
CICS Problem Support

Bob Sodan

sodanrob@us.ibm.com

APRIL 2008

AGENDA

- Introduction
 - CICS Level 2 Support Center
 - What are the different code areas, and who supports them?
 - Level 2 availability actions
 - New FTP address policy
 - CICS Problem Determination
 - Storage Violations
 - Hangs
 - Loops
 - Waits
 - Abends
 - The DFHPD410 Dump Formatter
 - CICS Level 2 recommended settings
-

The CICS Level 2 Support Center Hints & Tips

- Is it MVS or VSE?
 - "Code is Code"
 - CICS Host
 - FILE CNTRL/DATABASES(VSAM, IMS, DL/I, DB2), JOURNAL, TEMP STORAGE, MONITOR, STATS, SPOOLER, RECOVERY, DUMPS NOT PRODUCED, BACKOUT, RESTART, SHUTDOWN, TRANSIENT DATA, ANY AREA NOT COVERED ABOVE.
 - CICS TC
 - SECURITY, MRO, AUTOINSTALL, WEB, BMS, ISC, FEPI, TERMINAL INTERFACE(BTAM, VTAM, LU62), STORAGE PROBLEMS, RESOURCE DEF'N, SOCKETS, TCP/IP, REPORT CONTROLLER, TABLE MANAGER, XRF, SOAP, RECOVERY MANAGER.
-

CICS Level 2 Support Center Notes

- The code for CICS/TS for VSE/ESA has been ported from various CICS/TS releases. The code being used is dependent on which function is being utilized. However, the machine code in the storage does not know the difference between, VM, MVS or VSE as an operating system.
 - The CICS Level 2 Support Center is divided into two distinct areas: CICS Host and CICS TC. This separation attempts to align the most common code areas into the two support groups.
 - There is no longer a Level 1 and Level 2 support structure. The Customer Service Center fields a call and based on our customer selections, they will direct a call to the correct CICS Level 2 group. All calls now are handled on a callback action instead of the traditional "live call" action.
 - The CICS Level 2 Support Center also answers fee based "how to" questions on separate queues. When a call comes on the SupportLine queue an entitlement check is done. If our customer is not entitled to SupportLine help they will be told they are not supported.
-

FTP Information

THE TESTCASE SITE WILL NO LONGER ACCEPT DOCUMENTATION

PLEASE SEND YOUR DOCUMENTATION USING FTP.

- 1. CONNECT TO OUR FTP SITE: ftp.emea.ibm.com
(or 192.109.81.7) USER: ANONYMOUS PASSWORD: YOUR COMPLETE
E-MAIL ADDRESS.**
- 2. IF YOU ARE SENDING A RAW DUMP CHANGE TO BINARY MODE.**
- 3. IF THIS IS A FORMATTED DUMP/TRACE CHANGE TO ASCII MODE.**
- 4 Place the dataset in the /toibm/vse directory with put command**

**Ensure the data set name conforms to the following naming convention:
XXXXX.YYY.ZZZ.DDD.DDD where: XXXXX =PMR #; YYY =Branch #;
ZZZ =Country Code; DDD.DDD =Short Descriptive Name;**

(eg. PUT 'SYSDUMP.F4.DF500075' 87744.199.000.dump.untrs)

When the dump is transmitted to the ECUREP site, a secondary pmr will automatically be generated. This will let Level2 know the documentation is ready for analysis.

STORAGE VIOLATIONS

- Types of storage violations
 - CICS detected storage violations
 - SAAs and Check Zones
 - SCEs and SCFs
 - Storage violation debug
-

Types of Storage Violations

- CICS detected and reported
 - Reported by DFHSM0102 message
 - DFHSM0102 applid A storage violation (code X'code') has been detected by module modname
 - Undetected by CICS
 - Require different problem determination techniques
-

CICS Detected Storage Violations

- Initial or duplicate Storage Accounting Area (SAA) of a Terminal Input/Output Area (TIOA) storage element has become corrupted.
 - Leading or trailing Check Zone of a user-task storage element has become corrupted.
 - Detected at freemain time. Not when the violation actually happened.
 - SAA chains are checked when an individual element is requested to be freed, at least up to the target element.
 - SAA chains are checked during freemain of storage belonging to a TCTTE after the last output has taken place.
 - Check Zones are checked during freemain of a specific user-task storage element.
 - Check Zones chains are checked during freemain of all user-task storage during task termination.
-

SAAs

- **Are eight bytes long**
 - First eight bytes of the TIOA are known as the initial (leading) SAA
 - Last eight bytes of the TIOA are known as the duplicate (trailing) SAA
 - First word indicates Storage Class and length
 - First byte represents the Storage Class (x'85' = TIOA storage)
 - Last two bytes represent the length of the TIOA
 - Length includes initial SAA and useable portion of TIOA only
 - Second word is the chain pointer
 - Will point to another TIOA in the chain or to the owning TCTTE +4
 - **Comparison of the initial and duplicate SAA is done, at freemain time, to detect possible overlay**
-

SAA Storage for TIOAs

Storage Address	Offset		Storage			
12FBB310	0000	<u>85000118</u>	<u>12FBB000</u>	00000000	00000000	
12FBB320	0010	00000000	00000000	00000000	00000000	
12FBB330	0020	00000000	00000000	00000000	00000000	
12FBB340	0030	00000000	00000000	00000000	00000000	
12FBB350	0040	00000000	00000000	00000000	00000000	
12FBB360	0050	00000000	00000000	00000000	00000000	
12FBB370	0060	00000000	00000000	00000000	00000000	
		..				
		..				
12FBB3F0	00E0	00000000	00000000	00000000	00000000	
12FBB400	00F0	00000000	00000000	00000000	00000000	
12FBB410	0100	00000000	00000000	00000000	00000000	
12FBB420	0110	00000000	00000000	<u>85000118</u>	<u>12FBB000</u>	

Check Zones

- **Are eight bytes long**
 - First eight bytes are known as the leading check zone
 - Last eight bytes are known as the trailing check zone
 - Byte 0-3 indicates the task subpool name
 - C00 = CICS above the line storage (ECDSA)
 - M00 = CICS below the line storage (CDSA)
 - U00 = USER above the line storage (EUDSA)
 - B00 = USER below the line storage (UDSA)
 - **Bytes 4-7 indicates the owning task number**
 - **There are no chain pointers as Storage Manager is aware of all task storage and their lengths via Storage Element Descriptors (SCE) and Free Storage Descriptors (SCF).**
 - **Comparison of the leading and trailing check zone is done, at freemain time, to detect possible overlay**
-

Storage Violation Analysis

CICS24.00005 00046000 CICS storage below 16MB

```
0000 D4F0F0F0 F0F0F0F5 00B46EC4 C6C8C5C9 *M0000005..>DFHEI*
0010 E4E24040 40404040 00000000 00000000 *US .....*
0020 00000000 00000000 00000000 00000000 * .....*
0030 - 004F LINES SAME AS ABOVE
0050 000460D0 00000000 00000000 00000000 *..-}.....*
0060 00000000 00000000 00000000 00000000 * .....*
0070 - 009F LINES SAME AS ABOVE
00A0 00000000 00000000 00000000 00046050 * .....-&*
00B0 00046054 00000000 00000000 00000000 * ..-.....*
00C0 00656EC4 C6C8C1D7 6DC4C6C8 C5C9C25C *..>DFHAP_DFHEIB**
00D0 0000000C 0106080F C3E2E2E8 0000005C * .....CSSY...**
00E0 00000000 00000000 00000000 00000000 * .....*
00F0 00000000 00000000 00000000 00000000 * .....*
0100 00000040 40404040 40404000 00000000 * .....*
0110 00000000 00000000 00000000 00000000 * .....*
0120 - 045F LINES SAME AS ABOVE
0460 00000000 00000000 D4F0F0F0 F0F0F0F5 * .....M0000005*
```

SCEs and SCFs

SCE.M0000005 11FD0290 Storage Element Descriptor

```
0000 11FD3610 11FD3610 00046000 00000470 * .....-.....*  
0010 11F09040 00000000          *.0. .... *
```

SCF.M0000005 11FD0278 Free Storage Descriptor

```
0000 11FD3620 11FD3620 00046470 00000B90 * .....*  
0010 11F09040 00000000          *.0. .... *
```

Third word contains the location of the storage.
Fourth word contains the length of the storage.

Output created via VERBX DFHPD410 'SM'

Storage Violation Analysis ...

Message produced:

DFHSM0102 IYNXH A storage violation (code X'0F0C') has been detected by module DFHSMAR.

From the Messages and Codes manual:

A storage violation has been detected Explanation: by module modname.
The code X'code' is the exception trace point ID which uniquely identifies the type of storage violation.

System Action:

An exception entry (X'code' in the message) is made in the trace table.
Use the exception trace point ID, X'code', to investigate the cause of the storage violation. A description of the exception trace point ID, and the data it contains, is in the CICS Trace Entries manual.
A system dump is taken, unless you have specifically suppressed dumps in the dump table.
CICS continues unless you have specified in the dump table that CICS should terminate.

