

Introduction to IPCS for CICS People

Ezriel Gross
Rocket Software

November 2019
Session GM



Agenda

- Interactive Problem Control System (IPCS)
- VERBEXIT and TRS trace
- CICS prerequisites
- SDUMP options
- List, Find, and other IPCS subcommands
- MVS Console Dump

Interactive Problem Control System (IPCS)

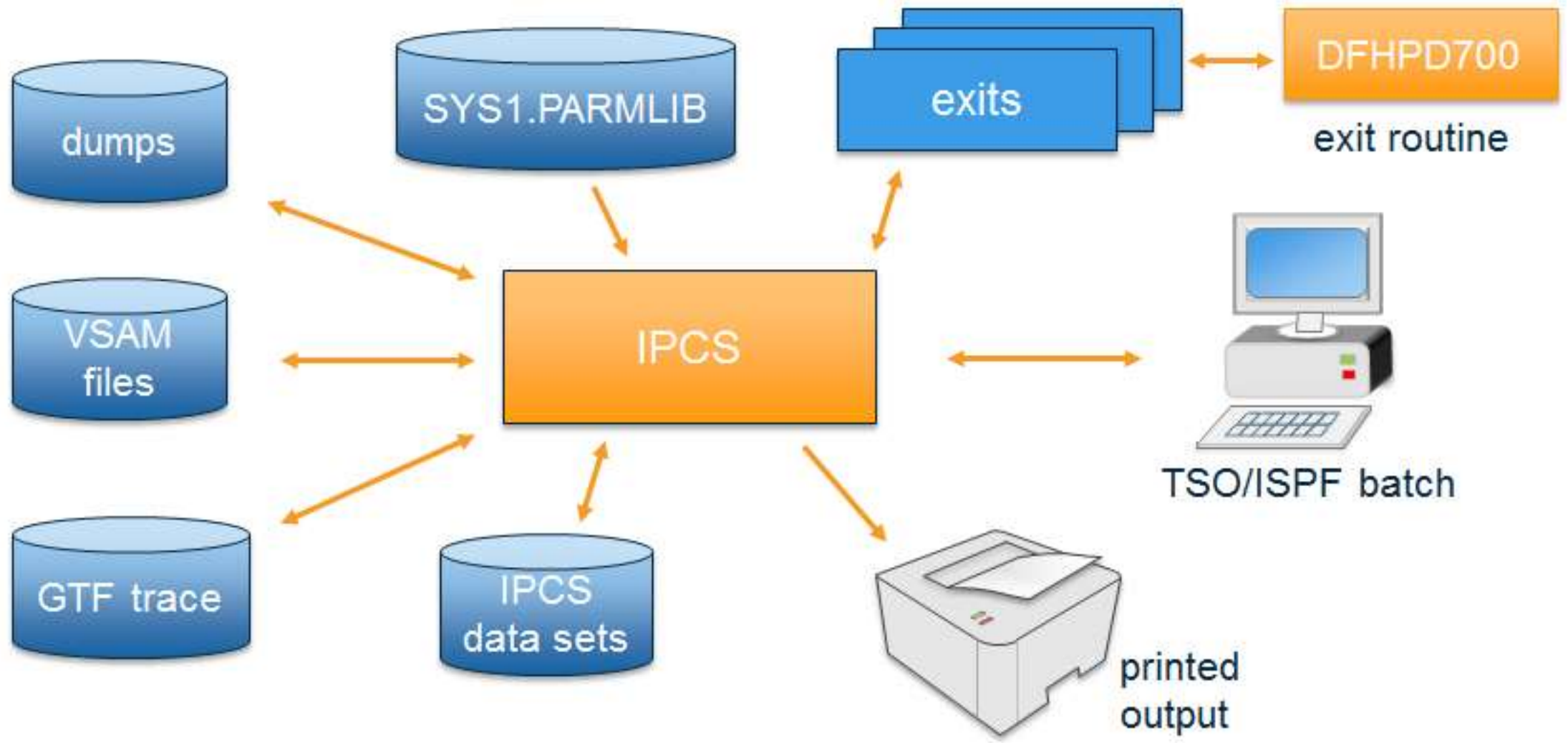
Interactive Problem Control System (IPCS)

IPCS provides an interactive facility for diagnosing software failures

“A component of MVS and z/OS that permits online problem management, interactive problem diagnosis, online debugging for disk-resident abend dumps, problem tracking, and problem reporting.”

