



IBM Software Group

CICS Debugging Basics

Sarah Bertram and Charlie Wiese
CICS Level 2 Support



WebSphere® Support Technical Exchange



Agenda

- IPCS Basics
 - ▶ What's needed
 - ▶ Commands
 - ▶ CICS verbexit

- Types of Dumps
 - ▶ SLIP dumps
 - ▶ Console dumps
 - ▶ CICS dumps

- CICS verbexit samples

- CICS References

IPCS Basics



Basics: What's Needed

- TSO ID
 - ▶ Logon "Size" parameter should be 16MB or larger
- Define and Initialize IPCS Dump directory
 - ▶ Create a VSAM dataset for directory
 - Allocate as FILE(IPCSDDIR)
 - Refer to z/OS® IPCS User's Guide
- CICS verbexit: DFHPDxxx
 - ▶ Must reside in a dataset which has been added to the LINKLST or an IPCS STEPLIB or TASKLIB
 - ▶ CICS-specific IPCS configuration description can be found in the CICS Operations and Utilities Guide

Version	Verbexit
CICS/TS 2.2	DFHPD620
CICS/TS 2.3	DFHPD630
CICS/TS 3.1	DFHPD640
CICS/TS 3.2	DFHPD650

Basics: IPCS Primary Menu

```

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

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

                                *****
                                * USERID  - JOEUSER
                                * DATE     - 07/09/06
                                * JULIAN   - 07.249
                                * TIME    - 16:20
                                * PREFIX   - JOEUSER
                                * TERMINAL- 3278
                                * PF KEYS - 24
                                *****

```

Enter END command to terminate IPCS dialog

```

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

```

Basics: Dump Selection

- On the IPCS Primary Options menu
 - ▶ Select option '0'
 - ▶ Enter dump dataset name (format shown below)

```
----- IPCS Default Values -----
Command ==>
```

You may change any of the defaults listed below. The defaults shown before any changes are LOCAL. Change scope to GLOBAL to display global defaults.

Scope ==> **BOTH** (LOCAL, GLOBAL, or BOTH)

If you change the Source default, IPCS will display the current default Address Space for the new source and will ignore any data entered in the Address Space field.

Source ==> **DSNAME('YOUR.DUMP.DATASET.NAME')**

← Enter your dump dataset name

Address Space ==>

Message Routing ==> PRINT TERMINAL

Message Control ==> CONFIRM VERIFY FLAG(WARNING)

Display Content ==> **NOMACHINE** REMARK REQUEST NOSTORAGE SYMBOL

Press ENTER to update defaults.

Use the END command to exit without an update.

Basics: MACHINE vs. NOMACHINE

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

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

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

← Note key of '88'

- Specifying display content of **NOMACHINE**
 - ▶ Displays just the storage area
 - ▶ Here is an example:

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

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

Basics: Common IPCS commands

- **CBFormat** – format a control block
 - ▶ RTCT
 - ▶ STR(TODCLOCK)
- **EQuate** - create user-defined symbol
- **Find** - search for a value
- **List**
 - ▶ Sliptrap
 - ▶ storage address
 - ▶ Title
- **RETP** - retrieve previous commands
- **SELECT** – display ASID information
 - ▶ All
 - ▶ Current
 - ▶ ASID
- **STATUS**
 - ▶ REGS
 - ▶ SYS
- **VERBX** – Verbexit
 - ▶ DFHPDxxx (where 'xxx' is the CICS release)
 - ▶ MTRACE

Basics: ST (System Status) command

===>ST SYS provides information about the Time of dump, and the Dump Requester

▶ In a SLIP dump

SYSTEM STATUS:

Nucleus member name: IEANUC01

I/O configuration data:

IODF data set name: IODFST.IODF26

IODF configuration ID: B710

EDT ID: A1

Sysplex name: UTCPLEXA

TIME OF DAY CLOCK: C015FFD5 DE97C417 01/31/2007 05:51:44.839036 local

TIME OF DAY CLOCK: C01642E4 01D7C417 01/31/2007 10:51:44.839036 GMT

Program Producing Dump: SLIPDUMP

Program Requesting Dump: IEAVTSDT

▶ In a Console dump

SYSTEM STATUS:

Nucleus member name: IEANUC01

I/O configuration data:

IODF data set name: SYS1.IODF03

IODF configuration ID: PLX1

EDT ID: P1

Sysplex name: SYSPLEX1

TIME OF DAY CLOCK: C1289889 1C88A7F9 09/06/2007 15:33:51.602826 local

TIME OF DAY CLOCK: C1288B1F E248A7F9 09/06/2007 14:33:51.602826 GMT

Program Producing Dump: SVCDUMP

Program Requesting Dump: IEAVTSDT

Incident token: SYSPLEX1 MV23 09/06/2007 14:34:10.614280 GMT

The MVS image issuing the dump request is MV23, a member of sysplex SYSPLEX1

Basics: ST (System Status) *continued*

===>ST SYS

▶ In a CICS Initiated SDUMP

SYSTEM STATUS:

Nucleus member name: IEANUC01

I/O configuration data:

IODF data set name: SYS1.IODF03

IODF configuration ID: PLX1

EDT ID: P1

Sysplex name: SYSPLEX1

TIME OF DAY CLOCK: C1288161 516B0239 09/06/2007 13:50:15.860912 local

TIME OF DAY CLOCK: C12873F8 172B0239 09/06/2007 12:50:15.860912 GMT

Program Producing Dump: SVCDUMP

Program Requesting Dump: DFHKETCB

Incident token: SYSPLEX1 MV23 09/06/2007 12:50:34.446615 GMT

Basics: Which ASIDs were dumped?

- To identify which address spaces are contained in the dump you need to format the Recovery Termination Control Table (RTCT), as in the following example:

====> CBF RTCT

```
RTCT: 00F51DB0
+0000 NAME..... RTCT      SAP..... 2FD0BE00  SUP..... 00100000  SYD..... FF800000  SDLA..... 0000
+0018 FASB..... 00000000  NAS..... 00000002  EEDA..... 02875AA8  SDDS..... 00000000  SDDC..... 0000
+002C DSV..... 00C43D98  SSTK..... 00000000  ADGL..... 00E03470  ADG1..... 00E03510  ADG2..... 00E03520
+0044 ADG4..... 00E035A0  ADG5..... 80E038AC  TABG..... 80D80100  TABQ..... 80D8011E  TABR..... 80D80160
+005C DIND..... 026C0360  DIRS..... 026C0730  SDAT..... 026B3D80  SMOD..... 026C01C0  SCON..... 029B3028
+0074 RPAR..... 01761FD8  BPXP..... 00000000  TABO..... 00CF2008  SDPL..... 026B3CC8  FMT..... 00000000
+00A8 MSRB..... 00F64460  TEST..... 00000000  SEQ#..... 1346      SDSW..... 0268F000  TDCB..... 00000000
+00C4      00000000  00000000  00000000  00000000  00000000  00000000  SDWK..... 00C43E68
+00E2 ECPU..... 0000      EASD..... 0000      ETIM..... 00000000  SAO..... 2FD0BE00  SUO..... 00100000
+00F8 SDO..... 5FA20000  SDNA..... 02      INDX..... 01      SDPR..... 00      BUFV..... 00000000
+010A ZZZ3..... 4000
```

	<u>SDAS</u>	SDF4	SDF5
	----	----	----
001	<u>0396</u>	A0	00
002	<u>000F</u>	80	00
003	0000	00	00
004	0000	00	00

The address spaces contained in this dump are: 0396 and 000F

Basics: What Jobname corresponds to those ASIDs?