Storage manager domain trace points

Excerpt from CICS Trace Entries manual:

Point ID	Module Lvl	Type	Data
SM 0F0A	DFHSMAR	Exc Insufficient storage for SCQs	1 SMAR parameter list
SM 0F0B	DFHSMAR	Exc Insufficient storage for SMXs	1 SMAR parameter list
SM 0F0C	DFHSMAR	Exc Storage check failure	1 SMAR parameter list 2 Address of storage element 3 Length of storage element 4 First 512 bytes (max) of storage element 5 Last 512 bytes (max) of storage element 6 Data preceding storage element (1K max) 7 Data following storage element

DFHSM0102 Abbreviated Trace

Output via: VERBX DFHPD410 'TR=1'

```
00515 QR  AP 05A8 APRC  ENTRY PERFORM_COMMIT      NO,FORWARD,00000001      =003685=
00515 QR  AP 05A9 APRC  EXIT PERFORM_COMMIT/OK    NO                      =003686=
00515 QR  RM FA12 RMUO  EXIT COMMIT_UOW/OK        =003687=
00515 QR  KE 0201 KEDD  ENTRY INQUIRE_ANCHOR     0000002C                =003688=
00515 QR  KE 0202 KEDD  EXIT INQUIRE_ANCHOR/OK   12075000                =003689=
00515 QR  DP 0900 DPXM  ENTRY RELEASE_XM_CLIENT   =003690=
00515 QR  DP 0901 DPXM  EXIT RELEASE_XM_CLIENT/OK =003691=
00515 QR  AP 0590 APXM  ENTRY RELEASE_XM_CLIENT   NORMAL                  =003692=
XM  QR  AP 0591 APXM  EXIT RELEASE_XM_CLIENT/OK   =003693=
XM  QR  US 0401 USXM  ENTRY END_TRANSACTION        =003694=
XM  QR  XS 0401 XSXM  ENTRY END_TRANSACTION        =003695=
XM  QR  XS 0402 XSXM  EXIT END_TRANSACTION/OK     =003696=
XM  QR  US 0402 USXM  EXIT END_TRANSACTION/OK     =003697=
XM  QR  PG 0801 PGXM  ENTRY TERMINATE_TRANSACTION =003698=
XM  QR  PG 0802 PGXM  EXIT TERMINATE_TRANSACTION/OK =003699=
XM  QR  SM 0F01 SMAR  ENTRY RELEASE_TRANSACTION_STG =003700=
XM  QR  XM 1001 XMIQ  ENTRY SET_TRANSACTION        INCREMENT                =003701=
XM  QR  XM 1002 XMIQ  EXIT SET_TRANSACTION/OK      =003702=
XM  QR  AP 1700 TFIQ  ENTRY SET_TERMINAL_FACILITY YES =003703=
XM  QR  AP 1701 TFIQ  EXIT SET_TERMINAL_FACILITY/OK =003704=
XM  QR  SM 0F0C SMAR  *EXC* Storage_check_failed_at_address 001007D0
                               RELEASE_TRANSACTION_STG          =003705=
XM  QR  ME 0301 MEME  ENTRY SEND_MESSAGE 66,SM0102,11C32D16,00000002,00000008SM =003706=
XM  QR  KE 0101 KETI  ENTRY INQ_LOCAL_DATETIME_DECIMAL =003707=
XM  QR  KE 0102 KETI  EXIT INQ_LOCAL_DATETIME_DECIMAL/OK =003708=
```


Storage Manager View

SCA.B0000515 31471458 Subpool Control Area

```
0000 C2F0F0F0 F0F5F1F5 314713A4 11FD3B60 08010200 00000000 00000000 00000000 *B0000515.u..-.*
0020 00000000 00000000 00000000 00000000 00000004 00000000 00000002 00000000 *.....*
0040 00000000 00000000 00000000 00000000 11FD0BF0 11FD0C98 00100470 00000000 *....0..q.....*
0060 11F9B368 11F9B308 7FFFFFFF 7FFFFFFF 00000000 11DF4470 314714D0 314714D0 *.9..9."..}.}*
0080 00000000 00000000 FFFFFFFF 00100201 01020000 00000000 00001000 00000840 *.....0.. *
00A0 00000000 00001000 11FEB228 00000000 00000000 *..... *
```

SCE.B0000515 11FD0BF0 Storage Element Descriptor

```
0000 11FD0C98 314714A8 001007D0 000003D0 11FD1A88 00000000 *..q..y..}... *
```

SCE.B0000515 11FD0C98 Storage Element Descriptor

```
0000 314714A8 11FD0BF0 00100000 00000470 11FD1A88 00000000 *..y...0..h.... *
```

SCF.B0000515 11F9B368 Free Storage Descriptor

```
0000 11F9B308 314714B8 00100470 00000360 11FD1A88 00000000 *.9....-..h... *
```

SCF.B0000515 11F9B308 Free Storage Descriptor

```
0000 314714B8 11F9B368 00100BA0 00000460 11FD1A88 00000000 *...9....-..h.. *
```

Storage Browse

```
00100470 C2F0F0F0 F0F5F1F5 0008034E 00000000 | B0000515...+....|
00100480 00000000 00000000 00000000 0010005C | .....*|
00100490 00000000 500C10DC 00000000 001007C4 | ...&.....D|
001004A0 001004F0 12BF5838 000C1028 12BF4F70 | ...0.....|.|
001004B0 12446F0F 12447F0E 12448F0D 12449F0C | ..?..."......|
001004C0 00100B94 1244BF0A 001000D0 008B8000 | ...m.....}....|
001004D0 00000000 00000000 00000000 00100488 | .....h|
001004E0 008B8000 001000D0 00000000 00000000 | .....}.....|
001004F0 800C184C 001005F8 801007C4 00000000 | ...<...8...D....|
.
. 001007E0.:100B8F. LENGTH(X'03B0')--All bytes contain X'00' |
00100B90 00000000 00400020 000400C8 00C800C8 | .....H.H.H|
00100BA0 0C02C000 0800008C 00120800 00080000 | ..{.....|
00100BB0 20400E08 00000000 00000000 00000000 | .....|
00100BC0.:100CCF. LENGTH(X'0110')--All bytes contain X'00' |
00100CD0 00000000 00000000 C2F0F0F0 F0F0F4F3 | .....B0000043|
```

00100470 = Free storage x'360' long (SCF)

00100BA0 = Free storage x'460' long (SCF)

001007D0 = Allocated storage x'3D0' long (SCE)

DFHSM0102 Full Trace

AP 00E1 EIP ENTRY GETMAIN REQ(0004) FIELD-A(00100488 ...h) FIELD-B(08000C02)

TASK-00515 KE_NUM-0049 TCB-QR/008CCE88 RET-500C10A2 TIME-15:31:20 INTERVAL-0.00000104
=003633=

SM 0C01 SMMG ENTRY - FUNCTION(GETMAIN) GET_LENGTH(3BC) SUSPEND(YES)
STORAGE_CLASS(USER24) CALLER(EXEC)

TASK-00515 KE_NUM-0049 TCB-QR/008CCE88 RET-928B1AAC TIME-15:31:20 INTERVAL-0.00000212
=003634=

.
.

SM 0C02 SMMG EXIT - FUNCTION(GETMAIN) RESPONSE(OK) ADDRESS(001007D8)

TASK-00515 KE_NUM-0049 TCB-QR/008CCE88 RET-928B1AAC TIME-15:31:20 INTERVAL-0.00000104
=003636=

Trace can be used to discover what program issued the getmain for the storage. To find the getmain, search for the address Storage Manager knows +x8'. This will find the trace entry showing the exit of DFHSMMG. Back up in trace to find the EIP ENTRY GETMAIN trace entry. The RET value will point to the program that issued the getmain.

Loader Domain

PROGRAM STORAGE MAP

PGM NAME	ENTRY PT	CSECT	LOAD PT.	REL. PTF	LVL.	LAST COMPILED	COPY NO.	USERS	LOCN
IBMRSAP	800ABA20	-noheda-	000ABA20					1	1
READUPDT	000C1000	DFHYA410	000C1000	410				1	0
RPL									SDSA

Output via: VERBX DFHPD410 'LD=1'

Storage Browse of READUPDT Program

```
000C1000 C4C6C8E8 C1F6F4F0 58F0021C 58F0F0D0 | DFHYA410.0...00} |
000C1010 58F0F014 58F0F00C 58FF000C 07FF5CC6 | .00..00.....*F |
000C1020 C9D3D3C9 D55C0000 47F0F028 23D9C5C1 | ILLIN*...00..REA |
000C1030 C4E4D7C4 E34DE45D 40F0F461 F0F461F0 | DUPDT(U) 04/04/0 |
000C1040 F640F1F6 4BF2F940 C5E2C1F6 F4F04040 | 6 16.29 R410 |
000C1050 90ECD00C 183F4510 3036033E 000058F0 | ..}.....0 |
000C1060 37C805EF 50D01004 18F1BF1F D0184780 | .H..&}...1..}... |
000C1070 3050D207 F05C1000 18DF58B0 D05CD201 | .&K.0*.....}*K. |
```

Note: To calculate a proper offset into the module, you must account for the length of the Exec Interface stub. The entry point is offset x'28' because this is the branch instruction that will branch to the STM instruction (offset x'50') upon entry to the module.

