

# Problem Determination Comparison

## CICS/VSE and CICS TS

- CICS usage of VSE sub-tasking ..... Page 2
- General Principles of CICS dump processing ..... Page 7
- Storage Violations..... Page 10
- Analyzing Transaction Waits
  - ⇒ CICS/VSE File Control Waits..... Page 13
  - ⇒ CICS/VSE General Transaction Waits..... Page 21
  - ⇒ CICS TS Transaction Waits ..... Page 24
- System Dumps
  - ⇒ CICS/VSE (DFHDAP) ..... Page 30
  - ⇒ CICS TS (DFHPD410)..... Page 31
- Combining VSE SYSDUMP files ..... Page 34
- STXIT save areas and error tables
  - ⇒ Program Check / Abend Trace Table (CICS/VSE)..... Page 39
  - ⇒ Kernel Error Table (CICS TS)..... Page 46

*“CICS for VSE/ESA Problem Determination Guide”* (SC33-0716-02)

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

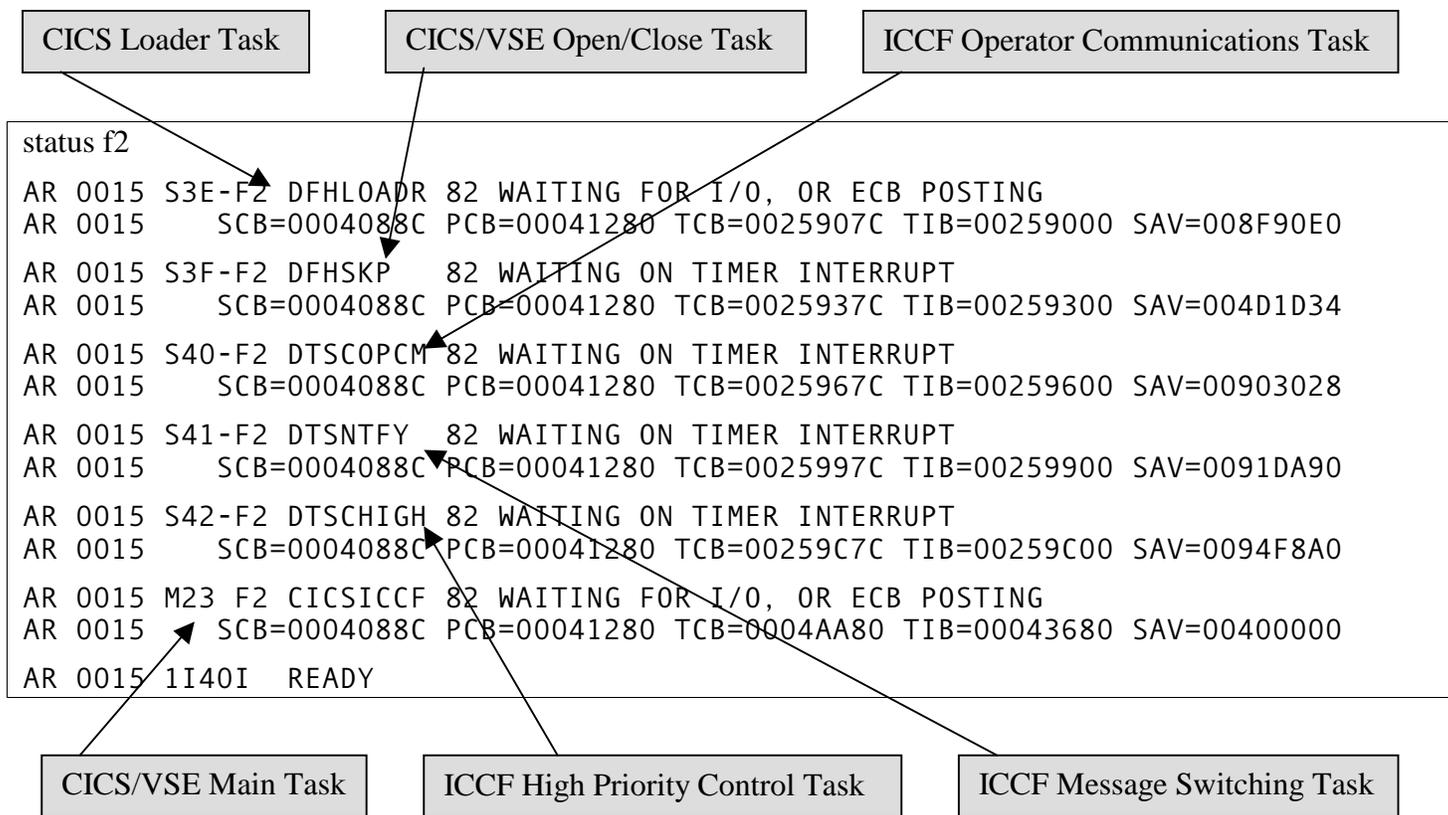
[RETURN TO INDEX](#)

VM/ESA and VSE/ESA  
Technical Conference  
Orlando, Florida  
May 31<sup>st</sup> – June 3<sup>rd</sup>, 2000

Charles E. Olsen

## VSE/ESA Sub-tasks for CICS/VSE

Under CICS/VSE, user transactions run in the CICS partition main-task. VSE sub-tasks are primarily used for File Control open/close and dump processing.



# VSE/ESA Sub-tasks for CICS/VSE

## VSE Main Task

(Normally)

```
AR 0015 M23 F2 CICSICCF 82 WAITING FOR I/O, OR ECB POSTING
AR 0015 SCB=0004088C PCB=00041280 TCB=0004AA80 TIB=00043680 SAV=00400000
```

show f2,400000.60

```
AR 0015 DATA FOUND AT 00400000
V00400000 C4C6C8E2 C9D74040 07BD0000 00406832 B6 *DFHSIP ..... ** R01313000
V00400010 000000009 00403C88A 004000A8B 00402F90C B6 *..... .h. .y. ** R01313010
V00400020 00401680D 00000000E 00408748F 008F11500 B6 *..... g....&* R01313020
V00400030 004C10001 004030902 004017243 00405CA04 B6 *.<... .. ** R01313030
V00400040 00406CA05 004CBA4C6 004C10407 004C13FC8 B6 *.%..<.<.<. <... ** R01313040
.
.
V00403F70 C4C6C840 000047A8 5CC4C6C8 D2C3D740 B6 *DFH ...y*DFHKCP * R01355F70
V00403F80 405C403F A4F0F2F3 F0C94040 4040E4D5 B6 * * .u0230I UN* R01355F80
V00403F90 F9F0F1F6 F8404000 00280248 004041A6 B6 *90168 ..... .w* R01355F90
V00403FA0 004041CE 90EBC020 05405850 44325820 B6 *.....{... .&.... ** R01355FA0
V00403FB0 D0804122 010058F0 C0009680 C01D50E0 B6 *}.....0{o.{.&\ ** R01355FB0
V00403FC0 C05805E0 91C0D084 47E0E01E 9180D085 B6 *{.. \j{}d. \.j. }e* R01355FC0
.
.
V00406800 47804B7A 05EF2600 00000004 000047FF B6 *..... ** R01361800
V00406810 4B7247F0 4B7A47F0 4B7E47F0 4B7E9180 B6 *...0.:.0.=.0.=j.* R01361810
V00406820 D10C710 4B92D203 700051C8 58102164 B6 *J....kK....H.... ** R01361820
V00406830 0A1D58E0 C00CBFFF E17C4780 4BCA58EE B6 *... \{....@..... ** R01361830
V00406840 01C841EE 02F0180E BFEFE00C 47804BCA B6 *.H...0.... \..... ** R01361840
V00406850 05EF2700 00000008 000047FF 4BBE47F0 B6 *.....0 ** R01361850
```

0A1D: (SVC 29) Wait multiple.  
Register 1 points at a list of ECBs, posting any of which will result in CICS/VSE being dispatched.

show f2,004C1000.60

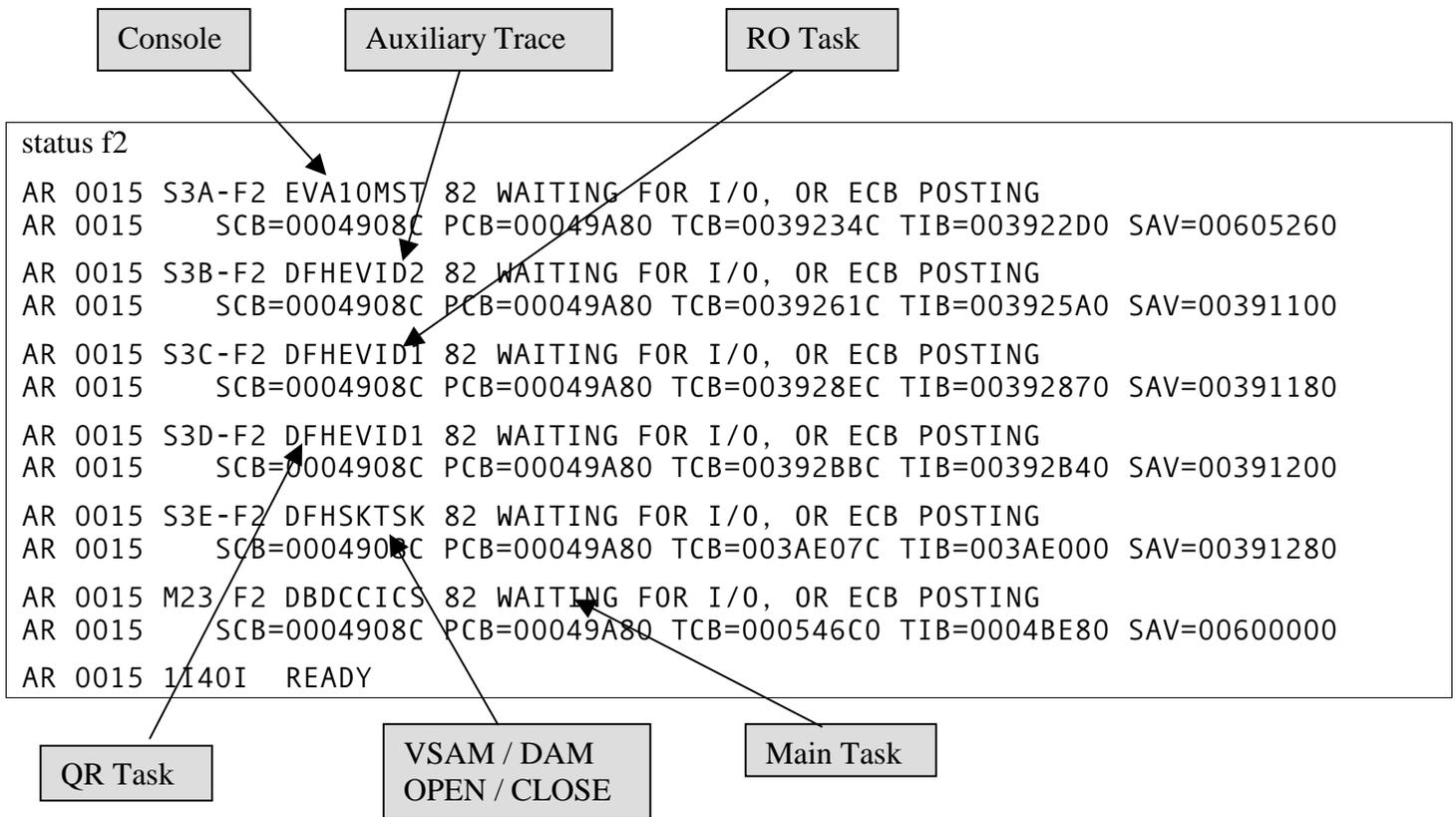
```
V004C1000 004016E8 004016E8 00400204 008F1328 B6 * .Y. .Y. .... ** R0146C000
V004C1010 008F1334 008F1340 008F134C 008F1358 B6 *..... <... ** R0146C010
V004C1020 008F1364 008F1370 008F137C 008F1388 B6 *.....@...h ** R0146C020
V004C1030 008F1394 004DF8BC 00880B48 004D9C8C B6 *...m.(8..h... (. ** R0146C030
V004C1040 008F2214 FFFFFFFF FFFFFFFF FFFFFFFF B6 *..... ** R0146C040
V004C1050 FFFFFFFF 00000000 00000000 00000000 B6 *..... ** R0146C050
```

## VSE/ESA Sub-tasks for CICS TS

CICS TS now uses multiple VSE/ESA subtasks, and the main task is dormant until CICS is terminated. Three 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)

In the VSE/ESA STATUS command display, the order from top is RO, QR then SZ if FEPI is active.



# VSE/ESA Sub-tasks for CICS TS

## 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 E5C9C4F1 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 01E2D2682 81F490E03 01E2D0004 06 *. ....SK.a4.\.S}* R00256430
V00391240 01F4A0DF5 000000006 806053B07 01EF60008 06 *.4.....-.....-* R00256440

V01F490E0 00486ED4 D6C4C8C5 C1C403D7 00F4F1F0 B4 *..>MODHEAD.P.410* R001380E0
V01F490F0 C4C6C8C4 E2C4E2F3 F0F161F2 F061F9F9 B4 *DFHSDSDS301/20/99* 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 07000A84 07000700 B4 *.....d....* R001388C0
V01F498D0 0A01B205 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 C4E2F340 F9F94BF0 F2F05A7F B4 *FHSDSDS3 99.020!"* R00139B50
V01F4AB60 5A815A83 5A855A87 5A895A8B 5A8D5A8F B4 *!a!c!e!g!i!i!i!i!i!* R00139B60
```

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

## VSE/ESA Sub-tasks for CICS TS

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

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

show 00391200.60

```
AR 0015 DATA FOUND AT 00391200
V00391200 C4C6C8C5 E5C9C4F1 07BD0000 801D23E0 06 *DFHEVID1.....\* R00256200
V00391210 00667BE09 00628310A 00667850B 001D2318C 06 *..#\..c....&....* R00256210
V00391220 011FAA18D 801D39A6E 001D457F 801D408E0 06 *.....w..."..* R00256220
V00391230 00667B801 000000002 01E13D843 00667B804 06 *..#.....d..#.* R00256230
V00391240 000004205 011FA1986 006284187 011FA1988 06 *.....~q..d...~q* R00256240
V00391250 0000B1FF 003D54EF 40404040 40404040 06 *.....* R00256250
AR 0015 1140I 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 *t....&I.m"&.k.&.* R005EA360
.
.
V001D23C0 0A4112FF 4780C0B4 1B000A06 5010B084 04 *.....{.....&..d* R005EA3C0
V001D23D0 81518E6 47F0C086 05EF18E6 47F0C0CA 04 *...W.O{f...W.O{* R005EA3D0
V001D23E0 91801002 4710C0D2 0A07C02A 4560C804 04 *j.....{K..{..-H.* R004A23E0
V001D23F0 58C0D14C 07FE4BC0 C826180E 5880D108 04 *.{J<...{H.....J.* R004A23F0
```

Reg1 (CCB):

```
V00667B80 00001400 000005C7 00667840 01000000 B0 *.....G... ..*
V00667B90 00000000 00000000 00000000 00667BC0 B0 *.....#.*
V00667840 63400010 00667BC0 47400010 00667BD0 B0 *.....#... ..#.*
V00667850 85002000 011CCA00 03000038 00000000 B0 *.....*
V00667860 00000000 00000000 00000000 00000000 B0 *.....*
V00667BC0 98C02000 00000040 00130000 001C000E B0 *.....*
V00667BD0 01800001 00130001 00130001 03482000 B0 *.....*
V00667BE0 00000000 13000104 00000100 00000000 B0 *.....*
```

# CICS Dump Processing

## CICS/VSE:

```
DFH2206  time applid TRANSACTION tranid HAS FAILED WITH ABEND abcode. RESOURCE
BACKOUT WAS SUCCESSFUL.
```

## CICS TS:

```
DFHAC2006 date time applid transaction tranid program progname abend abcode at
termid.
```

applid = CICS system  
tranid = transaction identifier  
progname = program name  
abcode = abend code  
termid = terminal identifier

## ALL abends should be investigated

**abcode**    **axxy**  
    **'a'**    is either **'A'** (CICS) or **'D'** (DL/I)  
    **xx**    is component / domain code  
    **y**     is alphanumeric code

ASRA (AKEA)    Termination due to program interrupt (progck)  
ASRB (AKEB)    Termination due to VSE system abend (invalid SVC, etc)  
ASRC (--)  
AICA (AKEC)    Termination due to run-away task

**'SR'** in CICS/VSE is "System Recovery".  
**'KE'** in CICS TS is "Kernel domain".

## CICS Dump Datasets:

CICS maintains one or two dump datasets: DFHDMPA, DFHDMPB. Defining CICS dump datasets is described in "*CICS/VSE System Definition and Operations Guide*" (SC33-0706) / "*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.

# CICS Dump Processing

## 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, 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 DFHDUP / DFH DU410 (See “*CICS System Definition and Operations Guide*”, SC33-0706 / SC33-1651). 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 DFHDAP / DFHPD410. PDUMPS are taken (directed to SYSLST) if an IDUMP request fails. (See “*CICS System Definition and Operations Guide*”, SC33-0706 / SC33-1651).

## 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]**

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 DFHDAP / 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*”, SC33-0713 / SC33-1658.
5. CICS/VSE:
  - If a storage violation occurs and "SVD=YES" or "SVC=nn".
  - If a transaction abends with ASRA or ASRB and FDUMP specified.
6. 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 Dump Processing

## Dump control for CICS TS:

1. System dumps (apart from CICS kernel domain dumps) can be suppressed or enabled:
  - DUMP (system initialization parameter)
  - CEMT SET SYSTEM DUMPING
  - EXEC CICS SET SYSTEM DUMPING
  - Dump domain XDUREQ global user exit program.
2. Transaction dumps can be suppressed or enabled for individual transactions:
  - EXEC CICS SET TRANSACTION DUMPING
  - DUMP attribute of the RDO TRANSACTION resource definition.
3. Transaction dumps can be suppressed or enabled for specific transaction abend codes:
  - CEMT SET TRDUMPCODE
  - EXEC CICS SET TRDUMPCODE

## Dump control for CICS/VSE:

Note, DFHPCT and RDO options over-ride SIT: See “*CICS Resource Definition (Macro)*” (SC33-0709)

### **DFHPCT TYPE=ENTRY**

**DUMP=(YES | NO)** => Should a transaction dump be taken with any abend code. Default is "YES".

**FDUMP=(ASRA | ASRB)** => In addition to a transaction dump, should a system dump be taken? Defaults to neither.

### **DFHPCT TYPE=INITIAL**

**FDUMP=(ASRA | ASRB)** => Sets default for TYPE=ENTRY statements

See “*CICS System Definition and Operations Guide*”, SC33-0706

If transactions are defined using RDO (Resource Definition Online), i.e. with CEDA, the equivalent to having **DUMP=YES** and **FDUMP=(ASRA,ASRB)** in DFHPCT is to code **DUMP=YES** in the **DEFINE APPLICATION**. (See “*CICS Resource Definition Online*”, SC33-0708; Note 2 in Appendix A “*Macro Operand to RDO Keyword*”.)

### **DFHSIT** (or CICS start-up SIT override parameters)

**ABDUMP=(YES | NO)** => Should a system dump be taken with ASRB abends.

**DUMP=(YES | NO)** => "**DUMP=NO**" suppresses all system dumps, except in case of CICS system abend. Default is "YES".

**PCDUMP=(YES | NO)** => Should a system dump be taken with ASRA abends.

**SVD=(0 | nn | YES | NO)** => Should a system dump be taken in cases of CICS storage violations. Defaults to "0".

See **CEMT SET DUMPOPTIONS** in “*CICS Supplied Transactions*”, SC33-0710

## Storage Violations

A storage violation occurs when data is written outside of the storage assigned to the current transaction. Often this overlay over-writes storage belonging to another transaction or CICS system tasks. To reduce the impact of storage violations, CICS TS isolates user storage in a different partition getvis location. Thus, storage violations by a user transaction are more likely to impact 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 parameters).

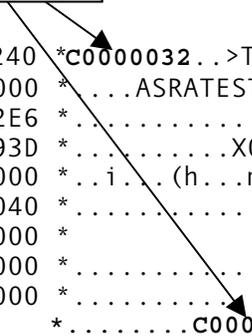
In CICS TS, user storage areas are no longer chained together via Storage Accounting Areas (SAAs). In their place, task data and TIOAs in UDSA (24-bit) and EUDSA (31-bit) subpools are framed with Check Zones” (or “Crumple Zones”) which are checked by CICS during FREEMAIN and end-of-task. (CHKSTRM={CURRENT|NONE} and CHKSTSK={ALL|CURRENT|NONE})

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). (STGRCVY={NO|YES}). In either case, CICS attempts to continue after a storage violation.

### Crumple Zone (CICS/TS):

012B9000	<b>C3F0F0F0</b>	<b>F0F0F3F2</b>	01206EE3	C1C3C240	00000000	00001846	C4C6C8E3	C1C3C240	* <b>C0000032</b> ...	>TACB	.....	DFHTACB	*
012B9020	00000000	C1E2D9C1	E3C5E2E3	C1C24040	00000000	00000000	00000000	00000000	* .....	ASRATESTAB	.....	.....	*
012B9040	00000000	00000000	00000000	0000B000	00000000	00000002	D9C5C7E2	50D7E2E6	* .....	.....	REGS.PSW	*	
012B9060	00000000	00000000	012AE7D8	81364940	012AEA60	0136593F	0136693E	0136793D	* .....	XQa	.....	.....	*
012B9080	0136893C	80684D88	012AED94	0128A834	00652000	00652000	806E7F80	00715000	* .. i ..	(h .. m .. y ..	.....	> .....	*
012B90A0	07BD0000	00715002	00020001	00000000	40404040	40404040	40404040	40404040	* .....	.....	.....	.....	*
012B90C0	40404040	40404040	40404040	40404040	00000000	00000000	00000000	00000000	* .....	.....	.....	.....	*
012B90E0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	* .....	.....	.....	.....	*
012B9100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	* .....	.....	.....	.....	*
012B9120	00000000	00000000	<b>C3F0F0F0</b>	<b>F0F0F3F2</b>	00000000	00000000	00000000	01000000	* .....	<b>C0000032</b>	.....	.....	*

'C' (EUDSA) plus task id



## Storage Violations

### SAA (CICS/VSE):

When a new transaction is started under CICS/VSE, storage is acquired for a Transaction Control Area (TCA). All further storage for this transaction is chained to the system area of the TCA. All transaction storage contains (in the first two full-words), a storage accounting area (SAA), which is repeated at the end of the area. CICS/VSE periodically checks the SAAs for a specific transaction to ensure they have not been overlaid, and cancels the transaction if corruption (“storage violation”) is detected.

Byte 0:	Type of storage:	42 = LIFO	81 = DCA	89 = RSA
		8A = TCA	8C = User	8F = File (VSWA or FWA)
		90 = RPL	93 = DWE	
Bytes 1-3:	Length of storage			
Bytes 4-7:	Storage chain forward pointer.			

### TCA(System):

```

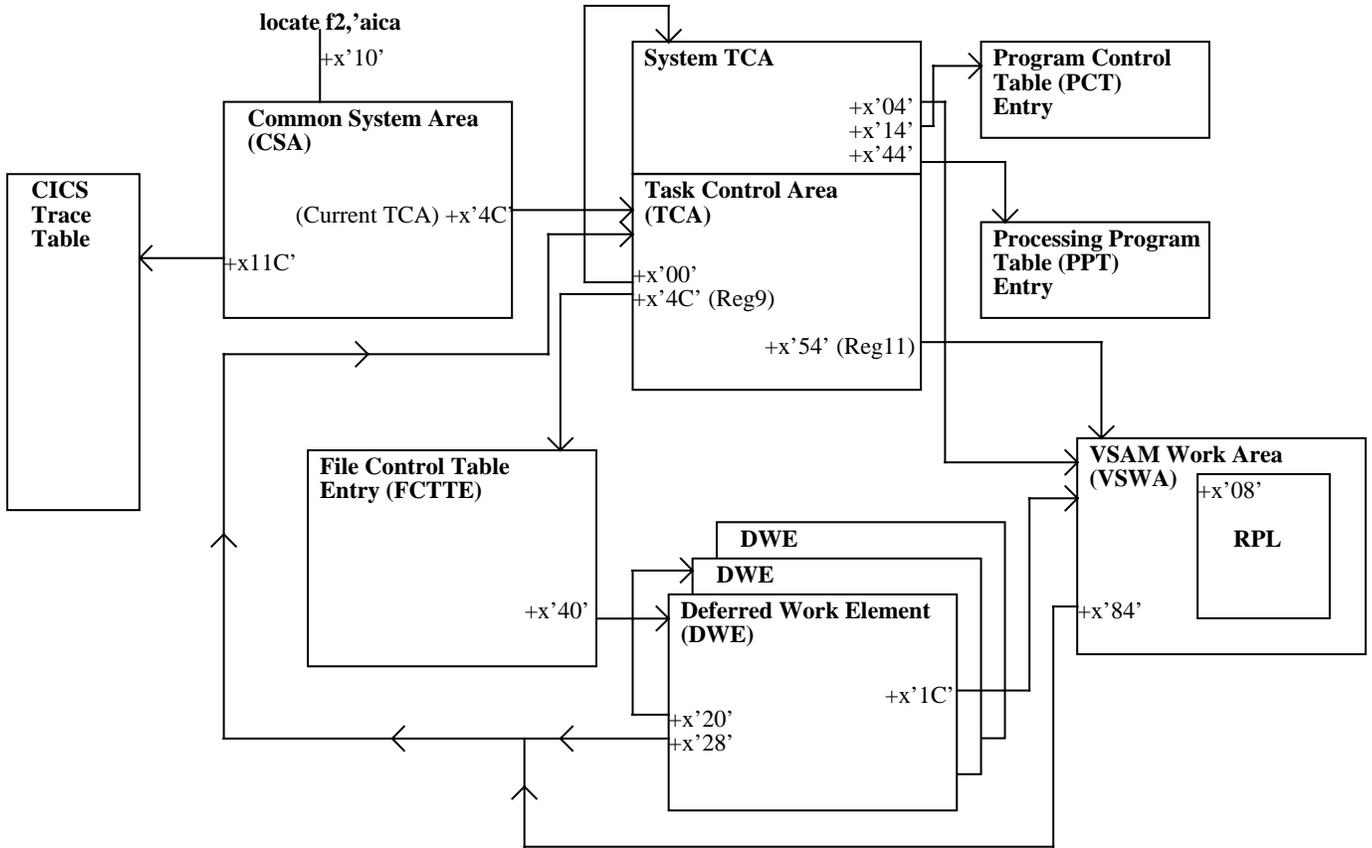
004E1800  8A041D88 004C2000 004C2F50 004E3670 4000090C 004C5684 004C01C0 00000000 *...H.....+.. ..D.....*
004E1820  00000000 00000000 00000000 00000000 00000000 004C5634 00000000 00000000 *.....*
004E1840  00000000 004E1AB0 00000000 00000000 00000000 00000000 00000000 00000000 *.....+.....*
004E1860  00000000 00000000 80000000 C1E2D9C1 00000000 00000000 00000000 00000000 *.....ASRA.....*
004E1880  FE4E1F60 00000000 FE4E1B30 FE4E3588 00403B68 004E1B40 00000000 00000000 *+.-.....+...+..H. ...+.....*
.
.
.
004E34F0  LINES TO 004E3570 SAME AS ABOVE
004E3570  00000000 00000000 00000000 00000000 00000000 00000000 8A041D88 004C2000 *.....H....*

004C2000  8C000F48 004E3590 D7C4E5C5 C3E3D6D9 00000000 00000000 00000000 00000000 *.....+..PDVECTOR.....*
004C2020  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
.
.
.
004C2DE0  F040F0F0 F4F0F1F6 F8F04040 F0F0F5C6 C5F8F0F8 40C6C6C6 C6C6C6C6 C502C000 *0 00401680 005FE808 FFFFFFFF.. *
004C2E00  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
004C2E20  LINES TO 0000F20 SAME AS ABOVE
004C2F40  00000000 00000000 8C000F48 004E3590 *.....+... *

```



# CICS/VSE Transaction wait on VSAM File



## CICS/VSE Transaction wait on VSAM File

### Locate Partition Save Area:

CSA (Reg 13)

```

show f2,500000.c0
V0500000  0  C4C6C8E2 C9D74040 07AD1000 8062E6B8 | DFHSIP  ....W. |
V0500010 10  0062DC929 0062EC92A 00D72A38B 006DE9B0C | ...k...k.P..._Z. |
V0500020 20  005097A0D 00509BA0E 00000000F 00D773E00 | ..p.....P.. |
V0500030 30  0062EE881 00D782F02 7062CC923 00D76D904 | ...h.Pb0...k.P_ |
V0500040 40  00D763F45 005000006 022FFFFFF7 00D76AD87 | .P.4.....P.Q |
V0500050 50  0000AC4F57625DFB6 421C26CE 945ABDCC | ...|.....m!.. |
V0500060 60  40404040 40404040 40404040 40404040 | | | | | | |
V0500070 70  40404040 40404040 5CC4C6C8 E2C9D740 | | | | | *DFHSIP |
V0500080 80  C4C6C840 000095AE 5CC4C6C8 E3C3E3C6 | DFH ..n.*DFHTCTF |
V0500090 90  C55C5000 A8F0F2F3 F0C90831 0109D7C9 | E*..y0230I....PI |
V05000A0 A0  C4D3C5E5 C5D30000 00D70E20 00D70E78 | DLEVEL...P...P.. |
V05000B0 B0  005000B4 005097A0 0050196C 00501976 | .....p..... |
    
```

Partition Begin can be found using the “MAP F2” command. In this case, the PSW points at CICS code (DFHFDP = Formatting Dump Program), so Reg13 points directly at the CSA. This saves us having to search for “AICA STORAGE”.

### Display CSA:

```

show f2,5097a0.150
V0509790 -10 C1C9C3C1 1C000000 E2E3D6D9 C1C7C540 | AICA....STORAGE |
V05097A0  0  00000000 00000000 00D8C7B0 006277A4 | .....QG.....u |
V05097B0 10  0062F2DC 0050196C 00627B6C 00626F3E | ..2.....?. |
V05097C0 20  00000002 00D72790 005000A8 00000024 | .....P.....y... |
V05097D0 30  806275B2 005E3050 00D70E80 00500B98 | .....;...P.....q |
V05097E0 40  00D72A38 00DB6A86 0030040C 006DE9B0 | .P.....f....._Z. |
V05097F0 50  1312306F 006C6E90 07D00100 00000000 | ...?..... |
V0509800 60  00488E7B 0005DC00 00008000 000083ED | .....c. |
V0509810 70  00005000 00500000 022FFFFFF 0096031F | .....o.. |
    
```

File Control Table (FCT)  
CSA+ x'12C

```

V05098A0 100 005DDBDC 00000000 0062F2DC 80D72FE0 | .).....2..P.. |
V05098B0 110 00000000 00516EDC 0151F9E8 00D77CA0 | .....9Y.P.. |
V05098C0 120 00000000 00000000 005000A8 005C5848 | .....y.*.. |
V05098D0 130 005E6BDC 0063BC28 00655000 0064FFD0 | ;;..... |
V05098E0 140 00000000 40D73010 0050196C 00000000 | .... P..... |
    