[Source: IBM Terminology]

IPCS Overview



What you can do with IPCS

Using IPCS you can:

- Examine the data in a dump
- Locate and verify control blocks associated with some functions or system components
- Trace and verify chains of control blocks
- Perform contention analysis on key MVS resources
- Locate modules and MVS control blocks
- Execute user-written exits for some control blocks
- Keep a list of names and locations of control blocks and areas of the dump that you consider important

IPCS data sources

- System dump data sets
 - Standalone dumps
 - SVC dumps written by SDUMP macro to SYS1.DUMPnn data sets
 - ABEND (SYSMDUMP) dumps
 - Console dumps (also SDUMP)
- Generalized trace facility (GTF) data sets
- Virtual Storage Access Method (VSAM) data sets

Note: IPCS is a z/OS tool and largely MVS-oriented. Only through user-supplied exit facilities can it be customized to be a CICS tool

IPCS batch

- You can print all or part of a dump using JCL such as this
- The STEPLIB must contain DFHPDnnn, the IPCS VERBEXIT
- The DFHSDUMP DD statement defines the dump that IPCS will process
- IPCSPRNT defines the output destination for the report
- DROPDUMP removes any previous directory entry for the data set specified by DFHSDUMP DD

```

VIEW          CIRCLE.CICWS.JCL(IPCS) - 01.01          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000003 //*****
000004 //*
000005 //* THIS JOB FORMATS A CICS SDUMP BY INVOKING IPCS VIA
000006 //* THE TSO TERMINAL MONITOR PROGRAM IN A BATCH JOB.
000007 //*
000008 //*****
000009 //IPCSDUMP EXEC PGM=IKJEFT01,REGION=30M
000010 //STEPLIB DD DISP=SHR,DSN=CICS.V700BASE.CICS.SDFHLOAD
000011 // DD DISP=SHR,DSN=CICS.V700BASE.CICS.SDFHPARM
000012 // DD DISP=SHR,DSN=CICS.V700BASE.CICS.SDFHLINK
000013 //IPCSPARAM DD DISP=SHR,DSN=SYS2.PARMLIB
000014 // DD DISP=SHR,DSN=CICS.V700BASE.CICS.SDFHPARM
000015 //DFHSDUMP DD DISP=SHR,DSN=FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340
000016 //DFHSDUMP DD SYSOUT=*
000017 //IPCSDDIR DD DISP=SHR,DSN=EJG.DDIR
000018 //IPCSTOC DD SYSOUT=*
000019 //*PCSPRNT DD DISP=SHR,DSN=CIRCLE.CICWS.EZRIEL.DUMP
000020 //IPCSPRNT DD SYSOUT=*
000021 //SYSPROC DD DISP=SHR,DSN=SYS1.SBLSCLI0
000022 //SYSTSPRT DD SYSOUT=*
000023 //IPCSDUMP.SYSTSIN DD *
000024 IPCS NOPARM
000025 DROPDUMP DD(DFHSDUMP)
000026 SETDEF DD(DFHSDUMP) NOPROBLEM NOCONFIRM NOTERMINAL PRINT LIST
000027 VERBEXIT DFHPD700 'TRS=<TYPETR=(AP00E1,APE160-E161)>'
000028 END
000029 /*
000030 //*LIST 0A814000 LENGTH(800)
000031 //* FCP=2,ICP=2,TCP=2,TDP=2,TSP=2,DM=3,LD=3,PG=2,PCT=2'
***** Bottom of Data *****