- The SELECT ALL command lists, in address space order, all jobs in the system at the time the dump was taken.
- It is important to note the dump does not contain all jobs listed.
- Using the IPCS SELECT command

- ▶ SELECT ALL - ASID to Jobname Cross Reference

ASID	JOBNAME	ASCBADDR	SELECTION CRITERIA
----	-----	-----	-----
0001	*MASTER*	00FD4600	ALL
0002	PCAUTH	00F4E080	ALL
0003	RASP	00FB0980	ALL
0004	TRACE	00FB0800	ALL
0005	DUMPSRV	00FB0680	ALL
...			
<u>000F</u>	<u>OMVS</u>	<u>00FB0B80</u>	<u>ALL</u>
...			
0395	BPXAS	00F0D900	ALL
<u>0396</u>	<u>CICSCWAA</u>	<u>00FBBF80</u>	<u>ALL</u>
0398	U0230254	00F77280	ALL

Basics: SELECT *continued*

- The SELECT CURRENT command lists the job which was executing at the point in time the dump was initiated. If the dump was issued via console dump command, the SELECT CURRENT command will display the Master scheduler address space.

SELECT CURRENT

```
ASID JOBNAME  ASCBADDR SELECTION CRITERIA
-----
0396 CICSCWAA 00FBBF80  CURRENT
```

- The SELECT ASID() command lists the specific ASID's requested

SELECT ASID(x'000f',x'0396')

```
ASID JOBNAME  ASCBADDR SELECTION CRITERIA
-----
000F OMVS      00FB0B80 ASID
0396 CICSCWAA 00FBBF80 ASID
```

Basics: Format 'Store Clock' fields

- **IPCS CBF *address* STR(TODCLOCK)**

- ▶ DFHL2BLOCK contains a timestamp at offset x'24'

```

0000                                6EC4C6C8 *                >DFH*        12FBD10C
0004  D3F2C2D3 D6C3D240 00000000 00000120 *L2BLOCK .....*        12FBD110
0014  00000000 00307CE8 80800000 00000002 *.....@Y.....*        12FBD120
0024  C1287308 54098CB9 131C0A00 0000FA00 *A.....*        12FBD130
0034  131C0A9D C4C6C8D3 D6C74040 01010000 *....DFHLOG ....*        12FBD140
    
```

- **IPCS CBF 12fbd130 STR(TODCLOCK) produces**

```

TODCLOCK: 00000000
          09/06/2007 12:46:04.451992
    
```

- Another option is to equate the address of the blocktime to a symbol; first list the address (in browse mode, command => **L 12fbd130**)

```

ASID(X'0040') ADDRESS(12FBD130.) STORAGE -----
Command ==> eq blocktime
12FBD130  C1287308  54098CB9  131C0A00  0000FA00  | A..... |
12FBD140  131C0A9D  C4C6C8D3  D6C74040  01010000  | ....DFHLOG .... |
12FBD150  0000F9CC  00000000  00000000  00000000  | ..9..... |
    
```

- **cbf blocktime str(todclock)** then produces the same results

```

TODCLOCK: 00000000
          09/06/2007 12:46:04.451992
    
```

Basics: RETP command

- **RETP** command

- ▶ ISPF command which displays the last 25 commands entered (both IPCS and ISPF)
- ▶ may be entered from either the IPCS or ISPF command line

```

----- IPCS PRIMARY OPTION MENU -----
      Retrieve
Options  Help
-----
      ISPF Retrieve Panel

Select the command
to be retrieved

                More:      +
1.  cbf blocktime str(t>
2.  eq blocktime
3.  =1
4.  ip 1 12fbd130
5.  IPCS CBF 12fbd130 S>
6.  ipcs cbf 12fbd134 s>
7.  ipcs cbf 12fbd124 s>
8.  f dfhl2block
9.  f dfhl2
10. verbx dfhpd640 'lg=>
11. ip select asid(x'40>
12. cbf rtct
13. st sys

dump and options
set
tents
analysis job to batch
, CLIST or REXX exec
functions
ntory
is
the IPCS dialog
log and list defaults

dialog

*****
* USERID - JOEUSER
* DATE - 07/09/06
* JULIAN - 07.249
* TIME - 15:02
* PREFIX - JOEUSER
* TERMINAL- 3278
* PF KEYS - 24
*****
    
```



Basics: Verbexit options

Keyword Functional area

AI = 0|2 Autoinstall Model Manager
AP = 0|1|2|3 Application Domain
 APS= <TASKID=nnnnn>
 AU = 0|2 CICS affinities utility
 BA = 0|1|2|3 Business application manager
 BR = 0|1|2|3 3270 bridge
 CC = 0|2 CICS catalog domain
 CP = 0|2 Common Programming Interface
 CQ = 0|1|2 Auto install model manager
 CSA= 0|2 CICS Common System Area
 DB2= 0|1|2|3 The CICS DB2 interface
 DD = 0|1|2|3 Directory Domain
 DH = 0|1|2|3 Document handling domain
 DLI = 0|2 CICS DL/I Interface
 DM = 0|1|2|3 Domain Manager
 DP = 0|1|2|3 Debug Profiles manager (630)
DS = 0|1|2|3 Dispatcher Domain
 DU = 0|2 Dump Domain
 EJ = 0|1 Enterprise JAVA (610)
 EM = 0|1|2|3 Event manager domain for BTS
 FCP= 0|2 File Control Program
 FT = 0|1|2|3 CICS WEB Interface
 ICP = 0|2 Interval Control Program
 IE = 0|1|2|3 IP ECI Domain (620)
 II = 0|1|2|3 IIOIP
 IND= 0|1|2|3 Page number indexes for output
 JCP= 0|2 Journal Control Program
KE = 0|1|2|3 CICS Kernel
LD = 0|1|2|3 Loader Domain
 LG = 0|1|2|3 Logger Domain
 LM = 0|1|2|3 Lock Manager domain
 ME = 0|2 Message domain
 MN = 0|1|2|3 Monitoring domain
 MRO= 0|2 CICS Multi-Region Operation

Keyword Functional area

NQ = 0|2 Enqueue Manager
 OT = 0|1|2|3 Object Transaction Domain (610)
PA = 0|2 Parameter manager domain
 PCP= 0|2 Program Control Program
 PCT= 0|2 Program Control Table
PG = 0|1|2|3 Program Manager Domain
 PI = 0|1|?? Pipeline Domain (640)
 PR = 0|2 Partner Resource management
 PT = 0|1|2|3 Partner Domain (620)
 RD = 0|2 Resource definition manager
 RM = 0|2 Recovery Management
 RX = 0|1|2|3 Recoverable EXCI domain
 RZ = 0|1|2|3 Request Streams (610)
 SH = 0|1 Scheduler services domain for BTS
 SJ = 0|1|2|3 JVM Domain (610)
SM = 0|1|2|3 Storage Manager domain
 SO = 0|1|2|3 Sockets domain
 SSA= 0|2 Static Storage Areas
 ST = 0|1|2|3 Statistics domain
 SZ = 0|1 Front End Programming Interface
 TCP= 0|1|2|3 Terminal Control Program
 TDP= 0|1|2|3 Transient Data Program
 TI = 0|1|2|3 Timer domain
 TMP= 0|2 Table Manager Program
TR = 0|1|2|3 Trace domain
 TRS= <trace selection parameters>
 TS = 0|1|2|3 Temporary Storage Program
 UEH= 0|2 User Exit Handler
 US = 0|1|2|3 User Domain
 WB = 0|1|2 The web interface
XM = 0|1|2|3 The transaction manager
 XRF= 0|2 The extended recovery facility
 XS = 0|1 Security Domain

Basics: Verbexit options *notes*

CICS Transaction Server is based on domain architecture. Each domain encapsulates the code and control blocks for a given function. Access to the data belonging to a given domain is via the domain interface. Each domain, with the exception of the application domain (AP), is responsible for a given CICS function. All domains are contained in each CICS address space.