```

CSA+x'12C' has a pointer to the begin of the FCT. The FCT starts with an entry for DFHCSD. Starting with the begin of the FCT, search for the filename of the file you believe to be active.

## CICS/VSE Transaction wait on VSAM File

### Display File Control Table Entry (FCTTE):

		# of DWEs FCTTE+x'20'	DWE Chain FCTTE+x'40'	VSAM ACB FCTTE+x'90'		
show f2,5c7c28.e0						
V05C7C28	0	C9D7E740	40404040	00E0BA06	8000C47F	IPX .....D"
V05C7C38	10	00320540	40400000	00000002	00004000	.....
V05C7C48	20	0001002C	0001AE15	0001606D	00000000	.....-
V05C7C58	30	FFFBAC00	FFFF9775	00000000	00000000	.....p.....
V05C7C68	40	006C2510	0069C644	0069C644	00000000	.....F..F.....
V05C7C78	50	00000000	00000000	00000200	40000000	.....
V05C7C88	60	00008004	FF000000	00010000	00060004	.....
V05C7C98	70	00000000	00016675	00000000	002F0011	.....
V05C7CA8	80	00080008	00000000	00000000	00000000	.....
V05C7CB8	90	A040004C	00F04518	0019BBF8	00000008	..<.0.....8
V05C7CC8	A0	0008DA20	28110600	00000000	C9D7E740	.....IPX
V05C7CD8	B0	40404040	00000000	00000000	01FA6100	...../.....
V05C7CE8	C0	00D723B0	00000000	00080000	00000000	.P.....
V05C7CF8	D0	00000000	01000000	00000000	00000000	.....

All requests against a specific FCTTE are assigned a Deferred Work element (DWE).

FCTTE+x'20': Number of currently active DWEs for this file

FCTTE+x'40': Points at the first DWE in the chain.

### Display Deferred Work Element (DWE):

		Forward Chain Ptr DWE+x'20'	TCA(U) DWE+x'28'	VSWA DWE+x'1C'		
show f2,6c2510.30						
V06C2510	0	93000030	00000000	006DF4E0	005D7B68	l....._4..) ..
V06C2520	10	10801100	00000000	00000000	006DDC60	.....-
V06C2530	20	00000000	005C7C28	006DE9B0	00000000	.....*_..._Z.....

Storage class for a DWE is x'93'.

DWE+x'20': Forward chain pointer for DWEs that belong to this FCTTE

DWE+x'08': Chain for all DWEs (for all files). DWEs are never erased, just cleared when they are freed. So, you may find unused DWEs as you run the chain.

DWE+x'1C': Pointer to a VSAM Work Area (VSWA), if this DWE has a VSAM request associated with it.

DWE+x'28': Pointer to the TCA for this request is at

## CICS/VSE Transaction wait on VSAM File

### Locate VSAM Work Area (VSWA):

	VSAM ACB DWE+x'20'	RPL String id DWE+x'24'	VSAM Request Code DWE+x'25'	VSAM RPL DWE+x'08'
show f2,6ddc60.90				
V06DDC60 0	8F6D00A8	106DDC10	00000034	00000000
V06DDC70 10	06DDA48	006DDC20	00000032	00000032
V06DDC80 20	005C7CB8	00110000	981000FF	00000000
V06DDC90 30	00000000	00000000	00000000	005097A0
V06DDCA0 40	00000000	00000000	00000000	005DB708
V06DDCB0 50	0019BBF8	0000000C	006DDC68	006DF180
V06DDCC0 60	505DB656	0000009B	005C7CB8	006DEE88
V06DDCD0 70	005D9BF8	0069C644	005C7C28	006DDC10
V06DDCE0 80	006DDC60	006DE9B0	2C5C7C28	00000000

VSWAs are File I/O Areas, so have a storage id of x'8F'.

VSWA+x'08': Start of VSAM RPL.

VSWA+x'20': Pointer to ACB (FCTTE+x'90') (RPL+x'18'). This can serve as a cross-check to ensure you are still dealing with the same file.

VSWA+x'24': RPL String id. If this is zero, the VSWA is inactive.

VSWA+x'25': VSAM request code (RPL + x'1D')

x'00' → Point

x'04' → Get

x'08' → Erase

x'0C' → Put or Update

x'10' → Insert

x'1C' → End Request (Normally follows a transaction abend)

VSWA+x'2B': x'FF' if a request is currently active against this RPL. (RPL+x'23').

VSWA+x'2F': VSAM Feedback Code (RPL+x'27'). See "VSE/ESA Messages and Codes" under "VSE/VSAM Return and Error Codes".

VSWA+x'3C': Points to the CSA.

VSWA+x'84': Pointer to the TCA(U)

# CICS/VSE Transaction wait on VSAM File

Let's look at our TCA:

TCA System Area TCA(U)+x'00'		ECB pointer TCA(U)+x'14'		DCI TCA(U)+x'18'	
show f2,6DE9B0.80					
V006DE9B0	0	006DE800	00000000	00000000	00509BA0 * . _Y..... * *
V006DE9C0	10	006DEE88	00F33040	80005A00	C4318010 * . _h.3. .!.D... * *
V006DE9D0	20	505DD4C6 <sup>E</sup>	005DD3FA <sup>F</sup>	8019ED70 <sup>0</sup>	00DDA580 <sup>1</sup> * .)MF.)L.....v. * *
V006DE9E0	30	006DF180 <sup>2</sup>	701A0FDE <sup>3</sup>	006DDCA0 <sup>4</sup>	00F33040 <sup>5</sup> * . _1....._...3. * *
V006DE9F0	40	006DDC68 <sup>6</sup>	005D9BF8 <sup>7</sup>	705DD3FC <sup>8</sup>	005C7C28 <sup>9</sup> * . _..._.8.)L... * *
V006DEA00	50	006DDC10 <sup>A</sup>	006DDC60 <sup>B</sup>	006277A4	406D8800 * . _..._.-...u _h. * *
V006DEA10	60	0060D4A8	00653530	00000000	006D884C * . -My....._h< * *
V006DEA20	70	0060C4C0	6061337A	006DE800	006DF3E8 * . -D.-/..._Y..._3Y * *

As you can see, the address of the user area of our TCA [TCA(U)] ends with x'9B0'. This is because the system area in CICS 2.3 is x'1B0' long, and TCAs are acquired on 2K boundaries.

TCA(U)+x'00': Pointer to the TCA system area [TCA(S)].

TCA(U)+x'18': Contains the dispatch control indicator (DCI) (See "Problem Determination Guide")

TCA(U)+x'14': If DCI is x'80' or x'88', the transaction is waiting on an ECB at this address

TCA(U)+x'20': Register save area (Reg14 – Reg11).

TCA(U)+x'4C': (Reg9) Pointer to FCTTE, if the transaction is waiting on file services

TCA(U)+x'54': (Reg11) Pointer to the VSWA, if the transaction is waiting on file services

## System TCA:

Transaction id TCA(S)+x'11'		Storage Chain TCA(S)+x'04'		PCT Entry TCA(S)+x'14'		PPT Entry TCA(S)+x'44'	
show f2,6DE800.50							
V006DE800	8A040CD8	006EF800	006D5000	007197F0	*	..Q..8..._...p0	*
V006DE810	804444C0	0056DE08	00653D70	0064FFD0	*	...<.....	*
V006DE820	00000000	006C46C0	00000000	00000000	*	.....	*
V006DE830	00000000	FF594B64	00000000	00719488	*	.....mh	*
V006DE840	00716800	006DEAB0	00000000	00000000	*	.....	*

'8A' indicates TCA storage.

TCA(S)+x'04': Pointer to the storage chain for this transaction. Often this contains user records or a VSWA (VSAM Work Area), containing the VSAM RPL.

TCA(S)+x'11': Contains the transaction id (3 bytes packed decimal).

TCA(S)+x'14': Pointer to the PCT entry for this transaction.

TCA(S)+x'44': Pointer to current PPT entry (sometimes).

## Locate PCT for this task (System TCA + x'14'):

Transaction Name PCT+x'00'		Initial program call PCT+x'14'	
show f2,56DE08.30			
V0056DE00	.....	E6C9F9F9	00680000 * .....
V0056DE10	0001521E	72805A01	00000000 D7F6F5F2 * .....
V0056DE20	F7404040	C0000000	00000000 00504000 * .....

## CICS/VSE Transaction wait on VSAM File

**Now, the question everyone has been waiting for, what is this transaction waiting for?**

		VSAM CCB	CCB SYS043 CCB+x '06'	CCW Chain CCB+x '08'	
V0F33000	-40				08000000 00008000 00000000
00000000					
V0F33010	-30				.....
V0F33020	-20				.....
V0F33030	-10				.8.....8.....
V0F33040	0				.....3.....
V0F33050	10				.....8..
V0F33060	20				.8. ....3...8..
V0F33070	30				.....
V0F33080	40				. ...3.A. ...3..
V0F33090	50				. ...3.C....3..
V0F330A0	60				.....3.....
V0F330B0	70				.....3.....3..
V0F330C0	80				....i.....
V0F330D0	90				i.....i...
V0F330E0	A0				.....i.....
V0F330F0	B0				.....
V0F33100	C0				.....3... ..
V0F33110	D0				.....3.X. ....
V0F33120	E0				.....3... ..
V0F33130	F0				.....3.....3..

Since  $TCA(U)+x'18' = x'80'$ , this transaction is waiting on an ECB. The ECB pointer is found at  $TCA(U)+x'14'$ , and points to  $x'00F33040'$ . This is a VSAM CCB, and is initializing a new dasd extent on SYS043 (Note the string of  $x'1D'$  = Write Count-Key-data CCWs). The VSAM ARDB (volume list) could be used to identify the volume being written to, but this is sufficient for this exercise. The CCB is not posted ( $CCB+x'02' = x'80'$  indicates I/O complete), so we have some kind of system or hardware error preventing this I/O from completing.

### Waiting on Exclusive Control:

712592	0	80000000	00000000	00001DD4	C1E2E34B	.....MAST.
7125A2	10	D3C1C2D6	D94BD4C1	D5C1C7C5	4BC5D4D7	LABOR.MANAGE.EMP
7125B2	20	4BE2E4D4	D4C1D9E8	00000000	00009300	.SUMMARY.....L.
7125C2	30	00500071	08040002	80000001	C3060000	.....C...
7125D2	40	00010000	00000000	00000000	01000040	.....

When CICS receives a return code back from a VSAM Record Management request indicating Exclusive Control conflict (RPL Feedback of  $x'14'$  or  $x'D0'$ ), it places the DCA in wait on an ECB in the Data Set Name Block (DFHDSND), which contains the 44-character file-id. Multiple transactions may wait on the same ECB. The next time CICS detects a VSAM completion with a zero return code, it will re-dispatch all transactions waiting on this file. Under CICS TS, this logic has been replaced by the RPL message area support.

## CICS/VSE Transaction wait on VSAM File

### Waiting on SHR(4) access:

E0D5A0	0	.....	001E1108	E5E5D6D3	F6F1F100		.....VV0L611.	
E0D5B0	10	011C041A	E5D6D3F6	F1F10004	00008301		....VOL611....c.	
E0D5C0	20	01000000	88100044	00000000	50000000		....h.....	
E0D5D0	30	E5E5D6D3	F6F1F100	011C0005	AB1CA5C6		VV0L611.....vF	
E0D5E0	40	44DFD200	0000A800	.....	.....		..K...y.....	

With APAR DY43302 (ESA 1.3.6+), VSAM no longer makes the entire CICS partition wait if a transaction is waiting on SHR(4) access. Rather the transaction is set into a wait on an ECB located x'24' into the DTL.

### FCTTE:

V05C7C28	0	C9D7E740	40404040	00E0BA06	8000C47F		IPX	
V05C7C38	10	00320540	40400000	00000002	00004000		... ..	
V05C7C48	20	00010020	0001AE15	0001606D	00000000		.....-.....	
V05C7C58	30	FFFBA000	FFFF9775	00000000	00000000		.....p.....	
V05C7C68	40	006C2510	0069C644	0069C644	00000000		.....F...F.....	
V05C7C78	50	00000000	00000000	00000200	40000000		.....	
V05C7C88	60	00008004	FF000000	00010000	00060004		.....	
V05C7C98	70	00000000	00016675	00000000	00200011		.....	
V05C7CA8	80	00080008	00000000	00000000	00000000		.....	
V05C7CB8	90	0040004C	00F04518	0019BBF8	00000008		.<.0.....8....	
V05C7CC8	A0	0008DA20	28110600	00000000	C9D7E740		.....IPX	
V05C7CD8	B0	40404040	00000000	00000000	01FA6100		...../.	
V05C7CE8	C0	00D723B0	00000000	00080000	00000000		.P.....	
V05C7CF8	D0	00000000	01000000	00000000	00000000		.....	

- FCT +x'14-15': ECB to wait for string availability. (x'40' indicates ECB posted = string available)
- +x'5C': (x'40') at least one transaction against this file is currently in EXCPAD (waiting for VSAM ECB).
- +x'63': ECB for a transaction to wait for VSAM split activity to complete. (x'40' indicates ECB posted)
- +x'68': Current number of transactions (strings) active against this file
- +x'6A': Number of transactions waiting because no more strings available.
- +x'6C': Maximum number of simultaneous transactions against this file (string count)
- +x'6E': Maximum number of simultaneous updates / adds allowed against this file.

If a transaction must wait on a string (either read or update/add access), the ECB is located at FCT+x'14', with a "pseudo" string wait ECB at FCT+x'15'.

If a transaction must wait because another string is performing a VSAM CI or CA split, the ECB is located at FCT+x'63'.

Note, an ECB is posted (available) if the x'40' bit is turned on.

## CICS/VSE Transaction wait on VSAM File

### Waiting on LSR pool strings:

V05629A8	0	D3E2D9D7	D6D6D3F5	40404040	01050014	LSRP00L5	....	
V05629B8	10	00000000	00603530	00320040	00100007	.....-	.....	
V05629C8	20	00000000	0000000C	0000000C	00560010	.....	.....	
V05629D8	30	00000000	00000000	00280200	00000000	.....	.....	

When LSR pools are defined, a specific number of strings (simultaneous accesses) are reserved. If the number of transactions against files serviced by this LSR pool exceeds the number of LSR strings, transactions must wait until LSR strings are available. The ECB for this wait is in the CICS LSRPOOL control block.

### Waiting on VSE Sub-task activity:

V069A6A8	0	5CC4C6C8	E2D2C340	405C69A6	C8F0F2F3	*DFHSKC	*.WH023	
V069A6B8	10	F0C94040	4040D7C9	C4D3C5E5	C5D31000	OI	PIDLEVEL..	
V069A6C8	20	50E0C058	58E0C010	90FDE010	D203E00C	.....	.....K...	
V069A6D8	30	C0580530	4120301C	48020004	5860C000	.....	.....-	
V069A6E8	40	58F60090	05EF0530	182147F0	301C0700	.6.....	.....0.....	

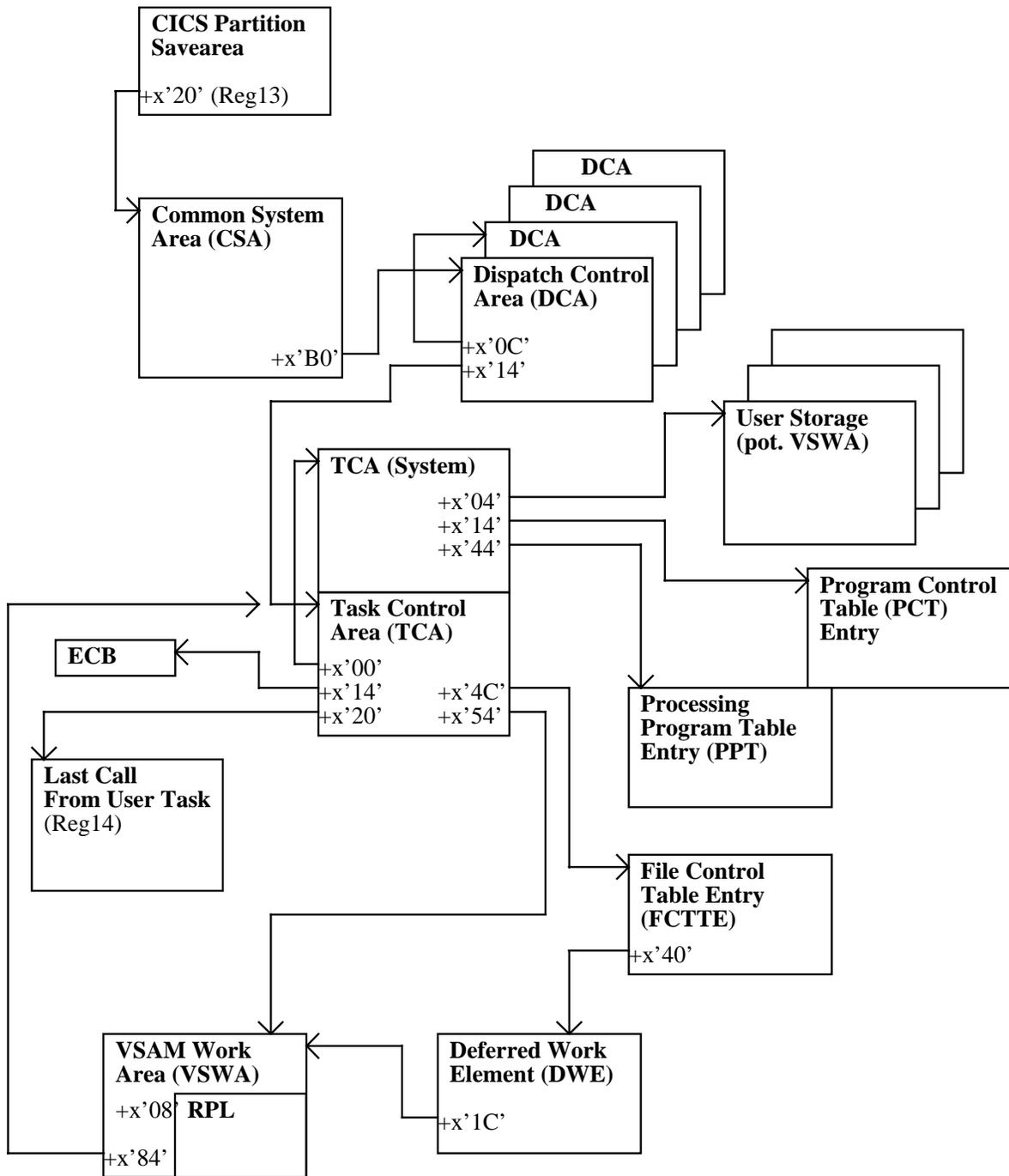
All VSAM OPEN and CLOSE activity for CICS file system managed files is performed by a VSE sub-task under control of DFHSKP. If DFHSKP is busy opening a file, and another transaction requests file open, this transaction is put into a wait until the previous open is complete. The TCA is queued in a QEA (Queue Element Area) chain, but the ECB pointer at TCA(U)+x'14' points at the beginning of the DFHSKP phase in CICS control storage. In this case, a "STATUS F2" command will probably show the DFHSKP sub-task waiting for some external event (e.g. a lock held by another partition).

### Waiting on terminal (user) response:

0051ECE0	0	.....	C1F0F0F2	91F20004	0051B2A0	A002j2.....	.....	
0051ECF0	10	0051B2A0	0050B1B0	00000000	0051EED4	....&.....	.....M	
0051ED00	20	FF000000	00000000	0C000000	00000100	.....	.....	
0051ED10	30	00000990	00007D14	0050AB4F	00000000	.....'	.....&.....	
0051ED20	40	00000000	00000000	00004080	07801850	.....	.....&	
0051ED30	50	0DEC1B84	D021A002	01B80000	00000000	...d.....	.....	

If a transaction is waiting for terminal I/O (response from operator), the ECB is actually the TCTTE (Terminal Control Table Terminal Entry).

# CICS/VSE Transaction Wait



# CICS/VSE Transaction Wait

CICS transaction dispatching occurs over a chain of DCAs (Dispatch Control Areas), which are chained off the CSA in two chains: Active and Suspended. Follow pointer in reg13 to CSA.

## CICS Partition Save Area (partition start):

500000	0	C4C6C8E2	C9D74040	07BD0000	0050B972	DFHSIP	.....
500010	10	00000000 <sup>9</sup>	00624584 <sup>A</sup>	007EF9B0 <sup>B</sup>	005080F0 <sup>C</sup>	.....d.=9.....0	
500020	20	00506810 <sup>D</sup>	00000000 <sup>E</sup>	0050D888 <sup>F</sup>	00DBB200 <sup>0</sup>	.....Qh.....	
500030	30	005CC000 <sup>1</sup>	005081F0 <sup>2</sup>	005068B4 <sup>3</sup>	0050ADE0 <sup>4</sup>	.*. ....a0.....	
500040	40	0050BDE0 <sup>5</sup>	005EE0FC <sup>6</sup>	005CC10C <sup>7</sup>	005CC3FC <sup>8</sup>	.....;*A...*C.	
500050	50	0000AAF5	17D05CD9	4E0AAF81	9F8A3F7C	...5...*R+..a....	
500060	60	422685F3	B645A1CA	40404040	40404040	..e3.....	
500070	70	40404040	40404040	5CC4C6C8	E2C9D740		*DFHSIP
500080	80	C4C6C840	0000661F	5CC4C6C8	E3C3E3C7	DFH	....*DFHTCTG
500090	90	E25C5000	A8F0F2F2	F0C91738	0510D7C9	S*..y0220I....PI	
5000A0	A0	C4D3C5E5	C5D30000	00DBB390	00DBB3C0	DLEVEL.....	

## Locate CSA:

Current TCA

506810	0	00000000	00000000	00DCBA00	70509132	.....j.	
506820	10	0084F000	801A1C6C	00E0C700	005081F0	.d0.....G...a0	
506830	20	701A0AFE	505090FA	0050A0FA	00787808	.....	
506840	30	00578338	7057BB2C	0056BF28	007766D0	..c.....	
506850	40	00787800	00507298	8011999C	0084F1A0	.....q..r..d1.	
506860	50	1707183F	00780C00	03E80100	00000000	.....Y.....	
506870	60	005E0D7C	0005DC00	00000000	00014D34	;;.....(.	
506880	70	00004000	00500000	031FFFFFF	0095114F	.. ..n.	
506890	80	005080F0	E0FFFFFFE	0000004B	00DC2480	...0.....	
5068A0	90	00008EFE	00000000	005EBBA0	C552C622	.....;..E.F.	
5068B0	A0	00000000	00692F70	005EE1C0	00711520	.....;.....	
5068C0	B0	005EE0E0	00507018	00507298	C500FF00	;;.....qE...	
5068D0	C0	00460042	01000400	00506C10	00C800A3	.....H.t	
5068E0	D0	00000000	00000000	00502064	00000000	.....	
.							
.							
.							
5069B0	1A0	07FE58F0	D19C07FF	011A2874	011A28B6	...0J.....	
5069C0	1B0	00177000	001C0300	E6D6D9D2	C1D9C5C1	.....WORKAREA	
5069D0	1C0	00000000	342C0000	063C205C	12440C00	.....*	

Max Tasks i      End of Active Task Queue      End of Suspended Task Queue

- CSA: +x'A8' End of DCA suspend queue. Normally, transactions experiencing difficulties are located on the active queue. Number of active tasks (in decimal)
- +x'B0' End of DCA active queue.
  - +x'B4' top of DCA active queue (highest priority transaction). I start with the end of the queue and work backwards, although you could just as easily start at the top and work forward.
  - +x'C0' Maximum active tasks (hex)
  - +x'1C8' Current number of active tasks (decimal)

If current number of active tasks is not less than maximum active tasks, the task you are waiting for may be waiting for a slot to run. Normally, in this case, you should check the other tasks in the active chain to see why they are hanging.

# CICS/VSE Transaction Wait

## First DCA:

5EE0E0	0	81000070	00000000	00000000	005EE3F0	a.....;T0
5EE0F0	10	005068B4	005F61A0	80400080	00DBC2C4	...../.. ..BD
5EE100	20	00000000	00000000	00000000	00000000	.....

'81' is Storage Accounting id for a DCA.

DCA+x'0C': **Backward pointer** (which we follow since we are running the chain backwards).

DCA+x'10': Forward chain pointer (points back at CSA at end of queue).

DCA+x'14': Points at the **TCA**

DCA+x'18': Contains the DCI (**transaction status**, same as TCA+x'18').

Common transaction status' are: '1008' Non-dispatchable

'8000' Currently dispatched

'8040' Wait on a single ECB

'80C0' Wait on a single ECB (termination requested)

'8840' Wait on multiple ECB

'4440' TCP transaction

'1340' Wait on terminal

'2020' Ready to be dispatched

'1400' Cannot be dispatched. CICS at max-task

## Look at TCA (User):

		System TCA	ECB Address	Task Status		
5F61A0	0	005F6000	00000000	00DBC2A8	00506C10	..-.....By...
5F61B0	10	005F6398	00DBC2C4	80400000	00818000	...q..BD. ...a..
5F61C0	20	5059F316	00508CD8	00000000	005F64C0	..3....Q.....
5F61D0	30	005F6398	4059F1A0	005F6330	00000000	...q .1.....

Remember, User TCA+x'00' points back at System TCA.

Note the transaction status at User TCA+x'14'. This matches what we just saw in the DCA.

User TCA +x'14' points at the ECB on which we are waiting, and User TCA+x'20' contains **Reg14** (bal reg for "DFHKC TYPE=WAIT")

## Look back at System TCA:

		Transaction Id	Storage Chain			
5F6000	0	8A040728	005F6000	005F6730 005F6730	.....-.....	
5F6010	10	0000017C	00DBC4FC	005EE0E0	00000000	.....D...;
5F6020	20	005EE9A0	00000000	00000000	00000000	;Z.....
5F6030	30	00000000	00000000	00000000	00000000	.....
5F6040	40	00000000	005F62A0	00000000	00000000	.....
5F6050	50	00000000	00000000	00000000	00000000	.....

'8A' is Storage Accounting id for TCA. System TCA+x'04' points to beginning of transaction storage chain. System TCA+x'11' contains the transaction id (normally packed decimal), and System TCA+x'14' points at **PCT** entry. System TCA+x'44' sometimes points at current **PPT** entry.

## CICS TS File Control Waits

Symptom: Transaction hang under CICS TS

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

