 0221C770 USER storage above 16MB**

```

0000 E4FOFOFO FOF0F8F1 D9D7B3C1 C4C4D940 7A404040 005A2548 40404040 40404040 *U0000081 HEADER : .!.. *
0020 40404040 40404040 E3C1E2D2 D5EAD440 7A404040 00072C40 40404040 40404040 * TASKNUM : ... *
0040 40404040 40404040 E3D9C1D5 E2C9C440 7A404040 E3E2E3F2 40404040 40404040 * TRANSID : TST2 *
0060 40404040 40404040 C6C3E3D5 C1D4C540 7A404040 D2E2C4E2 40404040 40404040 * FCTNAME : KSDS *
0080 40404040 40404040 E4FOFOFO FOF0F8F1 * U0000081 *
  
```

Pointer to VSWA /  
RPL we are waiting  
for

This transaction holds  
the resource we are  
waiting for

**CEMT I TA**

**STATUS: RESULTS - OVERTYPE TO MODIFY**

Tas(0000025) Tra(IES0) Sus Tas Pri ( 020 )  
Sta(S ) Use(CICSUSER) Rec(X' B59A0E8A80867884' ) Hty(EKCWAIT )

**Tas(0000072) Tra(TST2) Fac(A001) Sus Ter Pri ( 001 )**  
**Sta(T0) Use(SYSA ) Rec(X' B59A29A8BC9EB5CB' ) Hty(ZCICWAIT)**

? Tas(0000081) Tra(TST2) Fac(A002) Sus Ter Pri ( 001 )  
Sta(T0) Use(SYSB ) Rec(X' B59A2AD6DC5C528A' ) Hty(FCXCWAIT)

Tas(0000089) Tra(CEMT) Fac(A003) Run Ter Pri ( 255 )  
Sta(T0) Use(OPER ) Rec(X' B59A2B0E2DCE7EC2' )

# CICS TS Transaction Waits (Hangs)

## Summary:

1. User complained about a transaction not responding. CEMT Inquire Tasks showed transaction in FCXCWAIT status, indicating it was waiting for File Control resource.
2. We took a CEMT SNAP dump, and formatted it. According to the Dispatcher Domain Summary, the hung transaction was Kernel Task x'02151080'. Moving to the Kernel Domain Task Summary, the transaction id is found to be 0081, which matches the initial CEMT Inquire Tasks display.
3. Located the TCA for this task, then the User Savearea. Reg 14 pointed at the last EXEC CICS call.
4. The System EIB indicates the last EXEC CICS call was a Read Record to VSAM file KSDS.
5. The FCTE for KSDS did not show any transactions waiting on strings.
6. The FRTE for this transaction pointed at a VSWA, which showed that the File I/O request had been denied by VSAM because of an exclusive control conflict. The resource is currently owned by transaction 0072.
7. The trace table showed transaction 81 as waiting on a Get-for-Update on a VSAM record. Transaction 72 had previously read a record in update mode, then started a conversation with a terminal, which had not yet completed. This is a violation of VSAM file access protocol.

# CICS TS Trace (DFHTU410)

## CICS Tracing

1. "Normal" CICS tracing: is performed by the trace domain at predetermined trace points in CICS code during the regular flow of control:
  - **SYSTR={ON|OFF}** Switches the system master trace flag on or off at CICS startup.
  - **STNTR={1|(1,2)[,3)]|ALL|OFF}** Specifies the level of CICS standard tracing.
  - **STNTRXX={(1,2|1[,2][,3)]|ALL|OFF}** Specifies level of standard tracing for "xx" component.
2. "User" tracing is initiated by application calls:
  - **EXEC CICS ENTER TRACENUM**
  - **USERTR={ON|OFF}** Switches the user trace flag on or off at CICS startup.
3. "Special" tracing: Defined for specific CICS functions to assist with pre-defined types of problems:
  - **SPCTR={(1,2|1[,2][,3)]|ALL|OFF}** Specifies the level of special tracing.
  - **SPCTRXX={(1,2|1[,2][,3)]|ALL|OFF}** Specifies level of special tracing for "xx" component.

## CICS TS Trace (DFHTU410)

### 4. Internal tracing:

- **INTTR={ON|OFF}** Switches internal trace on or off at CICS startup.
- **TRTABSZ={16|NUMBER-OF-KILOBYTES}** Defines the size of the CICS internal trace table.