For example, the Catalog domain (CC) is responsible for the data content and access to the CICS local and global catalogs. When other domains require access to information managed by the catalog domain, it is obtained using a catalog domain call.

The exception to the rule is the Application domain which defines the environment for application execution. The Application domain includes file control (FCP), terminal control (TCP), and Multi-region operation (MRO) which are not yet full domains, but have separate formatting routines.

To format the control blocks and data, there are normally three levels of detail available as noted in the chart on the prior page:

1. **Summary only**
2. **Full Control Block formatting**
3. **Both 1 and 2**

Note: If you omit the level number, it defaults to level 3 for those components that have a summary, and level 2 for those that do not. Specifying Level=0 **suppresses** the listing for that component.

Also note, as new function is introduced, or significantly changed, the release is identified in the chart. For example the Pipeline (PI) domain was introduced in R640 (CICS/TS R3.1).

The CICS Problem Determination Guide (Appendix A) contains two cross reference listings providing the control blocks formatted by IPCS keyword and the IPCS keyword used to format a given control block.

Types of Dumps



Types of Dumps: SLIP

- To list the name of the dump, issue the LIST command,

- ▶ **LIST TITLE**

```
SLIP DUMP ID=xxxx
```

- To see the slip command that corresponds to the dump, issue:

- ▶ **LIST SLIPTRAP**

```
SLIP
```

```
SET,MSGID=DFHEJ0102,JOBNAME=CICSTST1,ACTION=SVCD,ID=3001,  
JOBLIST=(OMVS,CICSTST1),DSPNAME=('OMVS'.*,'CICSTST1'.*),M  
L=1
```

In this example, the slip triggered on CICS console message DFHEJ0102 and dumped the CICSTST1 and OMVS address spaces and data spaces.

Types of Dumps: SLIP (registers)

- Registers at the time the SLIP triggered
ST REGS

CPU STATUS:

PSW=**470C2000 8185FB36**

(Running in PRIMARY, key 0, AMODE 31, DAT ON)

ENABLED FOR PER I/O EXT MCH

ASID(X'0396') 0185FB36. IEANUC01.IEAVTSMG+0476 IN READ ONLY NUCLEUS
ASCB918 at FBBF80, JOB(CICSTST1), for the home ASID
ASXB918 at 8FDD00 for the home ASID. No block is dispatched
HOME ASID: 0396 PRIMARY ASID: 0396 SECONDARY ASID: 0396

General purpose register values

0-1	00000001_0185FFC0	00000000_0406F000
2-3	00000000_7F4EAF64	00000000_7F4EAF6C
4-5	00000000_00000000	00000000_00000000
6-7	00000000_7F4EB334	00000000_7F4EAF04
8-9	00000000_00000000	00000000_00000001
10-11	00000000_7F4EAF08	00000000_00000000
12-13	00000000_0185FFC0	00000000_7F4EB290
14-15	00000000_8185FAF8	00000003_00000008

Types of Dumps: Console dump

- Title supplied by the operator in the dump command

- ▶ DUMP COMM=('DUMP OF HUNG CICS REGION')

*02 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

- ▶ R 02,JOBNAME=CICSTST1,END

IEE600I REPLY TO 02 IS;JOBNAME=CICSTST1,END

- The first time you look at the dump, you will see the following:

TIME-03:36:15 PM. CPU-00:00:02 SERVICE-81484 SESSION-02:50:26 SEPTEMBER 6,2007

Initialization in progress for DSNAME('DUMP.MV01.#MASTER#.D070906.T153350.S00289')

TITLE='DUMP OF HUNG CICS REGION'

- ▶ The title can be viewed using the LIST TITLE command

- ST REGS produces registers for **ASID 1** (*Master*)

- ▶ These are not meaningful registers for a console dump.

Types of Dumps: MVS dump options

SDATA

- If you request a CICS dump using
CEMT PERFORM SNAP

You will *not* have to set any specific SDATA dump parameters because the dump would be taken by CICS, which will supply its own parms.

- If you request a Console dump you should insure the following SDATA parameters are supplied:
 - ▶ ALLPSA, SQA, SUMDUMP, NUC, RGN, LPA, TRT, CSA, GRSQ

These can be specified using the SDATA parameter in the dump request, as in the following example:

- ▶ DUMP COMM=(dump title goes here)
- ▶ R nn,JOBNAME=(region-name),
SDATA=(PSA,SQA,SUM,NUC,RGN,LPA,TRT,CSA,GRSQ),end

Types of Dumps: CICS Dump

- The “Title” of a CICS Dump identifies the type of dump:

L TITLE

TITLE=CICS DUMP: SYSTEM=IYNX30 CODE=SR0001 ID=1/0004

- **Dump Summary**

- ▶ Produced from each of the CICS Verbexit commands
- ▶ Provides high-level diagnostic information

VERBX DFHPD640 'KE=3'

=== DUMP SUMMARY

DUMPID: 1/0004

DUMPCODE: SR0001

DATE/TIME: 6/09/07 13:50:14 (LOCAL)

MESSAGE: DFHSR0001 IYNX30 An abend (code 0C7/AKEA) has occurred at offset X'00000434' in program SECNDPGM.

SYMPTOMS: PIDS/5655M1500 LVLS/640 MS/DFHSR0001

RIDS/DFHSRP PTFs/UK05645 AB/S00C7 AB/UAKEA RIDS/SECNDPGM
ADRS/00000434

TITLE: (None)

CALLER: (None)

ASID: X'0040'

CICS Verbexit samples



CICS Verbexit - Introduction

This section will illustrate a variety of common CICS Verbexit commands, using a CICS System Dump produced from the following error message:

```
DFHSR0001 IYNX30 An abend (code 0C7/AKEA) has occurred  
at offset X'00000434' in program SECNDPGM.
```

Verbx DFHPD640 'KE=3'

Kernel Domain

Issue: [F *Running](#)

- To locate the task in error look for a ***YES*** in the ERROR column

===KE: Kernel Domain KE_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB	ERROR
0001	11F6FC00	KTCB Step	00000000			00000000	11FB2000	
0002	11F6F800	KTCB QR	00000000			12203030	11FB5000	
0003	11F6F400	KTCB RO	00000000			12203148	11FB4000	
0004	11F6F000	KTCB FO	00000000			12203260	11FB3000	
0005	11F8CC00	Not Running	00000000			121B2080	11FB4000	
0006	11F8C800	KTCB L8001	00000000			122038F0	121C2000	
0007	11F8C400	KTCB SL	00000000			12203490	121E7000	
0008	11F8C000	Not Running	00000000			121B2500	11FB5000	
0009	11FA9C00	***Running**	00000000			121B2680	121BC000	
000A	11FA9800	Not Running	122AA680	00003	CSOL	121D7500	121E7000	
...								
0044	12ECC880	***Running**	122AD680	00064	OOPS	3164E080	11FB5000	*YES*
0045	12EE2080	Unused						
0046	12EE2480	Unused						
0048	12F6A480	Not Running	122AB680	00019	CFQS	121B2380	11FB5000	