```
cemt inq task
STATUS: RESULTS - OVERTYPE TO MODIFY
  Tas(0000045) Tra(TST1) Fac(L77A) Sus Ter Pri( 001 )
? Tas(0000048) Tra(TST1) Fac(L77B) Sus Ter Pri( 001 )
  Tas(0000056) Tra(CEMT) Fac(L77C) Run Ter Pri( 255 )

  Tas(0000048) Tra(TST1) Fac(L77B) Sus Ter Pri( 001 )
    Hty(FCXCWAIT) Hva(KSDSP ) Hti(009252) Sta(T0)
    Use(CICSUSER) Rec(X'B206AC118B261A82')
```

```
msg f2,data='cemt inq ta'
```

```
F2 0065 Tas(0000045) Tra(TST1) Fac(L77A) Sus Ter Pri( 001 )
      Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000339) Sta(T0)
      Use(CICSUSER) Rec(X'B206ABFA9EB37100')

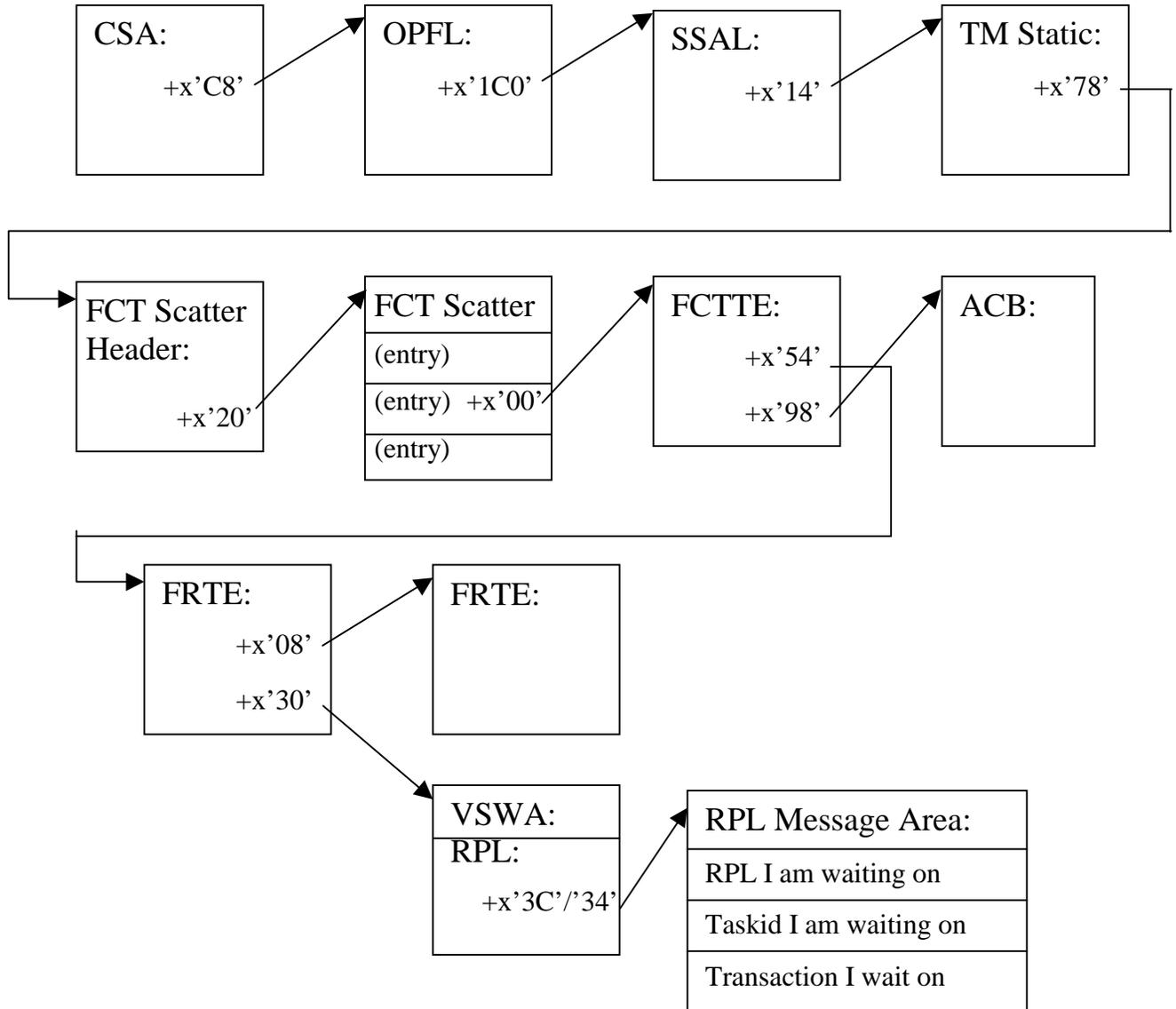
      Tas(0000048) Tra(TST1) Fac(L77B) Sus Ter Pri( 001 )
      Hty(FCXCWAIT) Hva(KSDSP ) Hti(000316) Sta(T0)
      Use(CICSUSER) Rec(X'B206AC118B261A82')

      Tas(0000050) Tra(CEMT) Fac(SYS ) Run Ter Pri( 255 )
      Use(CICSUSER) Rec(X'B206AD4FB551D682')
```

*“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” (GC33-1663-00)*

# CICS TS File Control Waits

Who am I waiting on?



## CICS TS File Control Waits