FINDINGS

- Task 515 was terminating when storage violation was determined
 - AP 0F0C trace entry indicates:
 - Violated storage is located at address 001007D0
 - Violated storage is x'3D0' bytes long
 - Trailing Check Zone is overlaid with value of 000400C8 00C800C8
 - Violation was at least x'8' bytes long
 - Unable to determine if violation affected more storage as the following storage was never allocated
 - Program READUPDT getmained this storage
 - Program READUPDT most likely wrote past his allocated storage
 - When / Where / Why
-

Command Protection

- If you ask, CICS will overlay storage for you
 - EXEC CICS READ INTO(a CICS storage area)
 - Corrupted CICS control blocks
 - EXEC CICS READ SET(another transaction's area)
 - Corrupted data, program or control blocks
 - EXEC CICS RECEIVE INTO(0)
 - **Could crash CICS**
 - CMDPROT={YES | NO }
 - Checks first byte of referenced target fields
 - ABEND AEYD if incorrect area is referenced
-

Storage Violation Tools

➤ Messages

❑ DFHSM0102/DFHSM0103

DFHSM0102 applid A storage violation (code x'code') has been detected in module name

'CODE' is the exception trace point that identifies the type of violation

DFHSR0622

DFHSR0622 applid An attempt to {overwrite|access} the dsaname has caused the abend which follows

DFHSR0618

DFHSR0618 applid An illegal macro call or reference to the CSA or TCA has caused the abend which follows

DFHAP0001/DFHSR0001 produced for program checks; may be an indication of storage overlays

Storage Violation Tools

- **SM0102/SM0103 dumps**
 - **Trace**
 - SM 0Cxx, SM 0Dxx, SM 0Exx, SM 0Fxx
 - SM 09xx - storage checker
 - Use a BIG trace
 - **CSFE DEBUG**
 - CHKSTSK=CURRENT|ALL
 - Validates SCZs for task storage
 - ALL option is gone in CICS TS 1.3
 - CHKSTRM=ALL
 - Validates SAAs
 - Produces SM0103 dump
 - **DFHTRAP**
-

Storage Violation Trap (CSFE)

- **Built in storage violation trap**
 - **Runs each time an old style AP trace entry is written (has old style Field A and Field B)**
 - Ensure level 1 trace is turned on for all components and level 1-2 for the EI component
 - Ensure Monitoring package is not suppressing EI trace entries
 - **Checks all storage areas on the transaction storage chain for the currently running task**
 - **Produces a DFHSM0103 dump and turns itself off when a violation is detected**
 - Will catch the violation within a window much closer to when the violation actually occurs
 - If window is still too large, consider adding additional user trace points to the application using EXEC CICS ENTER TRACENUM commands
 - **Turned ON/OFF via:**
 - SIT parameter CHKSTSK
 - CHKSTSK=CURRENT
 - CHKSTSK=NONE
 - **Manually as a transaction**
 - CSFE DEBUG,CHKSTSK=CURRENT
 - CSFE DEBUG,CHKSTSK=NONE
-

DFHSM0103 Debugging

- Message produced:
 - DFHSM0103 IYNXH A storage violation (code X'0932') has been detected by the storage violation trap. Trap is now inactive.
 - From the Messages and Codes manual:
 - Explanation: A storage violation has been detected by the storage violation trap, which may be enabled via the CHKSTSK or the CHKSTRM system initialization parameters or via the CSFE transaction. The code X'code' is the exception trace point ID which uniquely identifies the type of storage violation detected.
 - System Action: CICS disables the storage violation trap. An exception entry (X'code' in the message) is made in the trace table. A system dump is taken, unless you have specifically suppressed dumps in the dump table.
 - CICS continues unless you have specified in the dump table that CICS should terminate.
-

Storage Violation Summary

- **Use the "Preventative" facilities**
 - Reentrant Program Protection
 - WRITE protects ALL programs linked as RENT
 - Independent from other 'preventative' facilities
 - NO performance impact
 - **Storage Protection**
 - Minimum level of protection between CICS key storage and USER key storage only
 - No inter-transaction protection
 - No performance impact
 - **Command Protection**
 - Stops the overlay before it happens
 - Run in test
 - . 5% overhead
 - **Fix problem applications!**
 - **Suppressing the SM0102 dumps doesn't make the problem disappear!**
-

Storage manager domain trace points

Point ID	Module	Lvl	Type	Data
SM 0932	DFHSMCK		Exc	Storage zone check failed 1 SMCK parameter list 2 Subpool name 3 Address of storage 4 First 128 bytes of storage element 5 Last 16 bytes of storage element

DFHSM0103 Full Trace

AP 1940 APLI ENTRY - **FUNCTION(START_PROGRAM) PROGRAM(READUPDT) CEDF_STATUS(CEDF)**
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-9229C972 TIME-17:19:29 INTERVAL-0.0000046
=004694=

SM 0C01 SMMG ENTRY - **FUNCTION(GETMAIN) GET_LENGTH(34E) SUSPEND(YES) INITIAL_IMAGE(00)**
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80085E44 TIME-17:19:29 INTERVAL-0.0000065
=004695=

SM 0C02 SMMG EXIT - **FUNCTION(GETMAIN) RESPONSE(OK) ADDRESS(00100478)**
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80085E44 TIME-17:19:29 INTERVAL-0.0000016
=004696=

AP 00E1 EIP ENTRY GETMAIN REQ(0004) FIELD-A(00100488 ...h) FIELD-B(08000C02)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-500C10A2 TIME-17:19:29 INTERVAL-0.0000020
=004697=

SM 0901 SMCK ENTRY - **FUNCTION(CHECK_STORAGE) TASK_STORAGE(CURRENT_TASK)**
TP_STORAGE(NO)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80083BB6 TIME-17:19:29 INTERVAL-0.0000014
=004698=

SM 0902 SMCK EXIT - **FUNCTION(CHECK_STORAGE) RESPONSE(OK)**
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80083BB6 TIME-17:19:29 INTERVAL-0.0000013
=004699=

DFHSM0103 Full Trace (cont...)

AP 1701 TFIQ EXIT - FUNCTION(SET_TERMINAL_FACILITY) RESPONSE(OK)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-91CECDE6 TIME-17:19:29 INTERVAL-0.0000019
=004712=

SM 0932 SMCK *EXC* -Zone_check_failed-FUNCTION(CHECK_STORAGE)
TASK_STORAGE(CURRENT_TASK) TP_STOR

TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80083BB6 TIME-17:19:29 INTERVAL-0.00000034
=004713=

```
1-00 00200000 00000010 00000000 00000000 BC000000 00000000 01000100 0201C621 *.....F*
2-00 C2F0F0F0 F0F1F0F8                               *B0000108      *
3-00 001007D0                                       *...}      *
4-00 C2F0F0F0 F0F1F0F8 00000000 00000000 00000000 00000000 00000000 00000000 *B0000108.....*
   20 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
   40 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
   60 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
5-00 00000000 00400020 000400C8 00C800C8                               *..H.H.H  *
```

NOTE: The SM 0932 trace point does not offer the length of the violated storage.
Storage Manager must be used to determine the length.

Storage Manager

SCA.B0000108 314529F8 Subpool Control Area

```
00 C2F0F0F0 F0F1F0F8 31452944 31452AAC 08010200 00000000 00000000 00000000 *B0000108.....*
20 00000000 00000000 00000000 00000000 00000000 00000003 00000000 00000000 00000000 *.....*
40 00000000 00000000 00000000 00000000 11FD1D58 11FD1A70 00000000 00000000 *.....*
60 11FD1B90 11FD1B90 7FFFFFFF 7FFFFFFF 00000000 11DF4470 31452A70 31452A70 *....."..."*
80 00000000 00000000 FFFFFFFF 00100201 01020000 00000000 00001000 00000BA0 *.....0.....*
A0 00000000 00001000 11FF225C 00000000 00000000 *.....* *
SCE.B0000108 11FD1D58 Storage Element Descriptor
```

```
0000 11FD1518 31452A48 001007D0 000003D0 11FD2160 00000000 *.....}...}...- *
```

SCE.B0000108 11FD1518 Storage Element Descriptor

```
0000 11FD1A70 11FD1D58 00100470 00000360 11FD2160 00000000 *.....-...- *
```

SCE.B0000108 11FD1A70 Storage Element Descriptor

```
0000 31452A48 11FD1518 00100000 00000470 11FD2160 00000000 *.....- *
```

SCF.B0000108 11FD1B90 Free Storage Descriptor

```
0000 31452A58 31452A58 00100BA0 00000460 11FD2160 00000000 *.....-...- *
```

Output via: VERBX DFHPD410 'SM'

Storage Browse

```
001007D0 C2F0F0F0 F0F1F0F8 00000000 00000000 | B0000108..... |
001007E0.:100B8F. LENGTH(X'03B0')--All bytes contain X'00'
00100B90 00000000 00400020 000400C8 00C800C8 | .....H.H.H |
00100BA0 0C02C000 0800008C 00120800 00080000 | ..{..... |
00100BB0 20400E08 00000000 00000000 00000000 | . ..... |
00100BC0.:100FFF. LENGTH(X'0440')--All bytes contain X'00'
00101000 C2F0F0F0 F0F0F5F8 00B46EC4 C6C8C5C9 | B0000058..>DFHEI |
```

001007D0 addresses the violated storage (trailing check zone is overlaid)

00100BA0 addresses an SCF

Violation Caught !!!!