- Take note of the KE_NUM, TRAN #, and TRANSID

Verbx DFHPD640 'KE=3'

Stack entries for KE_NUM 0044

Issue: F 0044

KE_NUM	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFS	ERROR	NAME
0044	12EDB020	0130	Bot	91E01A00	91E01DAC		03AC		DFHKETA
0044	12EDB150	0330	Dom	91E19DC0	91E19FC8		0208		DFHDSKE
0044	12EDB480	0530	Dom	91E3F698	91E40690		0FF8		DFHXMTA
0044	12EDB9B0	0600	Dom	9249BCB0	9249CC60		0FB0		DFHPGPG
			Int	+02D4	9249BE38		0188		INITIAL_LINK
0044	12EDBFB0	0CD0	Dom	9264E270	92669D56		0000		DFHAPLI1
			Int	+24DC	9264EDF6		0B86		CICS_INTERFACE
0044	12EDCC80	0D60	Lifo	12669B10	9266AF0E		13FE		DFHEPC
0044	12EDD9E0	0770	Dom	92490360	924916E8		1388		DFHPGLE
			Int	+04F4	92490516		01B6		LINK_EXEC
0044	12EDE150	0CD0	Dom	9264E270	9229281E		0000	*YES*	DFHAPLI1
			Int	+3132	9264EDEE		0B7E		LE370_INTERFACE
			Int	+2F0A	926526F6		4486		INVOKE_FOR_RECURSION
0044	12EDEE20	04C0	Sub	9228FEB0	92290F6A		10BA		DFHSRP
0044	12EDF2E0	0E90	Dom	91E84748	91E88782		403A		DFHMEME
			Int	+3526	91E849C8		0280		SEND
			Int	+165A	91E87D5C		3614		CONTINUE_SEND
			Int	+3F62	91E85EBC		1774		TAKE_A_DUMP_FOR_CALLER
0044	12EE0170	0630	Dom	91F1B060	91F1BCD4		0C74		DFHDUDU
			Int	+0B16	91F1B250		01F0		SYSTEM_DUMP
			Int	+1924	91F1BFA4		0F44		TAKE_SYSTEM_DUMP

Verbx DFHPD640 'KE=3' notes

- Task 64 has abended 0C7. There are a number of domains which have knowledge of task 64, but the kernel is always a good place to begin. The summary information indicates task 64 was running at the time of the abend. It's associated with kernel ke_num 0044 and the transaction ID is "OOPS".
- Issuing a find for '0044 ' positions the cursor at the same line, issue a repeat find to locate the kernel stack entries for ke_num 0044 (note there may be additional hits before the stack entries are reached).
- Kernel stack entries are used for internal CICS modules, they're a combination register save area and working storage for the module. When control passes to another CICS module, a new stack entry is built and registers 14 through 12 are saved starting at offset x'C' into the stack of the module making the call.
- In the case of ke_num 0044, the stack labeled in error is for module DFHAPLI1. DFHAPLI1 is the module which causes the user application to be given control, it indicates control has been given to the user application program.

KE_NUM	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFS	ERROR	NAME
<u>0044</u>	<u>12EDE150</u>	<u>0CD0</u>	<u>Dom</u>	<u>9264E270</u>	<u>9229281E</u>	<u>0000</u>	<u>0000</u>	<u>*YES*</u>	<u>DFHAPLI1</u>
			Int	+3132	9264EDEC		0B7E		LE370_INTERFACE
			Int	+2F0A	926526F6		4486		INVOKE_FOR_RECURSION
0044	12EDEC20	04C0	Sub	9228FEB0	92290F6A		10BA		DFHSRP
0044	12EDF2E0	0E90	Dom	91E84748	91E88782		403A		DFHMEME

Verbx DFHPD640 'KE=3'

Kernel Error Table

Issue: F 'Error Table'

==KE: KE Domain Error Table Summary

ERR_NUM =====	ERR_TIME =====	KE_NUM =====	ERROR TYPE =====	ERR_CODE =====	MODULE =====	OFFSET =====
00000001	13:47:37	0044	TRAN_ABEND_PERCOLATE	---/ATCH	DFHPCP	000005E0
00000002	13:47:37	0044	TRAN_ABEND_PERCOLATE	---/ATCH	DFHZARQ	00002650
00000003	13:47:37	0044	TRAN_ABEND_PERCOLATE	---/ATCH	DFHETC	00000EA0
00000004	13:47:37	0044	TRAN_ABEND_PERCOLATE	---/ATCH	DFHCSAOF	00000048
00000005	13:47:37	0044	TRAN_ABEND_PERCOLATE	---/ATCH	DFHEPC	00000246
00000006	13:48:08	0044	TRAN_ABEND_PERCOLATE	---/AEIP	DFHPCP	000005E0
00000007	13:48:08	0044	TRAN_ABEND_PERCOLATE	---/AEIP	DFHEIP	00002498
00000008	13:50:12	0044	PROGRAM_CHECK	0C7/AKEA	DFHYI640	00000434

Kernel Error Table *notes*

- The Kernel error table is a wrap-around table that holds the most recent errors (up to 50 entries).
- Error numbers are assigned consecutively (in hex) by the kernel, starting from 00000001.
- The error number tells you the number of program checks and system abends that have occurred for this run of CICS. Not all of them have necessarily resulted in a system dump.
- You usually will want to focus on the most recent program check, in this example we will focus on error number 00000008.
- Examine the kernel error data -- there are a number of methods which can be used to locate the kernel error entries. I like to issue the command

F 'number: 00000008'

Verbx DFHPD640 'KE=3'

Kernel Error data

Issue: F 'NUMBER: 00000008' ← note there are 2 spaces between ':' and '0'

=KE: Error Number: 00000008

KERRD 11F69348 KERNEL ERROR DATA

```
0000 F0C3F761 C1D2C5C1 018400C7 0000FFFF C4C6C8C1 D7D3C9F1 1264E270 3164E080 *0C7/AKEA.d.G....DFHAPLI1..S...\.*
0020 122AD680 12ECC880 00000008 00000000 FF950007 00000000 079D1000 93E00704 *..O...H.....n.....l\..*
0040 00060007 00000000 93E00704 90000838 1300B208 13009F98 1300B270 13E006B4 *.....l\.....q.....\..*
```

Error Code: 0C7/AKEA Error Type: PROGRAM_CHECK Timestamp: C12874082F64C2FB

Date (GMT) : 06/09/07 Time (GMT) : 12:50:32.737356
Date (LOCAL) : 06/09/07 Time (LOCAL) : 13:50:12.737340

KE_NUM: 0044 KE_TASK: 12ECC880 TCA_ADDR: 122AD680 DS_TASK: 3164E080

Error happened in program DFHYI640 at offset 00000434

Error happened under the CICS RB.

Verbx DFHPD640 'KE=3'

PSW and registers

CICS Registers and PSW.