```

locate f2,'aica
AR 0015 MATCH FOUND AT 00684D78
V00684D70 ..... C1C9C3C1 1C000000 B6 * .....AICA....* R0064CD70
V00684D80 E2E3D6D9 C1C7C540 00000248 0060D8A0 B6 *STORAGE .....-Q.* R0064CD80
V00684D90 00620200 8139CFCE 001EDD38 80800000 B6 * .....a.....* R0064CD90
V00684DA0 012D2130 01F03018 00000000 01F19900 B6 * .....0.....1r.* R0064CDA0
AR+0015
15 e

show f2,684d88.200
V00684D80 ..... 00000248 0060D8A0 B6 * .....-Q.* R0064CD80
V00684D90 00620200 8139CFCE 001EDD38 80800000 B6 * .....a.....* R0064CD90
V00684DA0 012D2130 01F03018 00000000 01F19900 B6 * .....0.....1r.* R0064CDA0
V00684DB0 0000020C 00000000 8139CB60 012B4440 B6 * .....a..-...* R0064CDB0
V00684DC0 012D2130 012D15E8 8068BC70 01269600 B6 * .....Y.....o.* R0064CDC0
V00684DD0 0010005C 01269600 1424578F 0171E098 B6 * ...*.o.....\q* R0064CDD0
V00684DE0 00000100 018B8200 004F309D 00000000 B6 * .....b..|.....* R0064CDE0
V00684DF0 006E4C80 0000D57A 00000000 00600000 B6 * .><...N:.....-* R0064CDF0
V00684E00 01FFFFFF 0099090F 00652000 FFFFFFFE B6 * .....r.....* R0064CE00
V00684E10 0000B0B0 81E9F3A0 00000061 00000000 B6 * .....aZ3...../* R0064CE10
V00684E20 001E001E C564C641 81364450 81F9D4D0 B6 * .....E.F.a..&a9M}* R0064CE20
V00684E30 8136F650 81359D24 813570C0 0068C338 B6 * a.6&a...a..{..C.* R0064CE30
V00684E40 01269600 0060BA00 00692C00 00000000 B6 * ..o..-.....* R0064CE40
V00684E50 00684648 00000000 00000000 00000000 B6 * .....* R0064CE50
V00684E60 00000000 00000000 00000204 00700000 B6 * .....* R0064CE60
V00684E70 02000000 01E00EE4 00000042 01E16020 B6 * .....\.U.....-* R0064CE70

```

CSA+x'C8' → Optional Feature List (OPFL)

```

V00684640 ..... 00000000 00000000 B6 * .....* R001E6640
V00684650 01262FD0 00000000 00000000 00000000 B6 * ...}* R001E6650
V00684660 00000000 00000000 812431D0 812297F0 B6 * .....a..}a.p0* R001E6660
V00684670 00000000 00000000 00000000 00000000 B6 * .....* R001E6670
V00684680 8135DBB0 00000000 813C6C90 813C7510 B6 * a.....a.%a...* R001E6680
V00684690 813C7FD0 81364840 81388720 81388A30 B6 * a."}a.. a.g.a...* R001E6690
V006846A0 806C61C0 00000000 00000000 01F4F15B B6 * .%/ { .....41$* R001E66A0
.
.
V006847F0 8151FC78 00000000 813E1D70 814CF790 B6 * a.....a...a<7.* R001E67F0
V00684800 00000000 00000000 01262000 01299000 B6 * .....* R001E6800
V00684810 01262220 8148D060 00000000 00000000 B6 * .....a..}-.....* R001E6810
V00684820 00000000 00000000 00000000 00000000 B6 * .....* R001E6820
V00684830 01285000 00000000 40000000 00000000 B6 * ..&.....* R001E6830

```

CSA OPFL + x'1C0' → Static Storage Address List

## CICS TS File Control Waits

### CSA OPFL + x'1C0' → Static Storage Address List

V01262000	01262F70	012621F0	01262FA0	00000000	B4	* .....0.....*	R005F8000
V01262010	00000000	0068E010	00000000	00000000	B4	* .....\.....*	R005F8010
V01262020	0068E330	00000000	01262070	0068E3B0	B4	* ..T.....T.*	R005F8020
V01262030	012620A0	00000000	0068E950	00000000	B4	* .....Z&....*	R005F8030
V01262040	00000000	01EA6590	00000000	0068E850	B4	* .....Y&.*	R005F8040
V01262050	0068E8E0	FFFFFFFF	00000000	00000000	B4	* ..Y\.....*	R005F8050

### TM\_STATIC + x'14' → Table Manager Static Storage

V0068E000	C4C6C8E3	D4D74040	00000000	00000000	B6	* DFHTEMP.....*	R00178000
V0068E010	00000000	00000000	00000000	00000000	B6	* .....*	R00178010
V0068E020	00000000	00000000	00000000	00000000	B6	* .....*	R00178020
V0068E030	00000000	00000000	00000000	00000000	B6	* .....*	R00178030
V0068E040	00000000	00000000	00000000	00000000	B6	* .....*	R00178040
V0068E050	00000000	00000000	00000000	00000000	B6	* .....*	R00178050
V0068E060	00000000	00000000	00000000	012CE430	B6	* .....U.*	R00178060
V0068E070	007F000A	00000100	0000000A	00000100	B6	* ".....*	R00178070
V0068E080	00000000	012C4118	012CE860	007F0003	B6	* .....Y-.".*	R00178080
V0068E090	00000100	00000003	00000100	00000000	B6	* .....*	R00178090
V0068E0A0	01297038	01262B40	0091001A	00000100	B6	* .....j.....*	R001780A0
V0068E0B0	0000001A	00000100	00000000	012892B8	B6	* .....k.*	R001780B0
V0068E0C0	012FE430	00910002	00000100	00000002	B6	* ..U..j.....*	R001780C0
V0068E0D0	00000100	00000000	012BF018	01291430	B6	* .....0.....*	R001780D0

### TM\_STATIC + x'78' → TM FCT Scatter Table Header

V012CE860	00000430	6EC4C6C8	E3D4E2C3	C1E3E3C5	B4	* ....>DFHTMSCATTE*	R00345860
V012CE870	D9400000	00000505	00200008	00000100	B4	* R .....	R00345870
V012CE880	01297000	01297018	01297078	00000001	B4	* .....	R00345880
V012CE890	00000000	00000000	00000000	00000000	B4	* .....	R00345890

### TM\_FCT\_SCATTER\_HDR+x'20' → First Entry header

V01297000	0FF86EC4	C6C8E3D4	C4C9D9E2	C5C74040	B4	* .8>DFHTMDIRSEG *	R0061D000
V01297010	00000000	00000000	012E7000	00000000	B4	* .....	R0061D010
V01297020	00000000	01297018	01297038	05208000	B4	* .....	R0061D020
V01297030	C4C6C8C3	E2C44040	012E70F0	00000000	B4	* DFHCSD ...0...*	R0061D030
V01297040	00000000	01297038	01297058	05208000	B4	* .....	R0061D040
V01297050	D2E2C4E2	40404040	012E71E0	00000000	B4	* KSDS ...\...*	R0061D050
V01297060	00000000	01297058	00000000	05208000	B4	* .....	R0061D060
V01297070	D2E2C4E2	D7404040	00000000	00000000	B4	* KSDSP .....	R0061D070

Scatter Table entries are x'20' long, with the entry name **at the end of the entry**.

+x'00': Pointer to corresponding FCTTE

+x'10': Chain pointer to next scatter table entry. The entries are chained together, and if the list is very long, you have to locate the forward chain pointer to the next set of table entries.

+x'18': Name of the resource (filename for FCTTE).

# CICS TS File Control Waits

## FCTTE:

V012E71E0	D2E2C4E2	D7404040	012EB040	00000000	B6	*KSDSP	...	* R0069C1E0
V012E71F0	00000003	00EABA0A	80004400	10000000	B6	*	.....	* R0069C1F0
V012E7200	40400010	00000002	00000000	00000000	B6	*	.....	* R0069C200
V012E7210	00000000	00000000	00000000	00000013	B6	*	.....	* R0069C210
V012E7220	00000019	00000002	00000000	B205A116	B6	*	.....~	* R0069C220
V012E7230	FEADA285	012EE000	012EC1E0	012EC280	B6	*..se.. \...A\..B.*		* R0069C230
V012E7240	012BB204	00000000	00000000	00000000	B6	*	.....	* R0069C240
V012E7250	00000000	0303A850	02000000	00010000	B6	*	.....y&.....	* R0069C250
V012E7260	00050004	00000000	00000000	00000003	B6	*	.....	* R0069C260
V012E7270	00C80000	00060005	00699000	40000000	B6	*.H.....	...	* R0069C270

FCTTE +x'54' → FRTE (File Request Thread Entry)  
 +x'98' → VSAM ACB (In File Control Stack Storage)

V012EE000	00000000	00000000	012EE050	00000000	B6	* .....	\&....	* R001FB000
V012EE010	03000000	00000000	00000000	00000000	B6	*	.....	* R001FB010
V012EE020	00000000	012ED000	012E71E0	012EC280	B6	*	.....}	.... \..B.* R001FB020
V012EE030	0069F1D0	00000000	00000000	00000000	B6	* ..1}	.....	* R001FB030
V012EE040	00000000	00000000	00000000	00000000	B6	*	.....	* R001FB040
V012EE050	00000000	00000000	00000000	012EE000	B6	*	.....	\.* R001FB050

FRTE exists for each active file request. They are chained both by task (+x'00') and by resource (+x'08')  
 FRTE+x'30' → VSWA

V0069F1D0	8F0000E4	00000000	0011003C	00000000	B6	* .....	U.....	* R005DF1D0
V0069F1E0	0166B940	0166B718	000000C8	000000C8	B6	* .....	H...H*	* R005DF1E0
V0069F1F0	00699000	031C000A	99C00000	00080014	B6	*	.....r{.....	* R005DF1F0
V0069F200	00000000	01000001	00000000	0166DAA8	B6	*	.....y*	* R005DF200
V0069F210	00800000	00000000	012E71E0	0069F28C	B6	*	..... \..2.*	* R005DF210
V0069F220	0069F2A0	00140010	00000000	00000000	B6	* ..2.....	.....	* R005DF220
V0069F230	00000000	00000000	00000000	00000000	B6	*	.....	* R005DF230
V0069F240	0069F2C0	00000000	00000000	00000000	B6	* ..2{.....	.....	* R005DF240
V0069F250	0069F2C0	00900000	0000001C	00000000	B6	* ..2{.....	.....	* R005DF250
V0069F260	00000000	00693800	012EE000	00038000	B6	*	..... \.....	* R005DF260
V0069F270	00000000	00000000	00000000	D6D3E2C5	B6	*	..... OLSE*	* R005DF270
V0069F280	D5404040	404040D9	C5C3D6D9	012EC280	B6	* N	RECOR..B.*	* R005DF280

VSWA +x'08' VSAM Request List (RPL)  
 +x'3C' (RPL+x'34') → RPL message area  
 +x'94' → TCA  
 +x'BC' → DSNAME block

V0166DAA0	.....	.....	D9D7D3C1	C4C4D940	B6	* .....	RPLADDR	* R0049AAA0
V0166DAB0	7A404040	0069F2C8	40404040	40404040	B6	* :	..2H	* R0049AAB0
V0166DAC0	40404040	40404040	E3C1E2D2	D5E4D440	B6	*	TASKNUM	* R0049AAC0
V0166DAD0	7A404040	00069C40	40404040	40404040	B6	* :	...	* R0049AAD0
V0166DAE0	40404040	40404040	E3D9C1D5	E2C9C440	B6	*	TRANSID	* R0049AAE0
V0166DAF0	7A404040	E3E2E3F1	40404040	40404040	B6	* :	TST1	* R0049AAF0
V0166DB00	40404040	40404040	C6C3E3D5	C1D4C540	B6	*	FCTNAME	* R0049AB00
V0166DB10	7A404040	D2E2C4E2	D7404040	40404040	B6	* :	KSDSP	* R0049AB10
V0166DB20	40404040	40404040	E4F0F0F0	F0F0E7F2	B6	*	U0000072*	* R0049AB20

RPL Message Area +x'14' points at the RPL of the file which holds exclusive control.  
 I am waiting on task x'45', transaction 'TST1', on file 'KSDSP'.

## CICS TS File Control Waits

### Table Manager Static Storage:

Starting at +x'08, there are 17 x 28-byte entries pointing at each table which is under the control of Table Manager:

- 1 Reserved
- 2 Reserved
- 3 Reserved
- 4 PFT (Profile Table)
- 5 FCT (File Control Table)  
(4 x 28 = 112 = x'70' + x'08' = x'78')
- 6 DCT (Destination Control Table)
- 7 TCTE (Terminal Control Table)
- 8 TCTN (Terminal Table entry in terminal-owning address space)
- 9 TCTS (Terminal Table entry secondary index)
- 10 AFCT (Application File Control Table, owned by AP Domain,  
has corresponding FCTTE)
- 11 DSN (File Control Dataset Name, contains 44-char name for VSE file)
- 12 DSNA (Dataset Name Block, secondary index)
- 13 PRT (Partner Resource Table for APPC connections)
- 14 Reserved
- 15 TCNT (Terminal Control Netname Table)
- 16 AITM (AutoInstall Terminal Model Table)
- 17 Reserved

## CICS/VSE System Dumps (DFHDAP)

```
// JOB PRINT CICS/VSE DUMP FROM SYSDUMP LIBRARY
// LIBDEF *,SEARCH=(PRD1.BASE)
// EXEC PROC=DTRINFOA
// EXEC INFOANA,SIZE=300K
  SELECT DUMP MANAGEMENT
    DUMP NAME SYSDUMP.F2.DF200000
    RETURN
  SELECT DUMP VIEWING
    CALL DFHDAP
    RETURN
  SELECT END
/*
/ &
```

Formats the following CICS/VSE control blocks (>500 pages):

- CSA (and internal save areas)
- Active TCAs
- Active and Suspend DCA chain
- LIFO storage
- EIB storage
- FCT (File Control Table)
- SIT (System Initialization Table)
- Storage Control (page allocation map)
- Terminal Control (incl TCT and TWAs)
- Temporary Storage
- Transient Data (incl DCT)
- XRF
- Trace Table (internal)
- PPT (Processing Program Table)
- PCT (Transaction Table)
- FAQE
- SRT (System Recovery Table)
- LLA (Load List Area)
- DFHEMT0 phase (complete)
- VTAM control blocks (RPLs, Wait List, Exits)
- Static Storage
- DSG (Directory Segments)
- Scatter Tables
- All programs and control blocks sorted by name
- All programs and control blocks sorted by address

# CICS TS System Dumps (DFHPD410)

```
// JOB PRINT DUMP FROM SYSDUMP LIBRARY
// LIBDEF *,SEARCH=(PRD1.BASE,PRD2.SCEECICS,PRD2.SCEEBASE)
// EXEC INFOANA,SIZE=300K
  DUMP NAME SYSDUMP.VMEF2.DF200014
  SELECT DUMP SYMPTOMS
    PRINT DATA
    RETURN
  SELECT DUMP VIEWING
    CALL DFHPD410 DATA DEF=0,KE=3,DS,TR=1
    RETURN
  SELECT END
/*
/ &
```

===KE: Kernel Domain KE\_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB	ERROR
0001	01F11880	KTCB Step	00000000			00000000	01EF60B0	
0002	01F11500	KTCB QR	00000000			01EF6000	01EF8020	
0003	01F11180	KTCB R0	00000000			01E3F000	01EF7068	
0004	01F19C80	Unused						
0005	01F19900	Not Running	01269600	TCP	CSTP	01E24680	01EF8020	
0006	01F19580	Unused						
0007	01F19200	Not Running	00000000			01E04380	01EF8020	
0008	01F18E80	Not Running	01269C00	00020	CSNE	01E24880	01EF8020	
0009	012AAA80	Not Running	00692C00	00007	CSSY	01E24D80	01EF8020	
000A	0121A380	Unused						
000B	0121A700	Unused						
000D	0121AA80	Unused						
000E	012AA000	Unused						
000F	0128F380	Not Running	00693200	00019	CSSY	01E04480	01EF8020	
0010	012AA380	***Running**	00692000	00030	TST3	011F7080	01EF8020	*YES*
0012	01E27C80	Not Running	00692600	00006	CSSY	01E24B80	01EF8020	
0014	012AA700	Unused						
0019	0121A000	Not Running	00000000			01E04680	01EF8020	

-----  
 ==KE: KE Domain Error Table Summary

ERR_NUM	ERR_TIME	KE_NUM	ERROR TYPE	ERR_CODE	MODULE	OFFSET
=====	=====	=====	=====	=====	=====	=====
00000001	15:22:21	0010	PROGRAM_CHECK	0C1/AKEA	UNKNOWN	UNKNOWN
00000002	15:23:15	0010	TRAN_ABEND_PERCOLATE	---/ASRA	DFHSR1	00000380
00000003	15:23:15	0010	TRAN_ABEND_PERCOLATE	---/ASRA	DFHFCVS	00002EFE
00000004	15:24:02	0010	TRAN_ABEND_PERCOLATE	---/AFCZ	DFHPCP	000004EC
00000005	15:24:03	0010	TRAN_ABEND_PERCOLATE	---/AFCZ	DFHFCEI	000010D4
00000006	15:24:03	0010	TRAN_ABEND_PERCOLATE	---/AFCZ	DFHEIP	000003D4
00000007	15:25:48	0010	PROGRAM_CHECK	0CB/AKEA	DFHYI410	00000624

# CICS TS System Dumps (DFHPD410)

==KE: Task in Error; Error Data follows.

=KE: Error Number: 00000007