AP 00E1 EIP EXIT GETMAIN OK REQ(00F4) FIELD-A(00000000) FIELD-B(00000C02)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-500C10A2 TIME-17:19:29 INTERVAL-0.00000067
=004704=

SM 0901 SMCK ENTRY - FUNCTION(CHECK_STORAGE) TASK_STORAGE(CURRENT_TASK)
TP_STORAGE(NO)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80083BB6 TIME-17:19:29 INTERVAL-0.00000035
=004705=

SM 0902 SMCK EXIT - FUNCTION(CHECK_STORAGE) RESPONSE(OK)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80083BB6 TIME-17:19:29 INTERVAL-0.00000075
=004706=

AP 00E1 EIP ENTRY SUSPEND REQ(0004) FIELD-A(00100488 ...h) FIELD-B(08001208)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-500C10C6 TIME-17:19:29 INTERVAL-0.00000073
=004707=

SM 0901 SMCK ENTRY - FUNCTION(CHECK_STORAGE) TASK_STORAGE(CURRENT_TASK)
TP_STORAGE(NO)
TASK-00108 KE_NUM-0048 TCB-QR/008CCE88 RET-80083BB6 TIME-17:19:29 INTERVAL-0.00000035
=004708=

The window, of the violation being detected, is between the exit of the getmain request and the entry to the suspend request. The code running between this window is responsible for the violation. Make note of the two RETURN addresses for calculation of offset into responsible program.

Loader Domain

PROGRAM STORAGE MAP

PGM NAME	ENTRY PT	CSECT	LOAD PT.	REL.	PTF LVL.	LAST COMPILED	COPY NO.	USERS	LOCN	TYP
IBMRSAP	800ABA20	-noheda-	000ABA20					1	1	RDSA RPL
READUPDT	000C1000	DFHYA410	000C1000	410				1	0	SDSA RPL
DFHSIP	11C554B8	DFHCICS	11C00000	0410	HCI4100	I 02/03 08.21		1	0	ERGN
ANY										

Output via: VERBX DFHPD410 'LD=1

AP Domain within DFHSM0103 Dump

USER24.00108 001007D0 USER storage below 16MB

```
000 C2F0F0F0 F0F1F0F8 00000000 00000000 00000000 00000000 00000000 00000000 *B0000108.....*
020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
040 - 03BF LINES SAME AS ABOVE
3C0 00000000 00400020 000400C8 00C800C8 *.....H.H.H *
```

** DFHPD0125 Storage violation detected at 001007D0. Trailing SAA is invalid.

USER24.00108 00100470 USER storage below 16MB

```
000 C2F0F0F0 F0F1F0F8 0008034E 00000000 00000000 00000000 00000000 0010005C *B0000108...+.....*
020 00000000 500C10C6 00000000 001007C4 001004F0 12BEF838 000C1028 12BEEF70 *...&..F...D..0.8.*
```

Note: The AP domain is useful within a DFHSM0103 dump because the violation was caught during normal execution of the application and not at task termination time.

Output via: VERBX DFHPD410 'AP=3'

WARNING THIS FORMAT WILL TAKE SOME TIME TO COMPLETE AND THE OUTPUT IS LARGE

Findings / Solution

➤ Findings

- Program READUPDT was responsible for the violation
 - Violation was caught between an EXEC CICS Getmain command issued at offset x'78' and an EXEC CICS SUSPEND command issued at offset x'9C'
 - Inspection of program READUPDT:
 - Shows register 9 originally pointed to the getmained storage but was incremented by x'3BC' just prior to a MVC instruction to move x'20' bytes. This move instruction was the cause of the violation.
-

Findings / Solution (contd ...)

➤ Possible Solutions

- If you truly want to move data into this getmained storage:
 - Adjust the increment of register 9 to a value that falls within the getmained storage, allowing for x'20' bytes to be moved.
 - If the move was not meant to be:
 - Remove the MVC instruction or
 - Possibly remove both the LA and MVC instructions
-

CICS Problem Determination

ABENDS

SCENARIO:

Running normally and all of a sudden the CICS partition crashes.

MESSAGES:

DFHSR0615 Program interrupt has occurred in recovery task

DFHDU0201 ABOUT TO TAKE SDUMP. DUMPCODE: SR0615 ,
DUMPID: 1/0002

0S24I AN SDUMP OR SDUMPX MACRO WAS ISSUED 28/04-15:43:47

0S29I DUMP STARTED 28/04-15:43:47

0S30I DUMP STARTED. MEMBER=DF600034.DUMP IN SUBLIB=SYSDU 28/04-
15:43:47

CICS Problem Determination

ABENDS Example 1

DFHSR0615 applid Program interrupt has occurred in recovery task

Explanation: An operating system abnormal termination occurred. CICS started to abend the task with an abend code of ASRB when a program check occurred. DFHSRP terminates CICS.

System Action: CICS abnormally terminates with system dump SR0615 and exception trace entries giving the kernel error data for the operating system abend and the program check.

User Response: The most likely cause of the program check is an error in a global user exit program running at the XSRAB exit. This is the global user exit that can be invoked when an abend code is found in the SRT. If such a program was running, determine the cause of the program check and correct it.

For advice on problem determination, refer to the CICS Problem Determination Guide.

CICS Problem Determination

ABENDS Example 1

We issue: CALL DFHPD410 DATA KE=3 in the Infoana job to see what the running task is doing:

```
00A2 046CB080 ***Running** 007E6680 00055 TV44 09A13380 09B51020
*YES*
```

We then display this task's stack entry to see what the module flow is:

```
00A2 046CC020 0120 Bot 89B8ABE8 89B8AEB8 02D0 DFHKETA
00A2 046CC140 01F0 Dom 89BAF6F8 89BAF7E6 00EE DFHDSKE
00A2 046CC330 03F0 Dom 89C2A898 89C2B396 0AFE DFHXMTA
00A2 046CC720 03E0 Dom 841400C8 8414218C 20C4 DFHPGPG
      Int +00D4 84140156 008E INITIAL_LINK
      Int +1A06 841404A6 03DE ACQUIRE_FOR_LINK
00A2 046CCB00 0510 Dom 841AF480 841AFB9E 071E DFHAPLI1
00A2 046CD010 0450 Sub 841B4418 840612F6 0000 *YES* DFHAPLI3
00A2 046CD460 03A0 Sub 84061500 84061E14 0914 *YES* DFHSRP
00A2 046CD800 0F80 Dom 89BDEDD8 89BE23A2 35CA DFHMEME
```

CICS Problem Determination

ABENDS Example 1

We then examine KE Domain Error Summary to see the error flow:

==KE: KE Domain Error Table Summary

ERR_NUM OFFSET	ERR_TIME	KE_NUM	ERROR TYPE	ERR_CODE	MODULE
=====	=====	=====	=====	=====	=====
00000003 00000020	15:41:28	0000	TRAN_ABEND_PERCOLATE	---/APSI	DFHLIRET
00000004 00003022	15:43:47	00A2	ABEND	2C5/AKEB	DFHYC220
00000005 0000071A	15:43:47	00A2	PROGRAM_CHECK	0C1/AKEA	-noheda-

CICS Problem Determination

ABENDS Example 1

We see the initial abend2C5 happened so we find this KERRD information:

Error Code: 2C5/AKEB Error Type: ABEND Timestamp: C083E66DA3200980
Reason Code: 47020033

KE_NUM: 00A2 KE_TASK: 046CB080 TCA_ADDR: 007E6680
DS_TASK: 09A13380

Error happened in program DFHYC220 at offset 00003022

PSW: 07DD0000 0086E372 Instruction Length: 2 Interrupt Code: 25
Exception Address: 00000000

Execution key at Program Check/Abend: D

REGISTERS 0-15

0086E66E	0086EA40	0086F740	00826A10
0086B380	009C02AC	0086F740	00000D28
0086BEF0	6086E336	0086B380	0086B380
0086C1A8	0086E9A8	6086DBB2	0086E330

CICS Problem Determination

ABENDS Example 1

We now print storage go to the PSW address area x'0086E372' and go backwards to find the beginning of the module:

The abend happened in:

```
*0.j.....f.y.f...fAy.f.0*  
*.b...f..COBF3000V00P44 .fA.15/0*  
*1/9710.06.09.....0000.....*  
* .....*
```

We then go to the PSW address area in storage and look for something which would give us an indication of the cause of the abend:

```
* .....&.o..0..ILBDDBG03010/05/90* 0086E330
```

CICS Problem Determination

ABENDS Example 1

Analysis:

DFHSR0615 ABEND2C5 occurs after migrating to CICS TS for VSE/ESA 1.1.1

CICS Kernel error data (KERRD) offset x'2C' contains the reason code x'47020033'. The description of this reason code in the VSE Message and Codes manual indicates that a program has issued a STXIT AB macro although an ESTAEX exit is already defined.

The Program Status Word (PSW) at time of abend points into COBOL module **ILBDDBG0**.

CICS Problem Determination

ABENDS Example 1

Cause

A COBOL program has been compiled with a **restricted COBOL option or verb**. This causes ILBDDBG0 to attempt to set up its own abend recovery routines using the STXIT AB. However, VSE rejects the STXIT AB request because CICS has already set up its abend recovery routine using an ESTAE.