### 5. External Tracing:

- **AUXTR={OFF|ON}** Switches auxiliary trace on or off at CICS startup.
- **AUXTRSW={NO|ALL|NEXT}** Specifies automatic switching for auxiliary trace data sets when full.
- One or two dasd files (DFHAUXT and DFHBUXT), may be VSE/VSAM Managed SAM) or tape.

```
// DLBL DFHAUXT, ' CICS. AUXTRACE' , 0, VSAM, CAT=VSESPUC, X  
RECSIZE=4096, DISP=(NEW, KEEP) , RECORDS=(400, 0)
```

- When running DFHTU410 to print trace dataset, change **DISP** to (**OLD,KEEP**), to retain the aux trace dataset after print is complete, or (**OLD,DELETE**), to delete it.

### 6. Three levels of trace points:

- **Exception Trace:** Always included in internal trace whether trace is enabled or not.
- **Level-1 trace points:** Designed to give you enough diagnostic information to fix "user" errors.
- **Level-2 and higher:** Additional trace points for internal tracing. Only use on advice of IBM Support.

## CICS TS Trace (DFHTU410)

7. Use on-line transaction CETR to control tracing for this CICS session:

CETR	CICS Trace Control Facility		CIC1 DEDCCICS
Type in your choices.			
Item		Choice	Possible choices
Internal Trace Status	====>	STARTED	STArted, STOpped
Internal Trace Table Size	====>	80 K	16K - 1048576K
<b>Auxiliary Trace Status</b>	====>	<b>STOPPED</b>	<b>STArted, STOpped, Paused</b>
Auxiliary Trace Dataset	====>	A	A, B
Auxiliary Switch Status	====>	NO	NO, NExt, ALL
Master System Trace Flag	====>	ON	ON, OFF
Master User Trace Flag	====>	ON	ON, OFF
When finished, press ENTER.			
PF1=Hel p	3=Qui t	4=Comp onents	5=Ter/Trn 9=Error Li st

Change to "STA" to initiate external tracing

# CICS TS Trace (DFHTU410)

## Formatting and Printing the Trace

1. Internal trace can be formatted in one of two ways:

- From a CICS system dump, using the CICS print dump formatter, DFHPD410.  
Level of trace formatting is controlled via “TR={1|2|3}” parameter.
- From a transaction dump, using the CICS dump utility program, DFHDU410.  
Level of trace formatting is controlled via execution parm: “NOABBREV” or “NOFULL”.

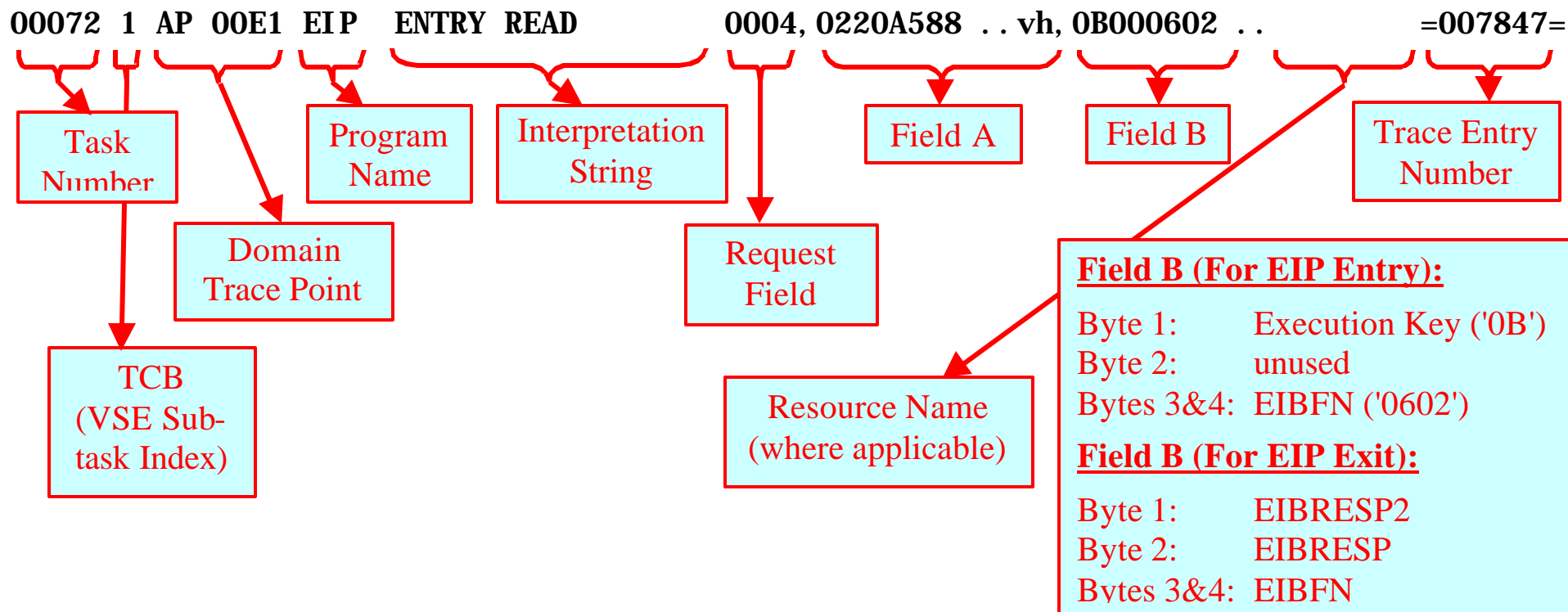
2. External trace can be formatted using DFHTU410:

```
// JOB PRINT TRACE DATASET
// LIBDEF *, SEARCH=(PRD1. BASE)
// DLBL DFHAUXT, ' CICS. AUXTRACE' , 0, VSAM, CAT=VSESPUC DISP=(OLD, KEEP)
// EXEC DFHTU410, SIZE=2M, PARM=' ABBREV, TASKID=(00081, 0072)'
/*
/ &
```