PSW: 079D1000 93E00704 **Instruction Length:** 6 **Interrupt Code:** 07 **Exception Address:** 00000000

Execution key at Program Check/Abend: 9 ← **Storage Protection is ACTIVE**

Space at Program Check/Abend: Basespace

REGISTERS 0-15

REGS 11F69398

0000	1300B208	13009F98	1300B270	13E006B4	13E00330	12EDE91C	00000000	130000D0	*.....q.....\....\....Z.....}* *...{...}\....\....\4...81\S....*
0020	1300B2C0	1300B0D0	13E00400	13E00504	13E003F4	13009EF8	93E006E2	00000000	

Data at PSW: 93E00704 Module: DFHYI640 Offset: **0000434**

PSWDATA 13E002D0

0000	C4C6C8E8	C9F6F4F0	58F0021C	58F0F0D0	58F0F014	58F0F00C	58FF000C	07FF5CC6	*DFHYI640.0...00}.00..00.....*F*
0020	C9D3D3C9	D55C0000	47F0F028	00C3C5C5	000000C8	00000014	47F0F001	98CEAC00	*ILLIN*...00..CEE...H.....00.q...*
0040	13E003AE	00000000	00000000	00000000	90ECD00C	4110F038	98EFF04C	07FF0000	*.\.....}...0.q.0<....*
0060	13E002F8	00000000	13E007A0	13E003A6	13E002F8	13E005EC	13E00A90	13E003C2	*.\.8.....\....\w.\.8.\....\....\B*
0080	00104001	00000008	E2C5C3D5	C4D7C7D4	F2F0F0F7	F0F9F0F6	F1F3F4F2	F3F6F0F2	*.. ..SECNDPGM2007090613423602*
...									
03E0	47F0B000	D203D08C	D0989640	9140D21D	D0A8A01C	4120D0A8	5020D0A0	9680D0A0	*.0..K.}.}qo j K.}y....}y&}.o.}.*
0400	4110D0A0	58F0A000	41009138	58C09080	05EF58C0	90E85820	912440F0	2000D201	*..}..0....j..{.....{.Y..j. 0..K.*
0420	8038A081	D201D0A0	8038960F	D0A1 FA10	D0A0A083	D2018038	D0A0960F	8039D210	*...aK.}...o.}...}..cK...}.o...K.*

Here is the failing instruction →

← **PSW points here**



PSW and Registers *notes*

PSW: 079D1000 **93E00704** **Instruction Length: 6** Interrupt Code: 07

Execution key at Program Check/Abend: 9

Space at Program Check/Abend: Basespace

- To locate the failing instruction, subtract the instruction length from the PSW. In this case
$$\underline{13E00704} - 6 = \underline{13E006FE}.$$
- The failing instruction at location 13E006FE is: **FA10 D0A0A083**
This is an Add Decimal instruction.
The interrupt code is a Data Exception due to operand 2 not being in packed format.
 - ▶ For more information on Program interrupts and instructions, refer to
z/Architecture Principles of Operation
- CICS will also provide important information about the environment taken from the SDWA. In this case the program was executing in key 9.
- When STGPROT=YES is specified in the SIT and the program is defined with EXECKey(USER) the program is given control in key 9.
- Storage around the PSW and each register is formatted.
 - ▶ However, it must be remembered **this is the storage at the time the dump was taken.**
Therefore, although the PSW and Registers shown for prior errors are correct, the DATA formatted for these errors has most likely changed, since it reflects storage from the current failure.

Verbx DFHPD640 'LD=3'

Loader Domain

==LD: PROGRAM REPERTOIRE

PGM NAME	USE	CNT.	USERS	LOADS	COPIES	LENGTH	USE	TYP	ATTRIBUTE	EXEC	DEFINITION	DATE/TIME	CPE	ADDR	STATUS
IGZINSH	0		0	0	0	00000000	APP	RPL	REUSABLE	USER	6/09/07	12:46:01	131A5258		UNUSED
IIBMMSGT	0		0	0	0	00000000	APP	RPL	REUSABLE	USER	6/09/07	12:46:01	131A5310		UNUSED
IIGZMSGT	0		0	0	0	00000000	APP	RPL	REUSABLE	USER	6/09/07	12:46:01	131A53C8		UNUSED
SECNDPGM	1		1	1	1	00001320	APP	RPL	REUSABLE	USER	6/09/07	12:47:24	131A8538		LOADED

==LD: PROGRAM STORAGE MAP

PGM NAME	ENTRY	PT	CSECT	LOAD	PT.	REL.	PTF	LVL.	LAST	COMPILED	COPY	NO.	USERS	LOCN	TYP	ATTRIBUTE	APE	ADDR
DFHCCNV	93C00028		DFHYA640	13C00000	640						1		0	ERDSA	RPL	RESIDENT		131F12D8
			DFHCCNV	13C001A8	0640	UK09263	I		18/11	13.35								
FIRSTPGM	93E00000		DFHYA620	13E00000	620						1		1	ESDSA	RPL	REUSABLE		12F3D8B0
SECNDPGM	93E002F8		DFHYI640	13E002D0	640						1		1	ESDSA	RPL	REUSABLE		12F3DC68

Loader Domain *notes*

- The Loader domain is responsible for loading and tracking all program usage, including exit modules.
- The Program repertoire is a listing of all programs known to the loader at the point the dump was taken. It contains information showing the count, number of users, number of times the program has been loaded, when it was defined to CICS and the program attributes.
- The program storage map is a sequential (by storage address) listing of all programs managed by the loader domain in the CICS address space. Information in the kernel error data gives the failing PSW address as [93E00704](#).
- Also notice the Dump Summary message
DFHSR0001 IYNX30 An abend (code 0C7/AKEA) has occurred
at **offset X'00000434'** in **program SECNDPGM**.

Subtracting x'434' from the PSW address of **13E00704** provides the address of the Load Point of the failing program, 13E002D0. In the loader domain program storage map, the SECNDPGM Entry Point is at 13E002F8, which is at x'28' from the Load Point. It resides in the ESDSA, there is one copy of the module which currently has a use count of 1.

- In this example, the Dump Summary informed us of the abending program. However, this technique can also be used to locate the abending program, when CICS cannot determine the actual program, by using the PSW and finding the program that encompasses that storage.

Verbx DFHPD640 'PG=3'

Program Manager Domain

==PG: PPTE SUMMARY

PPTE ADDRESS	PROGRAM NAME	MOD TYPE	LANG DEF	LANG DED	CEDF STAT	INST TYPE	AVAL STAT	DATA LOC	EXEC KEY	DPL SUBS	RE LOAD	LOAD STAT	HOLD STAT	USE COUNT
131A2DF0	IGZCPCC	PG	ASS	NDD	G	CED	E	B	U	F	N	L	C	1
131A2E48	IGZINSH	PG	ASS	NDD	G	CED	E	B	U	F	N	ND	T	
131A2EA0	IIBMMSGT	PG	ASS	NDD	G	CED	E	B	U	F	N	ND	T	
131A2EF8	IIGZMSGT	PG	ASS	NDD	G	CED	E	B	U	F	N	ND	T	
131A6768	SECNDPGM	PG	COB	CO2	R	CED	E	A	U	F	N	L	T	1

To find the task we are interested in, issue a find for the task number: F 00064

==PG: PTA SUMMARY FOR TRAN NUM : 00064, PTA ADDRESS : 122FC300

LOG-LVL : 2 SYS-LVL : 0 TASK-LLE : 00000000 PLCB : 12EDDCA0
 =PG: TASK PLCB SUMMARY
 PROG SECNDPGM LVL 2 LD 13E002D0 ENT 93E002F8 LEN 001320 ENV EXEC INV FIRSTPGM
 PROG FIRSTPGM LVL 1 LD 13E00000 ENT 93E00000 LEN 0002D0 ENV EXEC INV CICS