Solution

Change the COBOL application to remove the restricted verbs or options. To determine which COBOL programs contain the CICS restricted verbs and options, you can use LIBR LIST and look for eyecatcher ILBDDBG0. The presence of ILBDDBG0 indicates that the COBOL application was compiled with one of the restricted options.

The CICS Application Programming Guide contains a list of the restricted verbs and options under the topic COBOL Considerations .

CICS Problem Determination

ABENDS Example 2

Scenario:

Our customer upgraded to TCP/IP 15E Beta version for VSE. When they shutdown TCP/IP they get the following errors:

Messages:

```
DFHSO0106 06/28/06 09:45:21 A Language Environment Callable  
      Service error (code X'0226') has occurred in module DFH SOCK.  
SYMPTOMS: PIDS/564805400 LVLS/411 MS/DFHSO0106 RIDS/DFH SOCK  
      PTFS/UK08369 PRCS/00000226
```

```
DFHSO0002 A severe error (code X'0211') has occurred In module DFH SOCK.
```

CICS Problem Determination

ABENDS Example 2

We issue **CALL DFHPD410 DATA KE=3** in the Infoana job to find the running task:

KE_#	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB
0027	0531DB00	*Running	04D8B080	00024	CSOL	06680480	06790020

We then examine the **task's stack output** to see what the module flow shows:

KE_#	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFS	ERROR NAME
0027	053D5020	0120	Bot	867C9BE8	867C9EB8		02D0	DFHKETA
0027	053D5140	01F0	Dom	867EE6F8	867EE7E6		00EE	DFHDSKE
0027	053D5330	03F0	Dom	86869898	8686A396		0AFE	DFHXMTA
0027	053D5720	03E0	Dom	84E400D8	84E40A6C		0994	DFHPGPG
			Int	+00D4	84E40166		008E	INITIAL_LINK
0027	053D5B00	0510	Dom	84FEA3E0	851D7148		0000	DFHAPLI1
			Int	+166C	84FEA986		05A6	CICS_INTERFACE
0027	053D6010	0AF0	Dom	84E57AB0	84E5DABC		600C	DFH SOCK
0027	053D6B00	0F80	Dom	8681DDD8	868214C8		36F0	DFHMEME
			Int	+2CD6	8681DF4E		0176	SEND

CICS Problem Determination

ABENDS Example 2

We then issue **CALL DFHPD410 DATA TR=3** in the Infoana job and find the *EXC exception trace entries:

```
TR 0101 TRPT *EXC* - OVERLENGTH-ENTRY : CALLING DOMAIN WAS .. :  
PARAMETER LIST WAS FUNCTION(TRACE_PUT) POINT_ID(226) DATA1(035DF580  
    , 00000040) DATA2(02590000 , 00000398) DATA3(00000000 , 00000248)  
DATA4(00000000 , 02590308)
```

```
TASK-00054 KE_NUM-0007 TCB-00364000 RET-82017AC8 TIME-  
11:54:28.5079103310 INTERVAL-00.0000673281 =000113=  
1-0000 0021 * .. *
```

ME FF02 SUWT EVENT - BEFORE-VSE-WTO

```
TASK-00054 KE_NUM-0007 TCB-00364000 RET-8365FD4C TIME-  
11:54:28.5083155029 INTERVAL-00.0000474296 =000124=
```

```
1-0000 00268050 C4C6C8E2 D6F0F1F0 F640D7D9 D6C4C3C9 C3E24040  
DFHSO001
```

CICS Problem Determination

ABENDS ---- NOTES

CICS always performs exception tracing when it detects an exception condition (for example, bad parameters on a domain call, or an abnormal response from a called routine). The aim is "first failure data capture", to record data that might be relevant to the exception as soon as possible after it has been detected.

CICS uses a similar mechanism for both exception tracing and "normal" tracing. Exception trace entries are made from specific points in CICS code, and data is taken from areas that might provide information about the cause of the exception. The first data field in the trace entry is usually the parameter list from the last domain call, because this can indicate the reason for the exception.

Exception trace entries are always written to the internal trace table, even if no trace destinations are currently STARTED. That is why there is always an internal trace table in every CICS partition, to make sure there is always somewhere to write exception trace entries. If the other trace destinations are STARTED, the exception trace entries are written there, as well.

CICS Problem Determination

ABENDS ---- NOTES

You can select tracing options so that exception traces only are made to an auxiliary trace data set. This is likely to be useful for production regions, because it enables you to preserve exception traces in auxiliary storage without incurring any general tracing overhead. You need to disable all standard and special task tracing, and enable auxiliary trace:

1. Ensure that special tracing has not been specified for any task.
2. Set the master system trace flag off.
3. Set the auxiliary trace status to STARTED, and the auxiliary trace data set and the auxiliary switch status to whatever values you want.

Exception traces are now made to an auxiliary trace data set, but there is no other tracing overhead.

The format of an exception trace entry is almost identical to that of a normal trace entry. However, you can identify it by the eye-catcher *EXC* in the header.

CICS Problem Determination

ABENDS Example 2

SO 0211 **SOCK *EXC*** - ASYNCIO_ERROR RETURN_VALUE(1)
RETURN_CODE(1124) REASON_CODE(000000FF)

TASK-00054 KE_NUM-0007 TCB-00364000 RET-82017AC8 TIME-
11:54:42.9993327062 INTERVAL-00.0016391423 =000233=
1-0000 FFFFFFFF 00000464 000000FF 00000000 00000000 02590000
00000080 02590308 **
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 **
2-0000 03986EC4 C6C8E2D6 D3E3C540 40404040 01C22084 01C22084
043D0000 00008000 *.q>DFHSOLTE .B.d.B.d.....*
0020 00000000 00000000 001C6EC4 C6C8E2D6 E2E3C5C8 C5C1C440
02590028 02590028 *>DFHSOSTEHEAD*
0040 00000000 00000000 00000001 00008000 00000005 00000000 00000000
78000000 **

CICS Problem Determination

ABENDS Example 2

ANALYSIS:

We can see the errors are happening during SHUTDOWN processing. If the CICS Internal Trace is not started, this failure does not happen when closing the TCPIP SERVICE connection initially. Once the connection is closed and restarted, the error does not happen.

The trace domain shows us two errors:

1. An error response is returned on a `sock_call_async_accept_service`. The return codes are: `AioRv=-1` and `AioRc=ENOTCONN (1124)`. This error will give us the DFHSO0002 abend.
2. This in turn results in call `listener_error(tid_sock_accept_failure)`. `Listener_error` issues `trace_bpx_call_failure(trace_point_id)` this calls `trace_bpx` which selects which data areas to trace depending on the trace id. In this case (`tid_sock_accept_failure`) we trace data1,2,3,4. However, we have no data3 or data4 (STE and SOCKADDR_ADDR respectively) so both addresses and the lengths are unpredictable. In this case the length of `SOCADDR_ADDR` is overlength.

CICS Problem Determination

ABENDS Example 2

SOLUTION:

This failure resulted in creating two APARS for each error we found:

APAR Identifier PK29184 Last Changed 06/12/08

TR 0101 TRPT *EXC* - OVERLENGTH-ENTRY : CALLING DOMAIN WAS .. :
CLOSING A TCPIP SERVICE CONNECTION

The second APAR is:

APAR Identifier PK39175 Last Changed 07/03/09

SHUTTING DOWN TCP/IP LEADS TO A CICS DUMP DFHSO0106 X'0226' IN
DFH SOCK FOLLOWED BY DFHSO0002 WHEN INTERNAL TRACE IS ACTIVE.

CICS Problem Determination

ABENDS --- Summary

- Suggested steps to debug an ABEND
 - Ensure the DUMP=YES SIT parameter is set (this to get a dump of the abend)
 - Get all the CICS and VSE message logs
 - Issue “CALL DFHPD410 DATA KE=3,TR=3,LD=3” in the Infoana job
 - Call the IBM Support Center if you have any questions/problems
-

CICS Problem Determination

HANGS

SCENARIO

CICS shutdown hangs following the DFHTM1782 message that says that all non-system tasks have quiesced. The DFHSTP shutdown task is waiting on AP_QUIES CSASSI2 (which means it is waiting on terminal control quiesce.)

A CEMT SET VTAM FORCE has caused CICS to close the VTAM ACB, but the TCP task has not finished quiescing because the CSNE Task is still active.

The CSNE task has been stuck in an ICDELAY wait for several minutes because of an EXEC CICS DELAY INTERVAL(000001) done by a NEP. The CSNE task should have only suspended for 1 second with that command, but because of a problem, the CSNE task is now hung indefinitely.

CICS Problem Determination

HANGS

MESSAGES

DFHTM1782I applid All non-system tasks have been successfully terminated.

Explanation: This message is issued during shutdown of the CICS session after successful termination by the user of any active tasks which had previously prevented termination.

System Action: CICS shutdown continues normally.

User Response: None

Destination: Console

Modules: DFHSTP

XMEOUT Parameter: applid

CICS Problem Determination

Hangs

ANALYSIS:

The problem which causes the CSNE task to hang in the ICWAIT longer than it should is; the fact that DFHSTP breaks the ICE chain (near label STTSBY) after non-system tasks have quiesced and before waiting on the terminal control quiesce.

A break in the ICE chain while a task is waiting in an ICDELAY will cause that task to wait forever.