- Trace entries can be selected by trace entry number (**ENTRY\_NUM**), kernel domain task number (**KE\_NUM**), task id (**TASKID**), terminal id (**TERMID**), time period (**TIMERG**), transaction id (**TRANID**) or by domain / trace point (**TYPETR**).
  - Only exception trace entries may be printed, if requested (**EXCEPTION**).
  - Level of trace formatting (**ABBREV** or **FULL**)
3. “**ABBREV**”iated trace output prints a "one line per entry" trace table. Extended (or **FULL**) trace formatting prints a "many lines per entry" trace table.
4. There are two types of trace entries: Old Style and New Style:

# CICS TS Trace (DFHTU410)

## Old-Style Abbreviated Trace Entry:



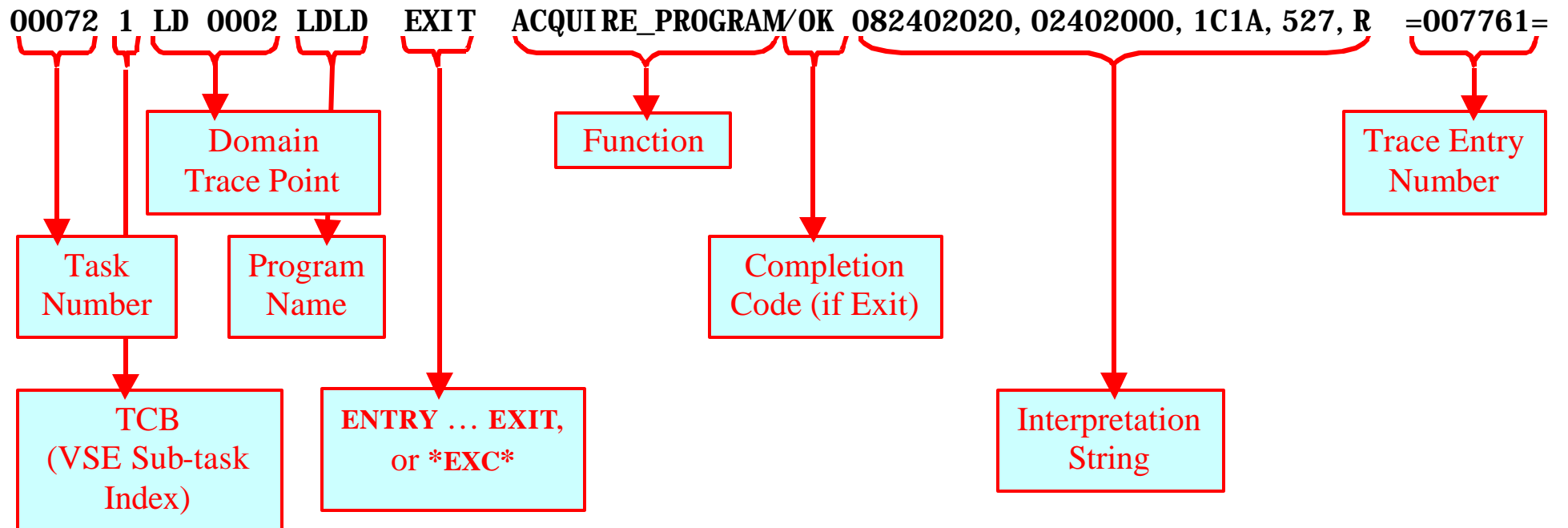
## Old-Style Full Trace Entry:

AP 00E1 EIP ENTRY READ REQ(0004) FIELD-A(0220A588 .. vh) FIELD-B(0B000602 .. . . .)

TASK-00072 KE\_NUM-0022 TCB-00342000 RET-8240268C TIME-14:47:42.7280805170  
 INTERVAL-00.0000023593 =007847=

# CICS TS Trace (DFHTU410)

## New-Style Abbreviated Trace Entry:





# CICS TS Trace (DFHTU410)

## New-Style Full Trace Entry:

The full trace entry defines the individual entries in the interpretation string. In addition, the full trace entry also includes other fields and control blocks pertaining to this trace point.

See “CICS TS Trace Entries” for a description of the additional data areas.

```

LD 0002 LDLD EXIT - FUNCTION(ACQUIRE_PROGRAM) RESPONSE(OK) ENTRY_POINT(82402020) LOAD_POINT(02402000)
                                                    PROGRAM_LENGTH(1C1A)

        FETCH_TIME(527) PROGRAM_ATTRIBUTE(REUSABLE) LOCATION(ESDSA) COPY_STATUS(NEW_COPY)

TASK-00072 KE_NUM-0022 TCB-00342000 RET-81D40D64 TIME- 14: 47: 39. 4471504072 INTERVAL-00. 0000037187      =007761=
  1-0000 00880000 0000001C 00000000 00000000  B4680A14 00000000 01680100 C3C9C3E2  *.h.....CICS*
  0020  D9C5C1C4 01CA5C70 02162858 01CA5C70  01020102 82402020 02402000 0A029420  *READ.*.....*
  0040  00001C1A 00000048 00000000 00000000  00000000 00000000 00000000 00000000  *.....*
  0060  10000100 00000527 8102BDD8 021104D0  010001D1 LDLD Parameter List *.....a.Q...}...J.*
  0080  00680200 81FEBE30                                *....a...*
```

### 1.1.3.7 Loader domain trace points

Table 90. Loader domain trace points

<b>Point ID</b>	<b>Mbdule</b>	<b>Lvl</b>	<b>Type</b>	<b>Data</b>
LD 0001	DFHLDLD	LD	Entry	1 LDLD parameter list
LD 0002	DFHLDLD	LD	Exit	1 LDLD parameter list

# CICS TS Trace (DFHTU410)

// EXEC DFHTU410, SIZE=2M PARM=' ABBREV, TASKID=(00081, 0072)'

```

.
.
.
00072 1 LD 0001 LDLD  ENTRY ACQUIRE_PROGRAM      01CA5C70          =007750=
00072 1 DS 0002 DSAT  ENTRY CHANGE_MODE          RO              =007751=
00072 2 DS 0003 DSAT  EXIT  CHANGE_MODE/OK       QR              =007752=

00072 2 DS 0002 DSAT  ENTRY CHANGE_MODE          QR              =007759=
00072 1 DS 0003 DSAT  EXIT  CHANGE_MODE/OK       RO              =007760=
00072 1 LD 0002 LDLD  EXIT  ACQUIRE_PROGRAM/OK  82402020, 02402000, 1C1A, 527, REUSABLE, ESDSA, NEW COPY =007761=
00072 1 AP 1940 APLI  ENTRY ESTABLISH_LANGUAGE    CICSREAD, 02402000, 82402020, 000          =007762=
00072 1 AP 1941 APLI  EXIT  ESTABLISH_LANGUAGE/OK COBOL_LE, 01CFFF6C, . . . .          =007769=
00072 1 AP 1940 APLI  ENTRY START_PROGRAM        CICSREAD, CEIE, FULLAPI, EXEC, NO,          =007770=
00072 1 AP 00E1 EIP   ENTRY READ                0004, 0220A588 ..vh, 0B000602 ...          =007847=

00072 1 AP 04E0 FCFR  ENTRY READ_INTO_UPDATE     0218FA80 , 0000000C, 0220A7DB,          =007850=
                                C8, 00000000, 0220A400, A, YES, GTEQ, NO, KEY, NO
00072 1 AP 04E1 FCFR  EXIT  READ_INTO_UPDATE/OK   C8, C8, 0, 00000000, , LENGTH_OK          =008119=
00072 1 AP 00E1 EIP   EXIT  READ                OK 00F4, 00000000 . . . . , 00000602          =008120=

00072 1 AP 00E1 EIP   ENTRY SEND-TC              0004, 0220A588 ..vh, 0B000404 ...          =008121=
00072 1 AP FD01 ZARQ  ENTRY APPL_REQ              021E9420, WAIT          =008122=
00072 1 AP FD18 ZSDS  ENTRY SEND_DFSYN             021E9420, A001, ZRVX    =008123=

00072 1 AP FD81 ZARQ  EXIT  APPL_REQ                =008155=
00072 1 AP 00E1 EIP   EXIT  SEND-TC                OK 00F4, 00000000 . . . . , 00000404          =008156=