```

KERRD 012AA558 KERNEL ERROR DATA
      TCA      ILC, Interrupt Code
0000  F0C3C261 C1D2C5C1 010400CB 0000FFFF  C4C6C8C1 D7D3C9F1 013D9030 011F7080 *OCB/AKEA.....DFHAPLI1.....*
0020  00692000 012AA380 00000007 0000000B  07BD000B E0000000 07BD2000 81B02254 *.....t.....a...*
0040  0006000B 00000000 81B02254 B0000000  0165B60C 0165B678 00000000 0165B840 *.....a.....*
0060  01B01C88 0165B190 00710B9C 00000000  007C00D0 0165B710 01B01D78 01B02034 *...h.....@.....*
0080  01B01D44 0165B4C8 81B02188 00000000  00000000 00000000 00000000 00000000 *.....Ha..h.....*
00A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *.....PSW.....*
00C0  00000000 00000000 00000000 00000000  07BD000B E0000000 07BD2000 81B02254 *.....a.....*
00E0  0006000B 00000000 81B02254 B0000000  0165B60C 0165B678 00000000 0165B840 *.....a.....*
0100  01B01C88 0165B190 00710B9C 00000000  007C00D0 0165B710 01B01D78 01B02034 *...h.....@.....*
0120  01B01D44 0165B4C8 81B02188 00000000  00000000 00000000 00000000 00000000 *.....Ha..h.....*
0140  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *.....*
0160  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *.....*
0180  4E000000 034E0417 4E000000 0002522C  40404040 40404040 00000000 00000000 *+...+...+.....s.Z*...5&...*
01A0  00000000 00000000
  
```

```

Error Code:  OCB/AKEA      Error Type:  PROGRAM_CHECK      Timestamp:  B205A28DE95C4080
Date (GMT)   :  30/03/99      Time (GMT)    :  15:25:48.638660
Date (LOCAL) :  30/03/99      Time (LOCAL)  :  15:25:48.638660
KE_NUM:  0010      KE_TASK:  012AA380      TCA_ADDR:  00692000      DS_TASK:  011F7080
Error happened in program DFHYI410 at offset 00000624
  
```

Registers and PSW.

```

PSW:  07BD2000 81B02254  Instruction Length:  6  Interrupt Code:  0B  Exception Address:  00000000
Execution key at Program Check/Abend:  B
REGISTERS 0-15 (REGS at 012AA5A8)
0000  0165B60C 0165B678 00000000 0165B840  01B01C88 0165B190 00710B9C 00000000 *.....h.....*
0020  007C00D0 0165B710 01B01D78 01B02034  01B01D44 0165B4C8 81B02188 00000000 *.@.....Ha..h....*
  
```

# CICS TS System Dumps (DFHPD410)

Data at PSW: 81B02254      Module: DFHYI410      Offset: 00000624

PSWDATA 01B01C30

```

0000 C4C6C8E8 C9F4F1F0 58F00014 58F0F0B4 58F0F00C 58FF000C 07FF0000 00000000 *DFHYI410.0...00..00.....*
0020 47F0F028 00C3C5C5 00000000 00000014 47F0F001 4ACEAC00 01B01CFC 00000000 *.00..CEE.....00.¢.....*
0040 00000000 00000000 90ECD00C 4110F038 98EFF04C 07FF0000 01B01C50 01B01D44 *.....0.q.0<.....&....*
0060 01B02658 01B01CB0 01B01C50 01B020B2 01B02CE0 01B01D10 00000000 00000007 *.....&.....*
0080 C3D6C2E3 C5E2E340 F1F9F9F9 F0F3F2F9 F1F1F0F4 F0F7F0F1 F0F1F0F0 00000000 *COBTEST 19990329110407010100....*
00A0 00000000 60E87C4C 00000000 10040209 00000000 09000000 00800000 00000063 *....-Y@<.....*
00C0 0000003E 00008000 40404040 05000001 01B02C60 00000000 FFFFFFFB 01B01C50 *.....-.....&*
00E0 01B01C88 00000008 00000006 01B01C88 01B02C60 01B02878 00000005 00000000 *...h.....h...-.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 40404040 40404040 *.....*
0120 40404040 40404040 40404040 40404040 40404040 40404000 00000C00 000F4040 *.....*

```

## Data at Registers

REG 0 0165B60C

31-bit data follows:

REGDATA 0165B60C

```

-0080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
-0060 00000000 01B01D44 00000001 0165B664 0165B3F8 01B01FCF 00000000 81B01C50 *.....8.....a.&*
-0040 01B01D7C 0165B620 01B01D74 00000000 0165B710 00000000 00000000 00000000 *...@.....*
-0020 00000000 00000000 00000000 0165B6D8 0165B710 00000000 007C00D0 00000000 *.....Q.....@.....*
0000 00000000 00000000 00000000 00000000 00000000 01B020B2 01B02628 01B02628 *.....*
0020 01B00A8E 01B00A8E 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 01B00A8E 00000000 00000000 00000000 01B020B2 C0000000 00000000 *.....*
0060 00000000 00000000 00000000 00000F68 0165BC60 00080F60 0165B840 00000000 *.....-...-.....*
0080 080CB950 0165B4C8 0165B6D8 0165B710 0165B6E0 C8C1D5C3 01656860 01656860 *...&...H...Q.....HANC...-...-*
00A0 00000000 8165B6A0 0165BCC0 00001000 000009E0 0165B6A0 00000600 000005F8 *...a.....8*
00C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
00E0 00000000 E2E8E2D6 E4E34040 00000000 00000000 0E000000 00000000 0F000000 *....SYSOUT.....*

```

24-bit data follows:

REGDATA 0065B60C

```

-0080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
-0060 - 00FF LINES SAME AS ABOVE

```

## Combining SYSDUMPs for DFHPD410

**Problem:** Many VSE customer respond to an online system hang by taking a stand-alone dump. With a VSE stand-alone dump, each partition, and the supervisor is in a separate file on the tape, and will be restored to a separate dump in the SYSDUMP sublibrary. DFHPD410 requires certain supervisor and system GETVIS areas to be physically resident in the same dump file as the CICS TS partition.

```
SELECT DUMP VIEWING
      CALL DFHPD410 DATA DEF=0,KE=1,DS=1,TR=1,FCP=2

-- DFHPD0110 UNABLE TO ACCESS DATA FOR CRADDR
      ADDRESS 00000014 -- TESTING FOR DMF DUMP
-- DFHPD0115 CICS/EXCI/DMF JOB NOT FOUND.
** DFHPD0120 CICS DUMP ANALYZER EXIT IS TERMINATING.
```

### VSE SYSDUMP dump format:

DITTO/ESA for VSE		TB - Tape Browse			
Blk 1	File 1	from begin of tape	Col 1	Fmt	HEX
	VOL1	DSN	Uprotect	18 Trk	ASCII NO
Block	Length	1 . . . 5 . . . .10 . . . . 5 . . . .20 . . . . 5 . . .			
1	4112	00000001E2D9F9F0F2F1F0F3F0F4F1F1B2188505BEAC2782000000F			
2	4112	0000000200000000000000000000061AF1F100080000000009100200030			
3	4112	0000000200001000000001000061AF1F100312268003172680031826			
4	4112	0000000200002000000002000061AF1F1FFFFFFFFFFFFFFFFFFFFFFFFF			
5	4112	0000000200003000000003000061AF1F1FFFFFFFFFFFFFFFFFFFFFFFFF			
6	4112	0000000200004000000004000061AF1F1002EF8C0000000000000000			
7	4112	0000000200005000000005000061AF1F100000019000000000000000			
8	4112	0000000200006000000006000061A0000024107EE0B0E5800C004583			
9	4112	0000000200007000000007000061A000000000000000000000000000			
10	4112	0000000200008000000008000061A000000045AE6A4013B880001460			
11	4112	0000000200009000000009000061A0000C4D3E9C2E2C5D6E30000000			
12	4112	000000020000A00000000A000061A000004040404040404040404040			
13	4112	000000020000B00000000B000061A000040008810000441101003BF1			
14	4112	000000020000C00000000C000061A000096265890026CD5009053047			
15	4112	000000020000D00000000D000061A00004770D8EEB1E0A00F4770D8E			

## Combining SYSDUMPs for DFHPD410

- Step 1: Scan the stand-alone dump tape, and identify the dumps to be onloaded.  
Step 2: Onload the supervisor and CICS TS dumps into SYSDUMP.

Combine the two dumps:

- Step 3: Offload the supervisor and CICS TS dumps to two different tapes.  
Step 4: Using DFSORT, combine the two dumps onto a third tape.  
Step 5: Reload (onload) the combined dump back into SYSDUMP  
Step 6: Update low core to point to CICS TS partition.

### **Step 1: Scan the stand-alone dump** (and identify the dumps to be onloaded)

```
r rdr,pausebg
BG 0000 // JOB PAUSEBG
BG-0000 // PAUSE
0 // exec dosvsdmp
BG 0000 4G01D SELECT ONE OF THE FOLLOWING FUNCTIONS:

  1  CREATE STAND-ALONE DUMP PROGRAM
  2  SCAN DUMP TAPE/DISK
  3  PRINT DUMP TAPE/DISK
  4  PRINT SDAID TAPE
  5  PRINT IPL DIAGNOSTICS
  R  END DOSVSDMP PROCESSING
0 2
BG 0000 4G04D SPECIFY ADDRESS OF DUMP DEVICE (CUU OR SYSNNN)
BG-0000 0C66D READY

0 7A8
```

### **SYSLST Output:**

```
PRINTOUT OF VSE DUMP TAPE
DIRECTORY OF VSE DUMP TAPE
DUMP FILE  DUMP TYPE  NAME          DATE          DATA DUMPED
-----
  001                                DOES NOT CONTAIN DUMP DATA
  002                                DOES NOT CONTAIN DUMP DATA
  003      SADUMP                                SUPERVISOR+SVA
  004      SADUMP      99/04/14      PMRAS-R
  005      SADUMP      99/04/14      PMRAS-00
  006      SADUMP      T77VDA      99/04/14      Z1-PARTITION
  007      SADUMP      T77VEA      99/04/14      Z2-PARTITION
  008      SADUMP      T77VFA      99/04/14      Z3-PARTITION
  009      SADUMP      VSECOCCF   99/04/14      F4-PARTITION
  010      SADUMP      VSECVTIE   99/04/14      F3-PARTITION
  011      SADUMP      POWSTART   99/04/14      F1-PARTITION
END OF DUMP
```

## Combining SYSDUMPs for DFHPD410

### **Step 2: Onload the supervisor and CICS TS dumps into SYSDUMP.**

```
// JOB INFOANAL ONLOAD TWO DUMPS FROM TAPE
// ASSGN SYS009,7A8
// MTC REW,SYS009
// ASSGN SYS016,DISK,VOL=SYSWK1,SHR
// ASSGN SYS017,DISK,VOL=SYSWK1,SHR
// EXEC INFOANA,SIZE=300K
DUMP NAME SYSDUMP.HAYN.SUP_DUMP
SELECT DUMP ONLOAD
VOLID TST999 SYS009
FILE 3
RETURN
DUMP NAME SYSDUMP.HAYN.Z1_DUMP
SELECT DUMP ONLOAD
VOLID TST999 SYS009
FILE 6 LAST
RETURN
SELECT END
/*
/ &
```

### **Step 3: Offload the supervisor and CICS TS dumps to two different tapes.**

```
// JOB INFOANAL OFFLOAD TWO DUMPS TO TWO TAPES
// ASSGN SYS009,7A8
// MTC REW,SYS009
// ASSGN SYS010,7AA
// MTC REW,SYS010
// ASSGN SYS016,DISK,VOL=SYSWK1,SHR      BLNDMF
// ASSGN SYS017,DISK,VOL=SYSWK1,SHR      BLNXTRN
// EXEC INFOANA,SIZE=300K
DUMP NAME SYSDUMP.HAYN.SUP_DUMP
SELECT DUMP OFFLOAD
VOLID TST999 SYS009
RETURN
DUMP NAME SYSDUMP.HAYN.Z1_DUMP
SELECT DUMP OFFLOAD
VOLID TST999 SYS010
RETURN
SELECT END
/*
/ &
```

## Combining SYSDUMPs for DFHPD410

### Step 4: Using DFSORT, combine the two tapes onto a third tape

This job took about 10 minutes to run, and required a work area of 733 tracks.

```
// JOB COPY ONE RECORD FROM Z1 TO OUTPUT TAPE
// ASSGN  SYS001,X'7A8'      SORT OUTPUT
// MTC REW,SYS001
// MTC WTM,SYS001,10
// MTC REW,SYS001
// ASSGN  SYS002,X'7AB'      CICS DUMP
// MTC REW,SYS002
// ASSGN  SYS003,X'7AA'      SUPERVISOR DUMP
// MTC REW,SYS003
// DLBL SORTWK1,,1,DA
// EXTENT SYS005,VSEUC4,1,0,22601,800
// ASSGN  SYS005,DISK,VOL=VSEUC4,SHR
// LIBDEF *,SEARCH=PRD3.PROD
* COPY ONE RECORD FROM CICS DUMP TO OUTPUT TAPE
// EXEC   SORT,SIZE=64K
        OPTION LABEL=(U,U,U),STOPAFT=1
        SORT FIELDS=(1,16,BI,A),FILES=1
        RECORD TYPE=F,LENGTH=(4112,,4112)
        OUTFIL NOTPMK
/*
// MTC REW,SYS001
// MTC FSR,SYS001,1
* SORT BOTH INPUT TAPES TO OUTPUT TAPE
* REPLY "0 DELETE" TO MSG4733D ON SORTWK1
// EXEC SORT,SIZE=64K
        OPTION LABEL=(U,U,U)
        SORT FIELDS=(1,16,BI,A),FILES=2
        RECORD TYPE=F,LENGTH=(4112,,4112)
        OMIT COND=(1,4,BI,EQ,X'00000001')
        OUTFIL OPEN=NORWD,NOTPMK
/*
/&
```

AR DUMP tapes may be processed in the same manner, except three tapes need to be merged: supervisor, SVA, and CICS TS partition. Add an ASSGN and MTC REW for SYS004, and change "FILES=2" to "FILES=3" in the second jobstep.

## Combining SYSDUMPs for DFHPD410

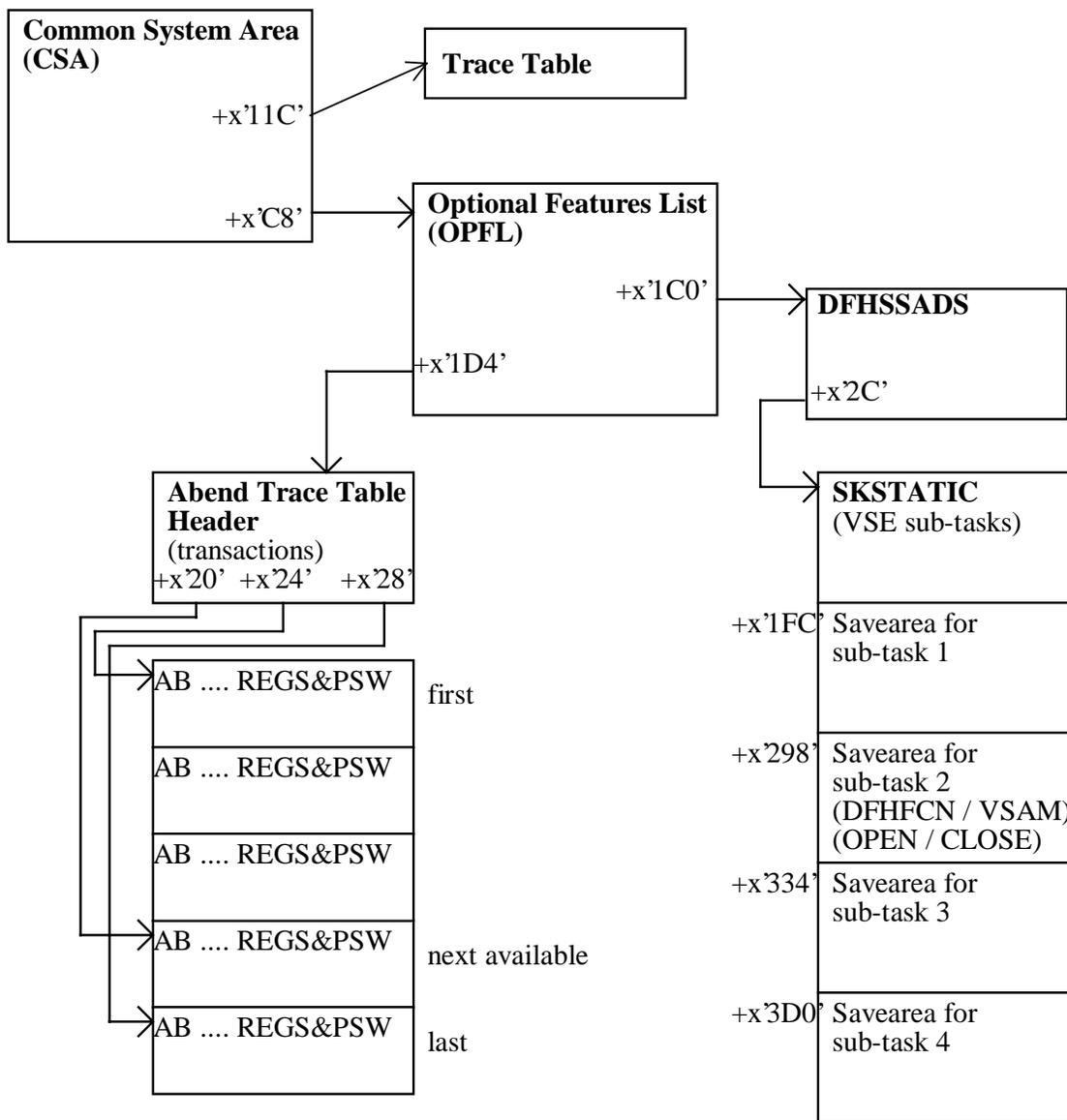
### **Step 5: Reload (onload) the new combined dump into SYSDUMP**