A clue that this problem has happened is the fact that the Time Of Suspend of the ICDELAY task (the CSNE task) is in the same second as time of the DFHTM1782 message.

CICS Problem Determination

TASK Hang --- SOLUTION

APAR Identifier PK42130 Last Changed 07/09/14

CICS SHUTDOWN HANGS AFTER DFHTM1782 MESSAGE. DFHSTP SHUTDOWN IS WAITING ON TERMINAL CONTROL QUIESCE (CSASSI2) .

Symptom IN INCORROUT Status CLOSED DOC
Severity 3 Date Closed 07/09/14
Component 564805400 Duplicate of
Reported Release B0P Fixed Release
Component Name CICSTS FOR VSE Special Notice
Current Target Date ..07/10/30 Flags
SCP
Platform

LOCAL FIX:

Don't do an EXEC CICS DELAY from a NEP. (Doing an EXEC CICS DELAY from a NEP isn't a good idea anyway.)

CICS Problem Determination

TASK Hang -- SOLUTION

The following **documentation change** will be made to the CICS Transaction Server for VSE/ESA Customization Guide (SC33-1652-00). Chapter 9 (Writing a node error program), the section titled 'Restrictions on the use of EXEC CICS commands' will be amended by the addition of the following:

- o Terminal control. For example, issuing EXEC CICS DELAY may cause the CSNE task to suspend and never resume, which can cause shutdown of the region to hang. ("CEMT-type" commands, such as EXEC CICS INQUIRE TERMINAL are permissible).
-

CICS Problem Determination

HANGS

SCENARIO

Because of a HANG Condition, a CICS partition had to be CANCELLED. This because when a shutdown had been started, it would not complete. CICS had been started, and even though TCPIP=NO was specified in the SIT, the CWBG transaction was in this hang state.

MESSAGES

MESSAGE: DFHXM0306 H3CICSA A severe error (code X'130C') has occurred while initializing task number 02395 with transaction identifier CWBG. The task is suspended indefinitely.

SYMPTOMS: PIDS/564805400 LVLS/411 MS/DFHXM0306 RIDS/DFHXMTA
PTFS/UQ58418

PRCS/0000130C

CICS Problem Determination

HANGS

MESSAGE: DFHXM0306 applid A severe error (code X'code') has occurred while initializing task number tasknum with transaction identifier tranid. The task is suspended indefinitely.

Explanation: An internal error has prevented the initialization of task number tasknum with identifier tranid.

The task cannot run and cannot be abended. The task has no principal facility bound to it. Rather than terminate CICS, the transaction manager keeps CICS running and preserves its integrity by suspending the task. The suspended task will hold its MXT slot until CICS is terminated.

Note that the task may hold resources (for example, locks and enqueues) so you should cancel CICS at your earliest convenience. Otherwise you may risk other tasks being prevented from running because they also need access to the same resources. You may wish to add a dump table entry to always terminate CICS on this message.

CICS Problem Determination

HANGS

MESSAGE: DFHXM0306 CONTD...

System Action: The task is suspended indefinitely. First failure diagnostics should be produced by the component which first detects the error. The transaction manager also takes a dump. Message DFHME0116 is normally produced containing the symptom string for this problem.

The task is suspended with a resource type of FOREVER and a resource name of DFHXMTA.

User Response: You must cancel CICS if you need to destroy the task. You cannot quiesce CICS since this task will not terminate. You cannot purge or force purge the task.

Note the error code X'code'. You need further assistance from IBM to resolve this problem. See Part 4 of the CICS Problem Determination Guide for guidance on how to proceed.

Destination: Console

CICS Problem Determination

Hangs

ANALYSIS:

In this scenario, the KERNEL and TRACE output did not show much information of much use for our customers. However, the stack and trace information was quite valuable to CICS support.

The problem is due a CICS module, DFHXMTA, called DFHUSXM for INIT_TRANSACTION_USER call and passing an invalid SESSION_USER_TOKEN. CWBG is a non terminal task and should never have a SESSION_USER_TOKEN.

DFHXMTA's stack storage was not being initialized to zeros. Thus causing CICS to make the bad choice of passing a SESSION_USER_TOKEN based on residual data within the area of the stack address.

CICS Problem Determination

Hangs

We first go into the KE=3 Infoana output to see what the running task shows:

```
0064 02D73780 ***Running** 020AA080 02395 CWBG 01DF8780 053B4020
```

We then look at the STACK entry for KE_NUM 0064 to see what that flow shows:

KE_NUM	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFS	ERROR	NAME
0064	02D7A020	0120	Bot	853E2BE8	853E2EB8	02D0		DFHKETA	
0064	02D7A140	01F0	Dom	85407530	8540761E	00EE		DFHDSKE	
0064	02D7A330	03F0	Dom	854826A8	854854FC	2E54		DFHXMTA	
	Int	+2240		85482BBA	0512			PROCESS_ATTACH_ERROR	
	Int	+2A16		85484C12	256A			ATTACH_WAIT_FOREVER	
	Int	+2DF2		854852AA	2C02			WAIT_FOREVER	
0064	02D7A720	0F80	Dom	85436BE8	8543A2D8	36F0		DFHMEME	
	Int	+2CD6		85436D5E	0176			SEND	
	Int	+14A2		854399AC	2DC4			CONTINUE_SEND	
	Int	+3618		85438108	1520			TAKE_A_DUMP_FOR_CALLER	
0064	02D7B6A0	0480	Dom	8540DA88	8540E6A8	0C20		DFHDUDU	
	Int	+0A3C		8540DB70	00E8			SYSTEM_DUMP	
	Int	+178E		8540E95C	0ED4			TAKE_SYSTEM_DUMP	

CICS Problem Determination

Hangs

We now go into the TR=2 Infoana output to see if there is good trace information:

```
US 0402 USXM EXIT - FUNCTION(INIT_TRANSACTION_USER)
RESPONSE(EXCEPTION) REASON(INVALID_USER_TOKEN) USDOM_TRANSACTION_TOKEN(00000000
, 00000000) PRIORITY(0)
```

```
TASK-02395 KE_NUM-0064 TCB-003E6000 RET-85482BA4 TIME-23:36:53.2703683759
INTERVAL-00.0000011718 =002282=
```

```
XM 130C XMTA *EXC* - Logic_error - FUNCTION(TASK_REPLY) USER_TOKEN(02AA6500)
TASK_TOKEN(040E0027) SUSPEND_TOKEN(040E0027)
```

```
TASK-02395 KE_NUM-0064 TCB-003E6000 RET-8540761E TIME-23:36:53.2703691416
INTERVAL-00.0000007656 =002283=
```

```
1-0000 00580000 00000003 00000001 00000000 A6000400 00000000 02F001F0 F0F04040
* .....w.....0.000
0020 40404040 02AA6500 040E0027 00000048 40404040 40404040 40404040 40404040 * .....
*
0040 40404040 40404040 040E0027 40404040 40404040 40404040 * .... *
```

CICS Problem Determination

TASK Hang -- SOLUTION

APAR Identifier PK37353 Last Changed 07/05/01
TRANSACTION CWBG WAIT FOREVER STATE CAUSES SHUTDOWN TO
HANG

Symptom MS MSGDFHXM0306

PTF List:

Release B0P : UK24146 available 07/05/01 (1000)

CICS Problem Determination

CICS Hangs

- Steps to debug a HANG in a CICS partition
 - Inspect the MSG log for signs of abends
 - If there are no abends dump the CICS partition
 - Run Infoana against the CICS partition dump
 - Issue “CALL DFHPD410 DATA
KE=3,TR=3,LD=3”
 - If you have questions/problems, call the IBM Support Center
-

CICS Problem Determination

VSE SUB-TASK Hangs

- Steps to debug a HANG
 - Issue the STATUS command (more than once if you suspect a Loop condition)
 - Issue MSG xx,dump 0-7FFFFFFF,uuu or
 - Create a Standalone dump or
 - Dump the Partition(s) in question, SVA and Supervisor
 - If you have questions/problems, call the IBM Support Center
-

CICS Problem Determination LOOPS

A loop is the repeated execution of some code. If you have not planned the loop, or if you have designed it into your application but for some reason it fails to terminate, you get a set of symptoms that vary depending on what the code is doing.

In some cases, a loop may at first be diagnosed as a wait or a performance problem, because the looping task competes for system resources with other tasks that are not involved in the loop.