Program Manager Domain *notes*

- It controls functions involving programs, such as EXEC CICS Link and Load commands, and autoinstalling programs.
- It contains information about the following CICS control blocks
 - ▶ Load List elements - LLE's
 - ▶ Program Processing Table Entry - PPTe's
 - ▶ Program Level Control Blocks - PLCB's
 - ▶ Handle Table Blocks - HTB's
- It is useful to find out the following information:
 - ▶ If Program Autoinstall is being used
 - ▶ Program attributes, such as Execution Key, Data Location, Language, Concurrency, and CEDF Status

Verbx DFHPD640 'AP=3' Application Domain

===AP: AP DOMAIN TRANSACTION SUMMARY

Tran No	Tran Id	Orig Tran	TCA Addr	TWA Addr	EIB Addr	SEIB Addr	EIS Addr	EIUS Addr	Facility Type	Facility Id
00021	CSNC	CSNC	0005E080	007B6000	000640D0	0005E494	0005E388	00064008		
00024	CSHQ	CSHQ	122AC680	007B6000	12F390D0	122ACA94	122AC988	12F39008		
00029	CSNE	CSNE	122AD080	007B6000	12F3A0D0	122AD494	122AD388	12F3A008		
00064	OOPS	OOPS	122AD680	007B6000	130000D0	122ADA94	122AD988	13000008	TC	TC21

===AP: AP DOMAIN TRANSACTION CONTROL BLOCKS

TCA.00064 122AD680 Task Control Area (User Area)

```

0000 122AD780 00000001 12F2D720 000502A0 12EDEC20 00000000 00000000 00000008 *..P.....2P.....*
0020 0000064C 00000000 00000000 92669D56 131F9000 00000000 007B8000 1300005C *...<.....k.....#.....**
0040 93E00064 000867E4 12EDC878 93E00028 12EDBFB0 1264F26F 00000014 00004000 *1\.....U..H.l\.....2?.....*
```

SYS_TCA.00064 122AD780 Task Control Area (System Area)

```

0000 00000000 00000000 00000000 00000000 0000064C 121F27DC 0000007A 00000000 *.....<.....:.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 12EDDD00 00000000 *.....*
0040 13009EF8 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *...8.....*
```

SYSEIB.00064 122ADA94 System EXEC Interface Block

```

-0008                                     5CE2E8E2 C5C9C240 *                                     *SYSEIB *
0000 0135012F 0107249F D6D6D7E2 0000064C E3C3F2F1 00000004 00007D02 0A00000000 *.....OOPS...<TC21.....'.....*
```

USER31.00064 130015C0 USER storage above 16MB

```

0000 E4F0F0F0 F0F0F6F4 4C4CD9E4 E6D76E6E 00000000 00000000 00008460 00008460 *U0000064<<RUWP>>.....d-..d-*
0020 00000000 130015F0 00010000 12EDEC20 12655497 12656496 E4F0F0F0 F0F0F6F4 *.....0.....p...oU0000064*
```



Application Domain *notes*

- AP=3 Formats storage pertaining to each active transaction starting with the lowest task number. Storage includes:
 - ▶ Task control area or TCA
 - ▶ EXEC Interface blocks, including EIS, EIB and EIUS
 - ▶ User task storage
- SYS EIB + x'1B' contains the current EIB function code; these can be found summarized in the appendix of the Application Programming Reference. In this example the SYSEIB function code was:
020A --- EXEC CICS IGNORE CONDITION
- EIB + x'1B', if different from the SYS EIBFN, contains the last completed function code. In this example EIB function code identical to the SYSEIB.
- The EIUS + x'50' contains the address of the application register save area. Registers from the last EXEC CICS command are stored here (registers 14 through 12).
- The users task storage is displayed, and if a storage check zone is corrupt (storage violation) it will be flagged with one of the following messages:

```
** DFHPD0124 Storage violation detected at xxxxxxxx. Leading SAA is invalid.  
** DFHPD0125 Storage violation detected at xxxxxxxx. Trailing SAA is invalid.
```

where xxxxxxxx is the starting address of the allocated storage.

Verbx DFHPD640 'XM=3'

Transaction Manager domain

==XM: GLOBAL STATE SUMMARY

```

XM domain status:                               Initialised
Maximum user tasks (MXT):                     32
System currently at MXT:                         No
XXMATT user exit currently:                       Inactive
XM state lock currently held:                     No
XM trandef state lock currently held:             No
System attaches delayed for SOS:                 No

Force-purge has been issued:                   Yes
Kill has been issued:                             No

```

==XM: TRANSACTION SUMMARY

Tran id	Tran num	<u>TxnAddr</u> TxdAddr	Start code	Sys Tran	Status	DS token	Facility type	Facility token	AP token
CSOL	00003	122091B8 122CF780	C	Yes	ACT	00860003	None		122AA680 00000000
CSSY	00005	122094C8 122CF030	C	Yes	ACT	01900003	None		0005C080 01050000
CSSY	00006	12209650 122CF030	C	Yes	ACT	01100003	None		0005C680 01050000
OOPS	00064	<u>12209340</u> 12FDA6B0	T	No	ACT	04000029	Terminal	12F2D720	122AD680 00000000

Transaction Manager Domain *notes*

- The Transaction Manager domain lists all transactions (user and system) that currently exist.
- Useful information found in the XM domain:
 - ▶ Maximum User Task setting (MXT from SIT)
 - ▶ Current number of active and queued tasks
 - ▶ Transaction ID and transaction number for each active task
 - ▶ Type of transaction - System vs. Non-system
 - ▶ Start code associated with the task
 - ▶ Transaction Class or TCLASS information
 - ▶ Address of the transaction control block - TXN
- What time was the task attached by transaction manager?
 - ▶ The TXN+50 contains a store clock value of the time the task was attached
 - ▶ In our case the TXN address for task 00064 is **12209340**
 - ▶ On the command line issue: [IP CBF 12209340+50 STR\(TODCLOCK\)](#)
 - ▶ Results: **TODCLOCK: 00000000**
09/06/2007 12:50:32.730252
 - ▶ So task 00064 was attached at 12:50:32 (GMT) on September 6, 2007

Verbx DFHPD640 'TR=1'

Trace Domain

===TR: TRACE DOMAIN ABBREVIATED TRACE

Internal trace is STARTED : Table size is 4096K

Auxiliary trace is STOPPED : Current extent is DFHAUXT : Autoswitch status is OFF