```


Accessing IPCS

- Usually an option on ISPF main menu. Sometimes an option you access from other software products
- ISPF panels and libraries might be protected by security, e.g. RACF
- Access to dump data sets is controlled by security
- When you start IPCS, dump directory space is allocated

```
Dump directory name 'JXW.DDIR' will be used  
ALLOCATE FILE(IPCSDDIR) REUSE DSNAME('JXW.DDIR') SHR /* @P1C*/  
Dump directory 'JXW.DDIR' allocated to FILE(IPCSDDIR)  
***
```

IPCS – main 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      - JXW
*  DATE        - 16/06/17
*  JULIAN      - 16.169
*  TIME        - 13:56
*  PREFIX      - JXW
*  TERMINAL    - 3278
*  PF KEYS    - 24
*****

```

Enter END command to terminate IPCS dialog

We will cover the crucial options: 0,1,4, and 6, and option 3 in part

IPCS options

Option	lets you...
0 DEFAULTS	review and update IPCS default values
1 BROWSE	display a dump in full-screen mode
2 ANALYSIS	analyze the status of the mainframe and its components
3 UTILITY	copy dumps, for example from SYS1.DUMPnn to another data set
4 INVENTORY	review the dumps in the IPCS dump directory. You can open, close, or specify a dump as the default dump source
5 SUBMIT	submit problem analysis jobs for batch processing
6 COMMAND	enter commands or CLISTS and view the output
7 TUTORIAL	view online instructions about how to use IPCS

0 DEFAULTS

Setting default options is a good starting point

Scope:

You can set separate values for LOCAL and GLOBAL scope, or BOTH to use same values for both

GLOBAL default values provide continuity between sessions; LOCAL defaults allow each ISPF logical screen to be used independently.

```
----- IPCS Default Values ----- LOCAL updated
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 ==> LOCAL (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('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340')
```

```
Address Space ==>
```

```
Message Routing ==> NOPRINT TERMINAL NOPDS
```

```
Message Control ==> CONFIRM VERIFY FLAG(WARNING)
```

```
Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL ALIGN
```

Press ENTER to update defaults.

Use the END command to exit without an update.

0 DEFAULTS

```

Message Routing ==> NOPRINT TERMINAL NOPDS
Message Control ==> CONFIRM VERIFY FLAG(WARNING)
  
```

- **PRINT|NOPRINT** and **TERMINAL|NOTERMINAL** allow you to specify where your IPCS output goes. To view the IPCS table of contents (TOC), you must specify PRINT, because the TOC is written only to the IPCSPRINT file
- **CONFIRM|NOCONFIRM** states whether a confirmation message is displayed before taking some actions
- **VERIFY|NOVERIFY** specifies whether subsequent subcommands will produce output. FLAG(level) sets the lowest level for message output. Level can be ERROR, SEVERE, INFORMATIONAL, TERMINATING, or WARNING.

0 DEFAULTS

Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL ALIGN

- **MACHINE|NOMACHINE** specifies whether ASID, virtual address, and other parameters are to be displayed for the current data
- **REMARK|NOREMARK** specifies whether user remarks associated with a symbol are displayed
- **REQUEST|NOREQUEST** specifies whether to display a model LIST subcommand to display the user data
- **STORAGE|NOSTORAGE** specifies whether to display the data at the specified address or a default address
- **SYMBOL|NOSYMBOL** specifies whether to display the associated symbol with the dumped data

1 BROWSE

```

----- IPCS - ENTRY PANEL -----
Command ==>

CURRENT DEFAULTS:
Source ==> DSNAME('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340')
Address space ==> ASID(X'0627')

OVERRIDE DEFAULTS:                                     (defaults used for blank fields)
Source ==> DSNAME('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340')
Address space ==>
Password ==>