```
// JOB INFOANAL ONLOAD NEW DUMP FROM TAPE
// ASSGN SYS009,7A8
// MTC REW,SYS009
// ASSGN SYS016,DISK,VOL=SYSWK1,SHR
// ASSGN SYS017,DISK,VOL=SYSWK1,SHR
// EXEC INFOANA,SIZE=300K
DUMP NAME SYSDUMP.HAYN.NEW_DUMP
SELECT DUMP ONLOAD
VOLID TST999 SYS009
FILE 1
RETURN
SELECT END
/*
/ &
```

# CICS/VSE Abend Trace Table

## Locating Abend Information in unformatted dump:

Obviously, IDUMPs in the VSE SYSDUMP library should be formatted using DFHDAP (See Chapter 17). However, if CICS was unable to issue an IDUMP (and issued a PDUMP to the POWER print queue), or the VSE SYSDUMP library is full, and the IDUMP went to the POWER print queue, do not panic. It is still possible to find transaction abend information in a raw dump.



## CICS/VSE Abend Trace Table

There are actually two sets of CICS Abend Save Areas which we will cover today. One for transaction abends, and the other for sub-task abends. Both Abend Save Area are addressed over the Optional Features List (CSA+x'C8'): Locate CSA by searching for "AICA STORAGE" eyecatcher. CSA starts at +x'10' past the eyecatcher. CSA+x'C8' points at the Optional Features List (OPFL). OPFL+x'1C0' points at VSE sub-task save areas. OPFL+x'1D4' points at CICS Abend Trace Table.

### CSA:

801630	-10	C1C9C3C1	1C000000	E2E3D6D9	C1C7C540	AICA....STORAGE	
801640	00	00000000	00000000	00A0F200	709A75F6	.....2....6	
801650	10	008CEDDC	B09A759C	00800598	009A7850	.....q....	
801660	20	009FE2D4	008F4010	00E8D500	008000A8	..SM.. ..YN....y	
801670	30	480003C8	008F34D0	009048F4	409A6850	...H.....4 ...	
801680	40	00800B98	008F51A0	001E999C	008F51A0	...q.....r....	
801690	50	0942446F	00000000	03E80100	00000000	...?.....Y.....	
8016A0	60	00355A10	00012C00	00000000	0000852B	..!.....e.	
8016B0	70	00004000	00800000	00DFFFFF	0095166F	.. .....n.?	
8016C0	80	00802F20	E0FFFFFFE	0000004B	00A053D0	.....E.F.	
8016D0	90	00000009	00000000	008F0E30	C552C622	.....E.F.	
8016E0	A0	00000000	008F18E0	008016DC	008016DC	.....HE...	
8016F0	B0	008F1870	00801E48	008020C8	C500FF00	.....HE...	
801700	C0	00140000	01000400	00801A40	00C800C0	.....H..	
801710	D0	00000000	00000000	00000000	00000000	.....<	
801720	E0	40803F24	00854C48	0085F808	038981A8	....e<..e8..iay	

### OPFL (Optional Features list):

801A40	00	00000000	00000000	00000000	008B4E68	.....+.	
801A50	10	0088F5A2	00000000	00899838	008998CC	.h5s....iq..iq.	
801A60	20	008C6B3C	008C7EE8	00000000	00000000	..,..=Y.....	
801A70	30	00000000	0089BB78	00000000	00000000	.....i.....	
801A80	40	00100000	00000000	00000000	00000000	.....<	
801A90	50	008CC3CC	00000000	00000000	00000000	..C.....	
.							
.							
801B70	130	008944C2	0088B842	0088D038	00891D72	.i.B.h...h...i..	
801B80	140	0088B032	008903A6	00893E72	008973A8	.h...i.w.i...i.y	
801B90	150	00892BD2	008CB538	0085E94C	00000000	.i.K.....eZ<....	
801BA0	160	00000000	0085202C	008D1D2C	00813CAC	.....e.....a..	
801BB0	170	00808BC8	00000000	0021D77C	008086B8	...H.....P...f.	
801BC0	180	008DC838	00861A90	008B8B28	0080FA9C	..H..f.....	
801BD0	190	008A5EA8	008A3A8C	008C821C	008C9E3C	..;y.....b.....	
801BE0	1A0	0089CCF8	00000000	009E4008	00000000	.i.8.....	
801BF0	1B0	0089F988	008A16F8	009FF670	00801D68	.i9h...8..6.....	
801C00	1C0	00A052E0	009048F4	009FDE30	008D17B8	.....4.....	
801C10	1D0	00000000	00CB1B28	00000000	00000000	.....<	
801C20	1E0	00000000	00000000	008DFCB8	008F4CD0	.....<	
801C30	1F0	40000000	00000000	00000000	00000000	.....<	

DFHSSA Pointer

Abend Trace Table Pointer

## CICS/VSE Abend Trace Table

### (Transaction) Abend Trace Table:

OPFL + x'1D4' points to the Abend Trace Table. This is used to store PSW and registers for CICS transactions which abend. Please note that most (if not all) the transaction storage will have already been freed, including the TCA and File I/O Work Areas.

CB1B28	0	D7D9D6C7	D9C1D440	C3C8C5C3	D261C1C2	PROGRAM CHECK/AB
CB1B38	10	C5D5C440	E3D9C1C3	C540E3C1	C2D3C57A	END TRACE TABLE:
CB1B48	20	00CB1BE8	00CB1B68	00CB1E68	00000000	...Y.....
CB1B58	30	00000001	00000340	00000000	00000000	.....
CB1B68	40	C1C20001	0074E9A0	00767830	00629580	AB....Z.....n.
CB1B78	50	C8000400	0043485C	505B5388	693BDC48	H.....*\$.h....
CB1B88	60	00000000	00000000	D9C5C7E2	50D7E2E6	.....REGS&PSW
CB1B98	70	0000000B <sup>0</sup>	00D2BE68 <sup>1</sup>	5018FAA4 <sup>2</sup>	00D2BE44 <sup>3</sup>	.....K.....u.K..
CB1BA8	80	0019A2D8 <sup>4</sup>	00CF1A00 <sup>5</sup>	00D2BE68 <sup>6</sup>	00008800	..sQ.....K....h.
CB1BB8	90	00CF1808 <sup>8</sup>	4018F948 <sup>9</sup>	00DC0DD8 <sup>A</sup>	00DC0A18 <sup>B</sup>	.....9....Q....
CB1BC8	A0	0074F708 <sup>C</sup>	00CF1808 <sup>D</sup>	8019A364 <sup>E</sup>	00000422 <sup>F</sup>	..7.....t.....
CB1BD8	B0	07BD2000	0019A4CE	0002006E	00000000	.....u.....
CB1BE8	C0	00000000	00000000	00000000	00000000	.....

The first three pointers following the eyecatcher are: **Next-available-entry**, **First-entry**, **Last-Entry**. Each entry is x'80' bytes long, and starts with two characters identifying the type of entry ("AB" = ASRB, "PC" = ASRA). Following the "REGS&PSW", are the registers 0 through 15, then the **PSW**. Following the PSW is the VSE cancel information. (2 bytes **ILC**, 2 bytes **VSE cancel code**)

When CICS issues a system dump, the "Next available entry" pointer normally points at the entry which caused the abend.

# CICS/VSE Abend Trace Table

Actually, the Program Check / Abend trace table is part of a larger structure, which also includes the STXIT PC and STXIT AB areas. These are all contained in the DFHSRP static storage, which can be accessed as follows: OPFL+X'1C0' points at the Static Storage Area (SSA). SSA+x'08' points at DFHSRP static.

DFHSRP Static +x'D8': STXIT PC area  
 +x'1D0': STXIT AB area  
 +x'2A8': Program Check / Abend Trace Table

## SSA (OPFL + x'1C0):

V00D782F0	0	00D77C10	00D75070	00D77DC0	00D72790	.P...P...P'..P..
V00D78300	10	00000000	00D781B0	00D77DE0	00D73010	.....Pa..P'..P..
V00D78310	20	00D72720	00D72670	00D72410	00D71F30	.P...P...P...P..
V00D78320	30	00D77C90	00D71A00	00D719C0	00D723B0	.P...P...P...P..

## DFHSRP Static Storage (SSA+x'0C):

V00D72780	-10	C4C6C8E2	D9D74040	00C0864	00000000	DFHSRP .....
V00D72790	0	07AD0007	E0000000	006F5050	008B9E58	.....?.....
V00D727A0	10	408B91C2	008B8C34	006FF054	408BA14E	.jB.....?0. ..+
V00D727B0	20	006F5050	006F5AE7	006F5AE8	008BA10C	.?..?!X?!Y....
V00D727C0	30	008B8038	008B8038	008B8C20	006F58C8	.....?..H
V00D727D0	40	806F5B58	00000000	07AD2000	008B9E6E	.\$.....

STXIT PC area

V00D72860	D0	00000000	00000000	07AD0007	E0000000	.....
V00D72870	E0	00717EA0	00C92658	40C919C2	00C91434	..=.I.. I.B.I..
V00D72880	F0	00717054	40C9294E	00717EA0	00718937	.... I.+..=.i.
V00D72890	100	00718938	00C9290C	00C90838	00C90838	..i..I..I..I..
V00D728A0	110	00C91420	00718718	806F0308	00000000	.I....g..?.....
V00D728B0	120	07AD2000	00C9266E	00060007	00000000	.....I.....
V00D728C0	130	00000000	00000000	00000000	00000000	.....

STXIT AB area

V00D72950	1C0	40404040	40404040	40404040	40404040	.....
V00D72960	1D0	07AD0000	60000000	8019F2C4	00F33040	....-.....2D.3.
V00D72970	1E0	00000000	021CE878	00F33040	00000420	.....Y..3. ....
V00D72980	1F0	021FFA98	00F852BF	021FFA98	00F85280	...q.8....q.8..
V00D72990	200	00F04600	00F85268	001A0B28	02217018	.0...8.....
V00D729A0	210	8019ED4E	0019F635	07AD2000	801A0BF2	...+..6.....2
V00D729B0	220	00020000	00000000	00000000	00000000	.....

V00D72A30	2A0	00000000	00000000	D7D9D6C7	D9C1D440	.....PROGRAM
V00D72A40	2B0	C3C8C5C3	D261C1C2	C5D5C440	E3D9C1C3	CHECK/ABEND TRAC
V00D72A50	2C0	C540E3C1	C2D3C57A	00D72B78	00D72A78	E TABLE:.P...P..
V00D72A60	2D0	00D72D78	00000139	00000001	00000340	.P.....
V00D72A70	2E0	00000000	00000000	D7C30139	006FE9B0	.....PC...?Z.
V00D72A80	2F0	006FF2D0	022DC800	E101F400	0025144C	.?2...H...4....<
V00D72A90	300	508B91B0	4240961E	00000000	00000000	..j.. o.....
V00D72AA0	310	D9C5C7E2	50D7E2E6	006F5050	008B9E58	REGS.PSW?... ..
V00D72AB0	320	408B91C2	008B8C34	006FF054	408BA14E	.jB.....?0. ..+

## CICS/VSE Abend Trace Table

### CICS Sub-task (DFHSKP) Abend Save Areas:

CICS performs much of its concurrent system activity (e.g. VSAM file OPENs and CLOSEs) via VSE sub-tasks. When the VSAM OPEN/CLOSE sub-task cancels, msgDFH0959 is issued:

```
DFH0959 OPEN OR CLOSE OF FILE filename FAILED. CICS DETECTED ERROR - eeee
```

eeee identifies where, within CICS/VSE File Control, the error was detected. See the description of abend code AFCI in "Transaction Abend Codes" in topic 10.2 of "VSE/ESA Messages and Codes".

The CICS driver module for these sub-tasks is DFHSKP. DFHSKP is actually comprised of three modules: DFHSKM, DFHSKC, and DFHSKE. DFHSKE is the actual module that drives the sub-tasks, and it issues its own STXITs (AB and PC). OPFL + x'1C0' points to DFHSSADS. SSA+x'2C' points at the SKSTATIC area, which contains 4 save areas (x'9C' long), one for each of the possible sub-tasks driven by DFHSKE. The task save areas start at SKSTATIC+x'1FC'. File Control uses task 2. Since sub-tasks use mostly partition GETVIS, most of the tables and work areas should still be available.

### SSA (OPFL + x'1C0'):

A052E0	0	00A04C00	00A02060	00A04DB0	009FF780		..<....-... (...7.	
A052F0	10	00000000	00A051A0	00A04DD0	00A00000		..... (...	
A05300	20	009FF710	009FF660	009FF400	009FEF20		..7...6-...4.....	
A05310	30	00A04C80	009FE9F0	009FE9B0	009FF3A0		..<...Z0..Z...3.	
A05320	40	00000000	00A04FB0	00000000	009FE750		..... .....X.	

### SKSTATIC (SSA+x'2C'):

9FEF10	-10	C4C6C8E2	D2D74040	002C046D	00000000		DFHSKP	..._.....	
9FEF20	0	009FEF28	00000000	009FEF8C	0091E580		.....jV.		
9FEF30	10	009F16D8	000000C8	0090898C	0090899C		...Q...H...i...i.		
9FEF40	20	008A457E	00908804	009FF1B8	00000000		...=.h...1.....		
9FEF50	30	009FEF28	00000003	00AFF514	00000000		.....5.....		

Save Area for Task 2

9FF1B0	290	00000000	00000000	008A457C	00000000		.....	
9FF1C0	2A0	00000000	00000000	00000000	00000000		.....	
9FF1D0	2B0	00000000	00008000	40000000	40000000		.....	
9FF1E0	2C0	00000000	10000000	009FF1D0	FF000000		.....1.....	
9FF1F0	2D0	00000000	00000000	00000000	00000000		.....	
9FF200	2E0	00000000	00002C00 <sup>0</sup>	00B80000 <sup>1</sup>	00B08818 <sup>2</sup>		.....h.	
9FF210	2F0	00AFA620 <sup>3</sup>	00AEC900 <sup>4</sup>	72000048 <sup>5</sup>	00001200 <sup>6</sup>		..w...I.....	
9FF220	300	00B81200 <sup>7</sup>	00000003 <sup>8</sup>	00AFF514 <sup>9</sup>	00B80000 <sup>A</sup>		.....5.....	
9FF230	310	40188C84 <sup>B</sup>	00AED390 <sup>C</sup>	00AED390 <sup>D</sup>	7018909E <sup>E</sup>		..d..L...L.....	
9FF240	320	00000000 <sup>F</sup>	073D1000	00189198	00040004		.....jq....	
9FF250	330	00000000	00000000	00000000	00000000		.....	

Registers 0 - 15 are found at +x'4C' into sub-task 2 save area, followed by the **PSW**. and **VSE Cancel Info**



## CICS/VSE Abend Trace Table

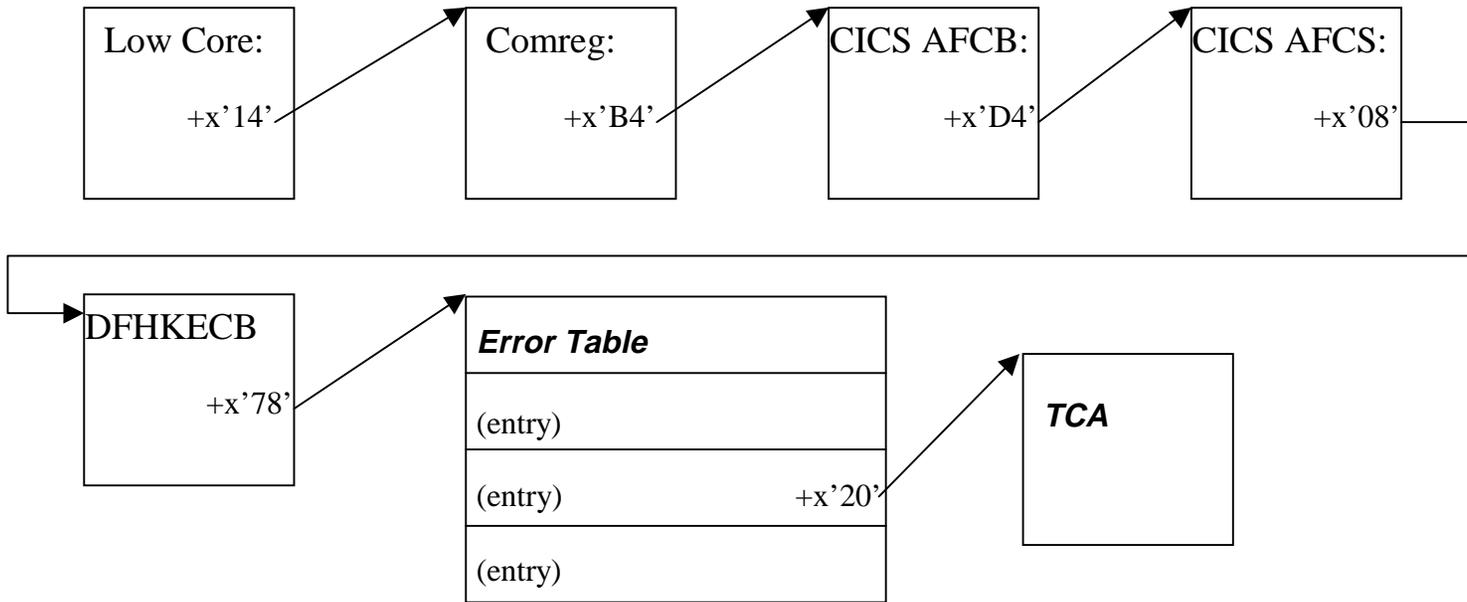
```

>>> FCT Get <<<
 13F14A0 F18F0400 0044144C 00BE009C 01C2A8C4 | 1.....ByD |
 13F14B0 40404040 40404040 50B6E1C8 54BB0F16 | ...H.... |
>>> Getmain File I/O Storage (VSWA)<<<
 13F14C0 C8000400 0044144C 00BFF4D0 8FBE00A8 | H.....4....y |
                                     |_____ VSWA Address
 13F14D0 40404040 40404040 50B55BEC 54BB0F18 | ..$. .... |
>>> Getmain Exit Trace <<<
 13F14E0 F18F0400 0044144C 00BF003A 01C2A8C4 | 1.....ByD |
 13F14F0 40404040 40404040 60B6E562 54BB0F1B | -.V.... |
>>> Getmain File Storage (3A bytes) <<<
 13F1500 C8000400 0044144C 00BE06E0 8FBF0048 | H..... |
 13F1510 40404040 40404040 50B55BEC 54BB0F1D | ..$. .... |
>>> Getmain Exit Trace <<<
 13F1520 F0C00400 0044144C 80000000 00F23840 | 0.....2. |
 13F1530 40404040 40404040 50B719DA 54BB0F58 | ..... |
>>> Task Control <<<
 13F1540 F1400400 0044144C 00BE06E0 01C2A8C4 | 1 .....ByD |
 13F1550 40404040 40404040 60B6E594 54BB0F93 | -.Vm...l |
>>> Freemain <<<
 13F1560 C9000400 0044144C 00BE06E0 8FBF0048 | I..... |
 13F1570 40404040 40404040 50B55C28 54BB0F96 | ..*....o |
>>> Freemain exit <<<
 13F1580 F581B500 0044144C 00080210 98100004 | 5a.....q... |
                                     |_____ VSAM Request (Get)
                                     |_____ VSAM RPL Options
                                     |_____ VSAM RPL Feedback (Record not found)
                                     |_____ VSAM Return
 13F1590 40404040 40404040 40B6D4A2 54BB0F99 | .Ms...r |
>>> VSAM Response <<<
 13F15A0 F5100300 0044144C 10BFF4D0 00000000 | 5.....4.... |
 13F15B0 40404040 40404040 80B6B706 54BB0F9B | ..... |
>>> File Control release FIOA (VSWA)
 13F15C0 F1400400 0044144C 00BFF4D0 01C2A8C4 | 1 .....4..ByD |
 13F15D0 40404040 40404040 40B6E594 54BB0FA5 | .Vm...v |
>>> Freemain <<<
 13F15E0 C9000400 0044144C 00BFF4D0 8FBE00A8 | I.....4....y |
 13F15F0 40404040 40404040 50B55C28 54BB0FA7 | ..*....x |
>>> Freemain Exit <<<
 13F1600 F5000500 0044144C 00000000 00000000 | 5..... |
 13F1610 40404040 40404040 40B6D4A2 54BB0FAA | .Ms.... |
 13F1620 E100F400 0044144C 81000000 00000602 | ..4....a..... |
                                     |_____ EIBFN (Read)
                                     |_____ EIBRCODE (Not Found)
 13F1630 40404040 40404040 8136F578 54BB0FAC | a.5..... |
>>> DFHEIP Exit <<<

```

See "CICS/VSE User's Handbook" (SX33-6079), chapter on "Trace" for a detailed description of the various trace table entry types.

# CICS TS Kernel Error Table



## CICS TS Kernel Error Table

00000000	000A0000	00000000	00000000	00000000	00007618	00003F20	070F1000	80003548	00	*	.....*
00000020	070D2000	80114EE6	070D0000	000E55C2	00000000	00000000	070F2000	80110DD4	00	*	.....+W.....B.....M*
00000040	00000000	04000000	000687F0	00000000	FF000000	00000000	040C0000	0000B71C	00	*	.....0.....*
00000060	040C0000	8000B2B4	000C0000	0000B964	04080000	00016772	040C0000	0000B786	00	*	.....*

Low core + x'14' → Current partition comreg

00003F20	F0F361F3	F061F9F9	61F1F900	00000000	00000000	00000080	C4C2C4C3	C3C9C3E2	00	*	*03/30/99/19.....DBDCCICS*
00003F40	00600FFF	00600245	00600245	000000B0	028FFFFF	FF5FCCD3	2000CCD0	19000000	00	*	*.-.-.-.-.-.....L.....*
00003F60	32B0B205	9EDC4612	15F81605	186C38F0	F3F3F0F9	F9F0F8F9	00001528	00000000	00	*	*.....8.....%.033099089...*
00003F80	50500000	F1F90000	000000B0	F24000B0	000073BB	00352660	0000096C	14580000	00	*	*&&..19.....2.....-.....%.*
00003FA0	00000000	05C811E1	0044C000	40400F40	40404040	40404000	40404040	40404000	00	*	*.....H.....*
00003FC0	0038EFC0	44000000	00346C20	020480C0	00000000	003C4200	00350C80	01000300	00	*	*.....%.....*
00003FE0	00040018	00000000	01FDD000	00000000	C9D1C2C6	C6F2091C	C4C6C8E2	C9D74040	00	*	*.....IJBFF2..DFHSIP*
00004000	003BF140	00000000	00000000	006000E0	00049C38	00600245	E2E8E240	40404040	00	*	*..1.....-.....-..SYS*
00004020	00000040	40000000	00000019	00000000	40404040	40404040	00000000	00000000	00	*	*.....*
00004040	00000000	003BA000	00000000	C6D6D9E2	C5C3C6F2	00000000	00000000	00000000	00	*	*.....FORSECF2.....*

Comreg + x'B4' → CICS Authorized Function Control Block (AFCB)

003C4200	C1C6C3E7	029600C0	80684D88	006EA7F0	00000000	00000000	00000000	00000000	00	*	*AFCX.....(>.0.....*
003C4220	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00	*	*.....*
003C4240	00000000	00000000	00000000	00000000	825D91F2	00000000	00000000	00000000	00	*	*.....).2.....*
003C4260	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00	*	*.....*
003C4280	00000000	00000000	811F5C20	01185000	81E3C320	00000000	00651020	00000000	00	*	*.....*...&..TC.....*
003C42A0	81EA4800	003C42E0	00000000	00000000	00000000	00000000	00000000	00000000	00	*	*.....*
003C42C0	00000000	00000000	811F52A0	00000000	00100000	003C00E0	01EF8020	00000000	00	*	*.....*
003C42E0	00686EC4	C6C8C4E2	C4E2C1E4	E3C24040	01EF6000	00000000	00000000	00000000	00	*	*..>DFHSDSAUTB ..-.....*
003C4300	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00	*	*.....*

AFCB + x'D4' → Authorized Function common Control block Storage (AFCS)

003C00E0	C1C6C3E2	01000088	00609000	0A9600B0	01EA4400	003C4000	80684D88	C4C2C4C3	00	*	*AFCS.....-.....(.DBDC*
003C0100	C3C9C3E2	C4C2C4C3	C3C9C3E2	C4C6C8C3	D3E34040	00000000	00000000	00000000	00	*	*CICSDBDCCICSDFHCLT.....*
003C0120	01EA4600	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00	*	*.....*
003C0140	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00	*	*.....*

AFCS + x'08' → DFHKECB (Kernel Anchor Table)

# CICS TS Kernel Error Table

AFCS + x'08' → DFHKECB (Kernel Anchor Table)

00609000	01E86EC4	C6C8D2C5	D2C3C240	40404040	81F1A080	81F1C398	81F1A480	81F2CD30	BO	*.Y>DFHKEKCB	.1..1C..1..*
00609020	81F1C150	81F1BCB8	81F1A880	81F1AC80	81FAB188	01F18D00	81F1A700	81F1AF00	BO	*.1A&.1...1...1...1...1...1...*	
00609040	00000000	00000000	00001388	01F185F8	00000005	04452000	01F11880	00000000	BO	*.....1.8.....1..*	
00609060	00000000	003A0000	012A3000	10AD0007	006091E8	00000000	01F11D00	01EF6080	BO	*.....-Y.....1...*	
00609080	00000000	7D000000	0000001A	00000023	00000006	0000A000	00000000	00000000	BO	*.....'.....*	
006090A0	00000003	00000001	8044C4E6	00000002	00000000	C4C2C4C3	C3C9C3E2	C4C2C4C3	BO	*.....DW.....DBDCCICSDBDC*	
006090C0	C3C9C3E2	C4C6C8C3	D3E34040	00000000	00000000	C3C9C3E2	C4C6C8E2	C9E34040	BO	*CICSDFHCLT...CICSDFHHSIT*	
006090E0	C5F6F4F0	006020B0	01F11C80	01F11C00	00000A96	00000000	00000000	40208000	BO	*E640.-...1...1.....*	

KECB + x'78' → DFHKERRD (CICS Kernel Error Table)

01F11D00	00286EC4	C6C8D2C5	C5D9C840	40404040	01F11D28	01F16FF8	01A80000	00000000	BO	*..>DFHKEERH	.1...1?8...*
01F11D20	00000008	00000007	F0C3F161	C1D2C5C1	010400C1	0000FFFF	C4C6C8C6	C3E5D940	BO	*..OC1/AKEA...A...DFHFCVR	*
01F11D40	01480A00	011F7080	00692000	012AA380	00000001	00000001	07BD0001	40000000	BO	*.....*	*
01F11D60	07BD0000	801D2332	00020001	00000000	801D2332	B0000000	801D408E	0064C8C0	BO	*.....H..*	*
01F11D80	00000000 <sup>2</sup>	01193758 <sup>3</sup>	0064C840 <sup>4</sup>	00000000 <sup>5</sup>	0064C880 <sup>6</sup>	0064E488 <sup>7</sup>	01192B1C <sup>8</sup>	0064C8E0 <sup>9</sup>	BO	*.....H...H...U.....H..*	*
01F11DA0	0064E380 <sup>A</sup>	0064C890 <sup>B</sup>	001D2318 <sup>C</sup>	01190D88 <sup>D</sup>	801D39A6 <sup>E</sup>	001D45A2 <sup>F</sup>	00000000	00000000	BO	*..T...H.....*	*
01F11DC0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	BO	*.....*	*
01F11DE0	00000000	00000000	00000000	00000000	00000000	00000000	07BD0001	40000000	BO	*.....*	*
01F11E00	07BD0000	801D2332	00020001	00000000	801D2332	B0000000	801D408E	0064C8C0	BO	*.....H..*	*
01F11E20	00000000	01193758	0064C840	00000000	0064C880	0064E488	01192B1C	0064C8E0	BO	*.....H...H...U...H..*	*
.											
.											
.											
01F12700	40404040	40404040	00000000	00000000	00000000	00000000	F0C3C261	C1D2C5C1	BO	*.....OCB/AKEA*	*
01F12720	010400CB	0000FFFF	C4C6C8C1	D7D3C9F1	013D9030	011F7080	00692000	012AA380	BO	*.....DFHAPLI1.....*	*
01F12740	00000007	0000000B	07BD000B	E0000000	07BD2000	81B02254	0006000B	00000000	BO	*.....*	*
01F12760	81B02254	B0000000	0165B60C <sup>0</sup>	0165B678 <sup>1</sup>	00000000 <sup>2</sup>	0165B840 <sup>3</sup>	01B01C88 <sup>4</sup>	0165B190 <sup>5</sup>	BO	*.....*	*
01F12780	00710B9C <sup>6</sup>	00000000 <sup>7</sup>	007C00D0 <sup>8</sup>	0165B710 <sup>9</sup>	01B01D78 <sup>A</sup>	01B02034 <sup>B</sup>	01B01D44 <sup>C</sup>	0165B4C8 <sup>D</sup>	BO	*.....@.....H..*	*
01F127A0	81B02188 <sup>E</sup>	00000000 <sup>F</sup>	00000000	00000000	00000000	00000000	00000000	00000000	BO	*.....*	*
01F127C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	BO	*.....*	*
01F127E0	00000000	00000000	07BD000B	E0000000	07BD2000	81B02254	0006000B	00000000	BO	*.....*	*

PSW

TCA

ILC, Interrupt Code

Error Table entry +x'20' → TCA (for older abend entries, this storage will probably have already been released)  
 +x'38' = Failing PSW  
 +x'50' = Registers (0-15)