CICS Problem Determination LOOPS

- The following are some characteristic symptoms of loops:
 - The 'system busy' symbol is permanently displayed in the operator information area of a display unit, or stays displayed for long periods.
 - The transaction abends with abend code AICA.
 - CPU usage is very high, perhaps approaching 100%, yet some tasks stay suspended or ready, but not running, for a long time. CPU usage is very high, perhaps approaching 100%, yet some tasks stay suspended or ready, but not running, for a long time. You can check what the CPU usage is for any VSE job by using the DISPLAY SYSTEM ACTIVITY screen of the VSE/ESA Interactive Interface. See the VSE/ESA Operation manual for more information.
 - There is reduced activity at terminals, or possibly no activity at all.
 - One or more CICS partitions appear to be stalled, or to be continuing only slowly.
-

CICS Problem Determination

LOOPS

➤ SCENARIO

- Our customer at 13:28 03/24, received msg DFHSM0133 CICS is under stress (short on storage above 16MB).
 - This was accompanied by multiple messages from CA-Explore/CICS reporting an Short on Storage (SOS) condition.
 - At the time the problem was occurring, a VSE GETVIS command showed 3.5M 24-Bit and 11M 32-bit GETVIS available in the partition.
 - From that point on, the CICS/TS System was unresponsive. Our customer was unable to sign on at a tn3270 terminal. An active CICS/TS session would not respond.
 - CICS/TS would respond to a MSG xx command from the console, and would accept commands entered from the console, but it does not appear that any of the commands that were entered were ever executed.
 - The CICS/TS partition had to be cancelled and restarted to recover.
-

CICS Problem Determination LOOPS

The first symptom was the **SOS** condition above the line. The **SM (Storage Manager)** output showed:

==SM: ECDSA Summary

Size:	31744K	
Cushion size:	128K	
Current free space:	80K (0%)	
* Lwm free space:	0K (0%)	
* Hwm free space:	1024K (3%)	
Largest free area:	36K	
* Times nostg returned:	0	
* Times request suspended:	67	
Current suspended:	0	
* Hwm suspended:	2	
* Times cushion released:	1570	
Currently SOS:	YES	<<<<<<<<<< SOS CONDITION
* Times went SOS:	3	
* Time at SOS:	00:00:00.067	
* Storage violations:	0	
CICS DUMP: SYSTEM=CICSPROD		
Access:	CICS	
* Extents added:	26	
* Extents released:	0	
Number of extents:	21	

CICS Problem Determination

LOOPS

Further on down in the SM (Storage Manager) domain output is the subpool summary. This presents information on different subpools to see who might be using more Pagestg than would be used for normal operation:

==SM: Domain subpool summary (ECDSA)

Name	Id	Initf	Bndry	Fxlen	Q-c	Gets	Frees	Elms	Elemstg	Pagestg
JCDYNLOG	C1			16		13142	6413	6729	13135056	13468K

The unequal number of Gets/Frees ratio and Pagestg amount shows us we are writing a lot to this JCDYNLOG and not freeing up the storage.

The Infoana command to get this output is:

CALL DFHPD410 DATA SM

CICS Problem Determination

LOOPS

In the TRace output there appears to be a looping condition as I see this pattern over and over:

```
31734 1 AP 00F9 JCP  ENTRY GETB 31734 1 AP F101 SMSCP ENTRY
DFHSC_FREEMAIN      DFHSC TYPE=FREEMAIN
31734 1 SM 0D01 SMMF  ENTRY FREEMAIN          007D7008,00000000,MACR
31734 1 SM 0D02 SMMF  EXIT  FREEMAIN/OK      CICS24 storage at 007D
31734 1 AP F102 SMSCP EXIT  DFHSC_FREEMAIN    DFHSC TYPE=FREEMAIN
31734 1 AP F101 SMSCP ENTRY DFHSC_GETMAIN    DFHSC TYPE=GETMAIN
31734 1 SM 0C01 SMMG  ENTRY GETMAIN          1D8,YES,CICS24_SAA,MAC
31734 1 SM 0C02 SMMG  EXIT  GETMAIN/OK      007D7008
31734 1 AP F102 SMSCP EXIT  DFHSC_GETMAIN    DFHSC TYPE=GETMAIN
31734 1 AP 00F9 JCP  EXIT  NORMAL
31734 1 AP EA00 TMP  ENTRY LOCATE             FCT,AFN1004
31734 1 AP EA01 TMP  EXIT  LOCATE            FCT,AFN1004,07A6C300,N
31734 1 AP 04E0 FCFR  ENTRY READ_SET_UPDATE  00000000,07A6C300,007D
31734 1 SM 0301 SMGF  ENTRY GETMAIN          7D0,YES,FCSS SET,CICS
31734 1 SM 0302 SMGF  EXIT  GETMAIN/OK      07D471B8
31734 1 AP F00A XCPC  ENTRY CONDITIONAL_ENQ
```

CICS Problem Determination

LOOPS

31734 1 AP 0460 FCJL ENTRY JOURNAL_REQUEST

07A6C300,007CEA3C,07D471B8,07D471B8

31734 1 AP 00F9 JCP ENTRY WRITE

31734 1 AP 00F9 JCP EXIT NORMAL

31734 1 AP 0461 FCJL EXIT JOURNAL_REQUEST/OK

31734 1 AP 04E1 FCFR EXIT READ_SET_UPDATE/OK

7D0,07D471B8,197,00000000,

31734 1 AP 04E0 FCFR ENTRY REWRITE_DELETE

00000000,07A6C300,YES,NO

31734 1 AP 0460 FCJL ENTRY JOURNAL_REQUEST

07A6C300,007CEA3C,SYNCHRONISE_READ_

31734 1 AP 00F9 JCP ENTRY WAIT

~~31734 1 DS 0004 DSSR ENTRY RESUME 008E0005~~

CICS Problem Determination

LOOPS

31734 1 DS 0005 DSSR EXIT RESUME/OK
31734 1 DS 0004 DSSR ENTRY WAIT_OLDC
J0191335.....ql.,JCIOBLOK,0068E5DC,
J01 1 DS 0005 DSSR EXIT SUSPEND/OK
J01 1 DS 0004 DSSR ENTRY WAIT_OLDW
JCTICA,JCIOCOMP,0068E639,NO,IO
J01 1 DS 0005 DSSR EXIT WAIT_OLDW/OK
J01 1 DS 0004 DSSR ENTRY SUSPEND 008E0005,DFHJ01A
.W^.....,JCJOURDS,
31734 1 DS 0005 DSSR EXIT WAIT_OLDC/OK
31734 1 AP 00F9 JCP EXIT NORMAL
31734 1 AP 0461 FCJL EXIT JOURNAL_REQUEST/OK
31734 1 AP EA00 TMP ENTRY LOCATE AFCT,AFN1004
31734 1 AP EA01 TMP EXIT LOCATE
AFCT,AFN1004,07A70060,NORMAL
31734 1 DS 0004 DSSR ENTRY WAIT_OLDW

CICS Problem Determination LOOPS

```
AFN1004,FCIOWAIT,00C54841,NO,IO
31734 1 DS 0005 DSSR EXIT WAIT_OLDW/OK
31734 1 SM 0301 SMGF ENTRY FREEMAIN          07D471B8,7D0,FCSS
SET,CICS
31734 1 SM 0302 SMGF EXIT FREEMAIN/OK
31734 1 AP 04E1 FCFR EXIT REWRITE_DELETE/OK  00000000
```

The Infoana command to see the Trace output is:

CALL DFHPD410 DATA TR

CICS Problem Determination

LOOPS

Here is the KERNAL output showing the running TASK-31734 and its stack entry and error information:

005B 0799A080 ***Running** 0073E680 31734 F1LS 0991E080 09A51020

KE_NUM @STACK LEN TYPE ADDRESS LINK REG OFFS ERROR NAME

005B 0799B020 0120 Bot 89A8ABE8 89A8AEB8 02D0 DFHKETA

005B 0799B140 01F0 Dom 89AAF6F8 89AAF7E6 00EE DFHDSKE

005B 0799B330 03F0 Dom 89B2A898 89B2B58E 0CF6 DFHXMTA

005B 0799B720 0280 Dom 90BFE220 90BFE81E 05FE DFHAPRM

Int +0500 90BFE2D4 00B4

ABNORMAL_TRANSACTION_END

005B 007140A0 0478 Lifo 07704000 87704790 0790 DFHDBP1\$

005B 0799B9A0 0240 Sub 90DD7F18 90DD8B62 0C4A DFHF CFR

Int +0BEA 90DD8326 040E ACCMTEST

005B 0799BBE0 03B0 Sub 90DF3BE8 90DF78C2 3CDA DFHF CVS

Int +3C7A 90DF400E 0426 FCJRNSRU

005B 0799BF90 0480 Sub 90DF28B0 90DF361C 0D6C DFHF CJL

Int +07EA 90DF296A 00BA JOURNAL_REQUEST

Int +0DAE 90DF31C8 0918

SYNCHRONISE_READ_UPDATE

Int +0D4E 90DF367C 0DCC

SYNCHRONISE_JOURNAL

005B 007D1020 0338 Lifo 00293E10 80295B94 1D84 DFHJCP

CICS Problem Determination LOOPS

==KE: KE Domain Error Table Summary

ERR_NUM OFFSET	ERR_TIME	KE_NUM	ERROR TYPE	ERR_CODE	MODULE
=====	=====	=====	=====	=====	=====
=====					
00000009 000004EC	13:49:57	005B	TRAN_ABEND_PERCOLATE	---/AFCY	DFHPCP
0000000A 00000F70	13:49:57	005B	TRAN_ABEND_PERCOLATE	---/AFCY	DFHFCEI
0000000B 0000038C	13:49:57	005B	TRAN_ABEND_PERCOLATE	---/AFCY	DFHEIP
0000000C 00000148	13:50:04	005B	TRAN_ABEND_PERCOLATE	---/AFCY	DFHEPC
0000000D 00000A48	13:51:58	0001	ABEND	122/AKEB	DFHKETCB

CICS Problem Determination LOOPS

It looks like the task was purged and this resulted in the AFCY abend:

AFCY

Explanation: The transaction issued a file request resulting in a call to the main file control program (DFHFCFR). During the processing of the request the transaction was purged (that is, was the subject of an explicit PURGE or FORCEPURGE request, was timed out, or was selected by CICS for termination in an attempt to alleviate an SOS condition). A purged response was returned from DFHFCFR to its caller.