```

## CICS TS Trace (DFHTU410)

<b>00072</b>	<b>1</b>	<b>AP</b>	<b>00E1</b>	<b>EIP</b>	<b>ENTRY RECEIVE-TC</b>	<b>0004, 0220A588 .. vh, 0B000402 ...</b>	<b>=008157=</b>
00072	1	AP	FD01	ZARQ	ENTRY APPL_REQ	021E9420, READ, WAIT	=008158=
00072	1	DS	0004	DSSR	ENTRY SUSPEND	01880017, DFHZARQ1, ZCIOWAIT, YES	=008161=
00081	1	LD	0001	LDLD	ENTRY ACQUIRE_PROGRAM	01CA5C70	=012472=
<b>00081</b>	<b>1</b>	<b>LD</b>	<b>0002</b>	<b>LDLD</b>	<b>EXIT ACQUIRE_PROGRAM/OK</b>	<b>82402020, 02402000, 1C1A, 0, REUSABLE, ESDSA, OLD_COPY</b>	<b>=012473=</b>
00081	1	AP	1940	APLI	ENTRY START_PROGRAM	CICSREAD, CEDF, FULLAPI, EXEC, NO,	=012474=
00081	1	AP	00E1	EIP	ENTRY READ	0004, 0221A588 .. vh, 0B000602 .	=012543=
<b>00081</b>	<b>1</b>	<b>AP</b>	<b>04E0</b>	<b>FCFR</b>	<b>ENTRY READ INTO UPDATE</b>	<b>0218FA80 , 0000000C, 0221A7DB, C8, 00000000, 0221AA00, A, YES, GTEQ, NO, KEY, NO</b>	<b>=012546=</b>
00081	1	SM	0301	SMGF	ENTRY GETMAIN	01BE9944 , 0000007E, E4, YES, VSWA	=012547=
00081	1	SM	0302	SMGF	EXIT GETMAIN/OK	005A2630	=012548=
00081	1	DS	0004	DSSR	ENTRY WAIT_OLDW	KSDS, FCIOWAIT, 00ABC841, NO, IO	=012549=
00081	1	DS	0005	DSSR	EXIT WAIT_OLDW/OK		=012550=
<b>00081</b>	<b>1</b>	<b>AP</b>	<b>04B7</b>	<b>FCVS</b>	<b>*EXC* VSAM</b>	<b>EXCEPTION VSAM RPL</b>	<b>=012551=</b>
00081	1	SM	0C01	SMG	ENTRY GETMAIN	80, YES, USER	=012552=
00081	1	SM	0C02	SMG	EXIT GETMAIN/OK	0221C778	=012553=
00081	1	AP	04B7	FCVS	*EXC* VSAM	EXCEPTION VSAM RPL	=012554=
<b>00081</b>	<b>1</b>	<b>AP</b>	<b>04BA</b>	<b>FCVS</b>	<b>*EXC* WAIT_FOR_EXCLUSIVE_CONTROL</b>		<b>=012555=</b>
00081	1	DS	0004	DSSR	ENTRY WAIT_OLDC	KSDS, FCXCWAIT, 005A26B4, YES, IO	=012556=