GTF trace is STOPPED

<u>Task</u>	<u>TCB</u>	<u>Trace Point</u>	<u>Mod</u>	<u>Function</u>	
00064	QR	AP 1948	APLI	EVENT CALL-TO-LE/370	Establish_Ownership_Type SECNDPGM
00064	QR	AP 1949	APLI	EVENT RETURN-FROM-LE/370	Establish_Ownership_Type OK SECNDPGM
00064	QR	SM 0301	SMGF	ENTRY FREEMAIN	130015C8,TASK
00064	QR	SM 0302	SMGF	EXIT FREEMAIN/OK	
00064	QR	AP 1941	APLI	EXIT ESTABLISH_LANGUAGE/OK	COBOL2,1322F360,,NOT_DEFINED,LE370_RUNTIME
00064	QR	AP 1940	APLI	ENTRY START_PROGRAM	SECNDPGM,CEDF,FULLAPI,EXEC,NO,1322F360,00000000
00064	QR	SM 0301	SMGF	ENTRY GETMAIN	28,YES,RUWAPOL,TASK31
00064	QR	SM 0302	SMGF	EXIT GETMAIN/OK	130015C8
00064	QR	SM 0301	SMGF	ENTRY GETMAIN	37D0,YES,00,LE_TWA,TASK31
00064	QR	SM 0302	SMGF	EXIT GETMAIN/OK	13001608
00064	QR	AP 1948	APLI	EVENT CALL-TO-LE/370	Thread_Initialization SECNDPGM
00064	QR	AP 1949	APLI	EVENT RETURN-FROM-LE/370	Thread_Initialization OK SECNDPGM
00064	QR	SM 0301	SMGF	ENTRY GETMAIN	8458,YES,LE_RUWA,TASK31
00064	QR	SM 0302	SMGF	EXIT GETMAIN/OK	13004DE8
00064	QR	AP 1948	APLI	EVENT CALL-TO-LE/370	Rununit_Init_&_Begin_Invo SECNDPGM
00064	QR	AP 00E1	EIP	ENTRY IGNORE-CONDITION	0004,13009EF8
00064	QR	AP E160	EXEC	ENTRY IGNORE	CONDITION LENGERR COBOLII 00028
00064	QR	PG 0700	PGHM	ENTRY IGNORE_CONDITIONS	13009FA5
00064	QR	PG 0701	PGHM	EXIT IGNORE_CONDITIONS/OK	0
00064	QR	AP E161	EXEC	EXIT IGNORE	CONDITION LENGERR 0,0,COBOLII,00028
00064	QR	AP 00E1	EIP	EXIT IGNORE-CONDITION	OK 00F4,00000000
00064	QR	AP 1942	APLI	*EXC* Program-Check	START_PROGRAM,SECNDPGM,CEDF,FULLAPI,EXEC,NO,1322F360

Verbx DFHPD640 'TR=2' Trace Domain

AP 00E1 EIP EXIT IGNORE-CONDITION OK REQ(00F4) FIELD-A(00000000) FIELD-B(0000020A)
 TASK-00064 KE_NUM-0044 TCB-QR /007DA510 RET-93E006E2 TIME-13:50:12.7360027800 INTERVAL-00.0000005000 =018368=

AP 1942 APLI *EXC* - Program-Check FUNCTION(START_PROGRAM) PROGRAM(SECNDPGM) CEDF_STATUS(CEDF) EXECUTION_SET(FULLAPI)
ENVIRONMENT_TYPE(EXEC) SYNCONRETURN(NO) LANGUAGE_BLOCK(1322F360) COMMAREA(00000000 , 00000000) LINK_LEVEL(2)
SYSEIB_REQUEST(NO)

TASK-00064	KE_NUM-0044	TCB-QR	/007DA510	RET-924916E8	TIME-13:50:12.7375516394	INTERVAL-00.0015488593	=018369=
1-0000	01A00000	000000DA	00000000	00000000	B81B4EA0	00000000 02000100 E2C5C3D5	*.....+.....SECN*
0020	C4D7C7D4	13E002D0	93E002F8	00001320	07010101	01030202 1322F360 1322F360	*DPGM.\.}l\.8.....3-..3-*
0040	00001320	91E21102	00000000	00000000	00000000	00020102 91020201 02404040	*....jS.....j....*
0060	40404040	0000002D	1202E0E0	131F04D0	01FB53C8	12109AF8 00000001 00007000	*.....\.\.}...H...8.....*
0080	12EDDFE0	00280000	12ECC880	00007000	00007758	80000000 12EDE300 91E1F450	*...\...H.....T.j.4&*
00A0	12EDD9E0	91E83360	D88C0000	0030D5C0	041201D8	22020000 12EDD678 00008030	*.R\jY.-Q....N{...Q....O....*
00C0	01000000	00000000	00000000	91E1F5DE	91E1F77A	91E48D80 11E4C31E 00008958	*.....j.5.j.7:ju...UC....i.*
00E0	12212000	12EDDE38	91E3647A	12EDE208	F3F9F1F3	91E07AC2 00000000 80000001	*.....jT...S.3913j\B.....*
0100	12EDE118	00000007	120F4030	00000001	12EDDE38	00000008 12EDE178 12FB53C8	*.....H*
0120	1210F378	00000001	0005B680	0002020B	12285000	12240684 00000000 00000000	*.3.....&...d.....*
0140	00000000	00000000	00000000	00000000	02A50000	00000270 F10002A0 12EDDD60	*.....v.....1.....-*
0160	00000000	12FB5420	00E00400	000006D1	00680000	00000028 00000000 00000000	*.....\.....J.....*
0180	B5000000	00000000	01EDE1E8	00000001	00000040	0002DFEC 12EDDE38 00000090	*.....Y.....*
2-0000	F0C3F761	C1D2C5C1	018400C7	0000FFFF	C4C6C8C1	D7D3C9F1 1264E270 3164E080	*OC7/AKEA.d.G...DFHAPLI1.S...\.*
0020	122AD680	12ECC880	00000008	00000000	FF950007	00000000 079D1000 93E00704	*..O...H.....n.....l\..*
0040	00060007	00000000	93E00704	90000838	1300B208	13009F98 1300B270 13E006B4	*.....l\.....q.....\..*
0060	13E00330	12EDE91C	00000000	130000D0	1300B2C0	1300B0D0 13E00400 13E00504	*.\...z.....}{...}\...*\.*
0080	13E003F4	13009EF8	93E006E2	00000000	00000000	00000000 00000000 00000000	*.\.4...8l\S.....}\...*\.*
00A0	00000000	00000000	00000000	00000000	00000000	00000000 00000000 00000000	*.....}\...}\...}\...}\...*\.*
00C0	00000000	00000000	00000000	00000003	00000000	00000000 00000000 00000000	*.....}\...}\...}\...}\...*\.*
00E0	00000000	00000000	00000000	00000000	00000000	00000000 00000000 00000000	*.....}\...}\...}\...}\...*\.*
0100	00000000	00000000	00000000	00000000	FF950007	00000000 079D1000 93E00704	*.....l\.....n.....l\..*
0120	00060007	00000000	93E00704	90000000	1300B208	13009F98 1300B270 13E006B4	*.....l\.....q.....\..*\.*
0140	13E00330	12EDE91C	00000000	130000D0	1300B2C0	1300B0D0 13E00400 13E00504	*.\...z.....}{...}\...}\...*\.*
0160	13E003F4	13009EF8	93E006E2	00000000	00000000	00000000 00000000 00000000	*.\.4...8l\S.....}\...}\...}\...}\...*\.*



Trace Domain *notes*

- The prior pages show examples of formatting the CICS trace:
 - ▶ first with **TR=1**, which displays the Abbreviated trace table
 - ▶ Next with **TR=2**, which produces the Full trace.
 - ▶ Optionally (but not shown here), you can specify **TR=3** to display first the Abbreviated, followed by the Full trace.