System Action: The task is abnormally terminated with a CICS transaction dump.

CICS Problem Determination

LOOPS

We then looked at the dynamic log to see what its contents show:

```
* ..>DFHJCDBL_BUFF.J... '.....' *
* .....@.....a.....C.AFN100*
*4 .....P1 SHD .....P1 SHD*
* 40500025254 ...0 ....P1 SHD .*
* .....P1 SHD 40500025254 ...0 .*
*8L0...I01 *
* 0001SHD4070 2523 *
TACTICAL CO*
```

The Infoana command to get this output is:

CALL DFHPD410 DATA AP

LOCATE ON THE STRING DBL

CICS Problem Determination

LOOPS -- SOLUTION

The dynamic log shows that the task was in a loop issuing repeated commands against the same file control record in file AFN1004.

Each dynamic log record is the same which is the tell-tale symptom of a looping task that changes the same RIDFLD over and over.

It would seem that the RIDFLD is the data beginning 'P1 SHD...'. The FCT shows that the key length is x'22'.

Eventually, this task was purged and had to backout the changes so DFHDBP1\$ validly had to read back along the dynamic log issuing repeated READ UPDATE / REWRITE DELETE commands against this same record key.

This eventually lead to the SOS condition/CICS hang that was reported. To resolve this problem the Customer needs to determine the cause of the original application loop.

CICS Problem Determination LOOPS

- Steps to debug LOOPS
 - Issue the STATUS command several times (this to see if the PSW is constantly changing)
 - Issue "CEMT P SNAP" to get a dump of the partition
 - Dump the SVA and SUPERVISOR
 - Using the DFHPD410 dump formatter, format KE=3, TR=3, AP=3
 - Gather all the CICS and message logs
 - Contact the IBM Support Center for assistance
-

CICS Problem Determination

WAITS

Table 20. Resources that a suspended task might be waiting on

Resource type	Resource name	Suspending module	DSSR call	Task
(none)	DMWTQUEU	DFHDMWQ	SUSPEND	System
(none)	LMQUEUE	DFHMLM	SUSPEND	User
AP_QUIES	CSASSI2	DFHSTP	WAIT_OLDC	System
AP_TERM	STP_DONE	DFHAPDM	WAIT_	System
			EXTERNAL	only
EKCWAIT	Value of NAME argument on EXEC CICS WAIT EVENT command	DFHEKC	WAIT_OLDW	User
FCIOWAIT	file ID	DFHFCBD or DFHFCVR	WAIT_OLDW	User
FCPSWAIT	file ID	DFHFCVR	WAIT_OLDC	User
FOREVER	DFHXMTA	DFHXMTA	WAIT_	User
			EXTERNAL	
TCP_NORM	DFHZDSP	DFHZDSP	WAIT_OLDW	System
ZCIOWAIT	DFHZARQ1	DFHZARQ	SUSPEND	User

CICS Problem Determination WAITS

DMWTQUEU

A system wait typically when domains are shutting down.

LMQUEUE

It means that the suspended task cannot acquire the lock on a resource it has requested, probably because another task has not released it.

CSASSI2

It means the AP domain is trying to shutdown and it has not finished. Typically it waits on the Terminals to finish their shutdown process.

STP_DONE

This wait is done to wait for DFHSTP to complete its processing before returning to Domain Manager, as DM will assume we have completed QUIESCE when we return and set the phase point etc, allowing other domains to complete their QUIESCE processing.

EKCWAIT

EKCWAIT indicates that a task has issued an EXEC CICS WAIT EVENT command. USERWAIT indicates that a task has issued an EXEC CICS WAITCICS or EXEC CICS WAIT EXTERNAL command. If the wait is prolonged, you should identify the event being waited on.

CICS Problem Determination

WAITS

FCIOWAIT

A wait on resource type FCIOWAIT occurs when the exclusive control conflict is deferred internally by VSAM and not returned as an error condition to CICS. An example of this is when a request against an LSR file is made for exclusive control of a control interval (for example, by EXEC CICS WRITE or READ UPDATE) and either this task or another task already holds shared control of this control interval (for example, by STARTBR).

FCPSWAIT

If your task is waiting on either of resource types FCPSWAIT or FCSRSUSP, it means that it cannot get a VSAM string. FCPSWAIT shows that the wait is for a private string, and FCSRSUSP shows that the wait is for a shared resource string.

FOREVER

If you have found that a user task is waiting on a resource type of FOREVER, and resource name DFHXMTA, transaction manager has detected a severe error during task initialization or task termination. Transaction manager has suspended the task.

ZCIOWAIT

Suspends on resource type ZCIOWAIT occur when the task is waiting for some terminal I/O. Once the expected I/O event occurs, the task is resumed.

DFHZARQ1 - resource type and name for all application requested waits involving NON-LU 6.2 devices.

DFHZARR1 - waiting for a receive issued to a LU6.2 ISC connection.

DFHZARL1 - waiting for a send issued to a LU6.2 device.

CICS Problem Determination WAITS

SCENARIO

Our customer found CICS was in a “hang”, customer has to cancel CICS partition and dump was generated at 11:09:08. Last trace entry time is 10:50:59 which means CICS was “hung” after this time.

CICS Problem Determination

WAITS

Here is the CICS KERNAL entries showing the running task and what it is doing at the time of the "hang":

10 078EEB00 ***Running** 07897680 58065 FU18 1785AD80 1832D020

KE_NUM	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFS	ERROR	NAME
0010	078F8020	0120	Bot	98366BE8	98366EB8	02D0			DFHKETA
0010	078F8140	01F0	Dom	9838B6F8	9838B7E6	00EE			DFHDSKE
0010	078F8330	03F0	Dom	98406898	984073A4	0B0C			DFHXMTA
0010	078F8720	03E0	Dom	879400C8	87940A5C	0994			DFHPGPG
			Int	+00D4 87940156	008E				INITIAL_LINK
0010	078F8B00	0510	Dom	87AEA410	89982D3E	0000			DFHAPLI1
			Int	+223E 87AEA9AE	059E				LE370_INTERFACE
			Int	+2022 87AECC5C	284C				

INVOKE_FOR_RECURSION

10 078F9010 06A0 Lifo 006EB358 879CD894 0000 DFHERM

CICS Problem Determination

WAITS

We now know that there was a call to an external resource manager to do some work. We need to see what the last trace entry shows as this will give us the request information:

```
AP 00E7 ERM ENTRY APPLICATION-EXEC-DLI          REQ(0004) FIELD-  
A(C22D71F4 B..4) FIELD-B(D5075904 N...) RESOURCE(DLI  
          TASK-58065 KE_NUM-0010 TCB-0041F000 RET-89982D3E  
TIME-10:50:59.0021395947 INTERVAL-00.0000005312  =116992=
```

CICS Problem Determination

WAITS

We then go to the RETurn address listed in the previous trace entry to see who is making the call to the external resource manager.

This call is made at offset x'1BFC' in the application:

```
| .....00..CEE..... | *  
| .....00.ϕ....q.. | *  
| ..... | *  
| ..0.q.0<.....q. | *  
| .q...q...q...q. | *  
| .q...q...q..... | *  
| ....CFUOM1802008 | *  
| 0327161010010100 | *  
| .....-...h... | *  
| ..... | *  
*****
```

CICS Problem Determination

WAITS -- SOLUTION

- We know CICS was last dispatched at the call to the External Resource Manager DL/I.
 - We know that control was never returned from this DL/I call.
 - There was CPU activity going on in the DL/I partition.
 - Our customer needed to find out what is going on with the DL/I activity and see why it never returned to CICS.
-

CICS Problem Determination WAITS

- Steps to DEBUG WAITS
 - DUMP the CICS Partition --- CEMT P SNAP
 - Run the "Analyze CICS Dumps" format the following domains: KE=3, TR=3, DS=1
 - Contact the IBM CICS Support Center
-

CICS dump formatter

```
* $$ JOB JNM=DMPACD1,DISP=D,PRI=8,  
* $$ NTFY=YES,  
* $$ CLASS=0  
* $$ LST DISP=H,RBS=1000  
// JOB DMPACD1 ANALYZE CICS/TS DUMP  
// EXEC PROC=DTRINFOA  
// EXEC INFOANA,SIZE=INFOANA,OS390  
    SELECT DUMP MANAGEMENT  
    DUMP NAME SYSDUMP.BG.DBG00002  
    RETURN  
  
    SELECT DUMP VIEWING  
    CALL DFHPD410 DATA AP=0,KE=3,DS=1,TR=2,LD=3  
    RETURN  
        DUMP NAME SYSDUMP.BG.DBG00002  
    RETURN  
    SELECT END  
  
/*  
/&  
* $$ EOJ
```

```
C /===/  
C *====*  
  *====*  
  *====*  
  *====*  
  *====*  
  *====*  
    *====*  
    *====*  
    *====*  
      *====*  
      *====*  
      *====*  
        *====*  
        *====*  
        *====*  
          *====*  
          *====*  
          *====*  
            *====*  
            *====*  
            *====*
```