- Exception (*EXC*) trace entries cannot be suppressed

- TR=2 times are LOCAL (compared to DS times, which are GLOBAL)

- Dynamic changes to Trace settings can be made using the CETR transaction; those same changes can be made static through various System Initialization Table (SIT) parameters or overrides
 - ▶ For further information about CETR settings and usage, refer to CICS-Supplied Transactions
 - ▶ For further information about SIT settings, refer to the CICS System Definition Guide

Verbx DFHPD640 'DS=1'

Dispatcher Domain

KEY FOR SUMMARY

T = TYPE OF TASK **S=SYSTEM N=NON-SYSTEM**
S = STATE OF TASK D=DISPATCHABLE S=SUSPENDED **R=RUNNING** A=RUNNING ABTERM YES J=RUNNING IN JVM E=RESUMED EARLY
F = PURGEABILITY FLAG P=PURGEABLE N=NOT PURGEABLE
P = PURGE STATUS N=NO PURGE X=PURGED P=PURGE PENDING A=ABTERM PENDING
TT = TIMEOUT TYPE IN=INTERVAL DD=DEADLOCK DELAYED DI=DEADLOCK IMMEDIATE
W = WAIT/SUSPEND TYPE M=WAIT_MVS S=SUSPEND C=WAIT_OLDC W=WAIT_OLDW
DTA= DISPATCHER TASK AREA
AD = ATTACHING DOMAIN
M = **TASK MODE**

DS_TOKEN	KE_TASK	T	S	F	P	TT	RESOURCE TYPE	RESOURCE_NAME	W	TIME OF SUSPEND	TIMEOUT DUE	DTA (DSTSK)	AD	M	SUSPAREA	XM_TXN_TOKEN
00000001	11F8CC00	S	S	N	N	-	ENF	NOTIFY	M	12:45:58.610	-	121B2080	DM	RO	12217998	
00040003	12F6A480	S	S	N	N	IN	FCCFQS		M	12:46:00.425	12:51:00.425	121B2380	XM	QR	12F624C0	12209C700000019C
00060003	11F8C000	S	S	N	N	-	TIEXPIRY	DS_NUDGE	S	12:46:03.584	-	121B2500	TI	QR	121B3860	
00080003	11FA9C00	S	R									121B2680	AP	CQ		
00100003	12F1C080	S	S	N	N	-	ICEXPIRY	DFHAPTIX	S	12:46:03.584	-	121FCC80	XM	QR	121B3BF0	122096500000006C
00900003	121FBC00	S	S	N	N	-	ICMIDNTE	DFHAPTIM	S	12:46:00.312	-	31669C80	XM	QR	31669C80	122094C80000005C
00040001	1222A080	S	S	N	N	IN	SMSYSTEM		S	12:45:58.614	12:50:58.614	316B2380	SM	QR	121B3920	
00060001	1222A480	S	S	N	N	-	smsyre	smva_ecb	M	12:45:59.462	-	316B2500	SM	QR	11FF4594	
00080001	12F1C880	S	S	N	N	-	TCP_NORM	DFHZDSP	W	12:50:32.730	-	316B2680	XM	QR	00058100	122099600000008C
00000029	12ECC880	N	R									3164E080	XM	QR		122093400000064C
00800003	12F38880	S	S	N	N	-	CSNC	MROQUEUE	M	12:46:01.968	-	31640080	XM	QR	131B603C	12209DF80000021C

Note: 'DS' Times are GMT!

Task Numbers

Dispatcher Domain *notes*

- The Dispatcher domain is most useful for examining task 'hang' conditions.
- For these types of problems, you should examine the type of resource(s) the tasks are waiting for. The excerpt below is a 'DS=1' sample output from a dump taken when transactions were backing up.

This dump was taken at 22:09:05 GMT

RESOURCE TYPE	RESOURCE_NAME	W	TIME OF SUSPEND
TCP_NORM	DFHZDSP	W	22:09:05.075
	LMQUEUE	S	20:33:20.283
	LMQUEUE	S	20:33:05.398
	LMQUEUE	S	20:33:30.540
	LMQUEUE	S	20:33:26.600
	LMQUEUE	S	20:33:25.978
	LMQUEUE	S	20:33:18.029
	LMQUEUE	S	20:33:24.446
	LMQUEUE	S	20:33:10.234

← TCP task matches the time of the dump

↙ these tasks were waiting 1 1/2 hours for a lock



Verbx MTRACE

Console Message display

*** MASTER TRACE TABLE ***

TAG	IMM DATA	-----	MESSAGE DATA	-----
0001	0000025D	N 8000000 MV23	07249 13:50:01.16 STC16806	00000281 *DFS3384I DATA SHARING STOPPED IM9C
0001	0000004B	N 4000000 MV23	07249 13:50:09.58 JOB00027	00000080 +06/09/07 13:50:10
0001	00000040	M C020000 MV23	07249 13:50:12.73 JOB17692	00000090 +DFHDP0100 IYNX30 067
0001	00000040	D	067 00000090	An unexpected exception response with reason code X'22'
		has been		
0001	00000040	D	067 00000090	returned by CICS file control to module DFHDPFM. The
		debugging		
0001	00000040	E	067 00000090	profiles file, DFHDPFMP, is not usable.
0001	00000040	N C020000 MV23	07249 13:50:12.73 JOB17692	00000090 <u>+DFHSR0001 IYNX30 An abend (code 0C7/AKEA) has occurred</u>
				<u>at offset X'00000434' in program SECNDPGM.</u>
0001	00000040	M C020000 MV23	07249 13:50:12.73 JOB17692	00000090 +DFHME0116 IYNX30 069
0001	00000040	D	069 00000090	(Module:DFHMEME) CICS symptom string for message
		DFHSR0001 is		
0001	00000040	D	069 00000090	PIDS/5655M1500 LVLS/640 MS/DFHSR0001 RIDS/DFHSRP
		PTFS/UK05645		
0001	00000040	E	069 00000090	AB/S00C7 AB/UAKEA RIDS/SECNDPGM ADRS/00000434
0001	00000040	N FFFF000 MV23	07249 13:50:12.73 JOB17692	00000090 +DFHDU0201 IYNX30 ABOUT TO TAKE SDUMP. DUMPCODE: SR0001
		, DUMPID: 1/0004		

References

- CICS Support Web Page
 - ▶ <http://www.ibm.com/software/htp/cics/tserver/support/>
 - Helpful references and links to CICS Info Center, Technotes, Flashes, “Must Gather” documents for defect support, SupportPacs, Redbooks, and more!

- Useful Manuals
 - ▶ CICS Operations and Utilities Guide
 - *Details and instructions for configuring IPCS for CICS*
 - ▶ CICS Problem Determination Guide
 - *Assistance for investigating dumps and abends*
 - ▶ CICS Trace Entries
 - ▶ CICS Data Areas and Supplementary Data Areas
 - ▶ CICS Supplied Transactions

 - ▶ z/OS® Interactive Problem Control System User's Guide
 - ▶ z/OS® Interactive Problem Control System Customization
 - ▶ z/OS® Interactive Problem Control System Commands
 - ▶ z/OS® Principles of Operation
 - *Information on program interrupts and instructions*

Summary

- IPCS Basics
 - ▶ What's needed
 - ▶ Commands
 - ▶ CICS verbexit
- Type of dumps:
 - ▶ SLIP Dumps
 - ▶ Console Dumps
 - ▶ CICS Dumps
- CICS verbexit samples
- CICS References

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: ibm.com/software/info/education/assistant
- Learn about the Electronic Service Request (ESR) tool for submitting problems electronically: www.ibm.com/software/support/viewlet/probsub/ESR_Overview_viewlet_swf.html
- Sign up to receive weekly technical My support emails: www.ibm.com/software/support/einfo.html

Questions and Answers