POINTER:
Address ==> 7000                                     (blank to display pointer stack)
Remark ==> Kernel anchor block                       (optional text)
  
```

1 BROWSE

- Option 1 BROWSE lets you view unformatted storage in a dump
- Typically, you want to look at areas of storage immediately before or after a damaged area of storage, or you might want to look at an unformatted control block
- Change the defaults to the name of the dump you want to browse and press ENTER
- A pointers screen is displayed with, initially, a single entry pointer set to the address 0
- Type s into the pointer (PTR) field to view unformatted storage
- You can also press TAB and overwrite the 0 address with the valid address that you need before pressing ENTER. For example, you could enter s into the PTR field, tab to the 0 address, and press ENTER

1 BROWSE - output

- This is a request to display address x7000 which is the kernel anchor block (KCB)
- Using PF7 and PF8 to scroll through storage
- Jump to any address displayed on the screen by typing in front of an address:
 - % for a 24-bit address
 - ? for a 31-bit address
 - ! for a 64-bit address
- If the first character is alphabetic, add a zero in front of the address or a period at the end

```

ASID(X'0627') ADDRESS(7000.) STORAGE -----
Command ==>
00007000 05006EC4 C6C8D2C5 D2C3C240 40404040 | ..>DFHKEKCB |
00007010 00000000 22200401 00000000 A2200A00 | .....S... |
00007020 00000000 A2201500 00000000 22200800 | ....S..... |
00007030 00000000 A2201700 00000000 A2201B00 | ....S.....S... |
00007040 7F6FBF38 A2201500 00000000 22202301 | "?..S..... |
00007050 00000000 A2201F00 00000000 22202C01 | ....S..... |
00007060 00000000 A2202800 00000000 A2202B00 | ....S.....S... |
00007070 00000000 A2203100 00000000 A2203500 | ....S.....S... |
00007080.:708F.--All bytes contain X'00'
00007090 00000000 22251B21 00000000 2247D000 | .....}. |
000070A0 00000000 2226BE08 00000000 2226D5A0 | .....N. |
000070B0 00000000 2226CD40 00000000 A226BB28 | .....S... |
000070C0 00000000 2226F108 00000000 00000000 | .....1..... |
000070D0.:70FF.--All bytes contain X'00'
00007100 00000000 223C5800 00000000 00000000 | ..... |
00007110 ! 00000000 223C5000 00000000 223E0000 | .....&..... |
00007120 00000000 2243F000 C0000036 00000000 | .....0.{..... |
00007130 00000034 0000003C 00000000 223EFA30 | ..... |
00007140 00000000 223EFAE0 0000AD8 00000000 | .....\.Q... |
00007150 00000000 223C41A8 00006B49 D2000000 | .....y.,.K... |
00007160 80200000 00000000 00000000 80008E7C | .....@ |
00007170 00000000 A2200C72 00000000 00009200 | ....s.....k. |
00007180 00000000 A2274680 00000000 0000A000 | ....s..... |
00007190 00000000 00007500 80000000 00000000 | ..... |
000071A0 00000000 00000000 00000001 00000001 | ..... |
000071B0 00000001 00000000 00000000 00000000 | ..... |
000071C0.:71FF.--All bytes contain X'00'
00007200 C3C3E5E3 F5F3C940 C3C3E5E3 F5F3C940 | CCVT53I CCVT53I |
00007210 C4C6C8C3 D3E34040 40404040 40404040 | DFHCLT |
00007220 C4C6C8C9 D9F0F0F0 C4C6C8E2 C9E34040 | DFHIR000DFHSIT |
00007230 E3F5F3C9 E9F7F2F2 001EB000 00000000 | T53IZ722..... |
  
```

1 BROWSE - address

- This shows the output of an address display using the command
`L 00000000_223C6000`
- You can also perform indirect addressing.

```

ASID(X'0627') ADDRESS(223C6000.) STORAGE -----
Command ==> 1 00000000_223C6000
                SCROLL ==> CSR
223C6000  C4C6C8D2  C5404040  00000001  00800000  | DFHKE ..... |
223C6010  00000000  00007000  80000000  00000000  | ..... |
223C6020  80000000  00000000  00000000  00000000  | ..... |
223C6030.:223C603F.--All bytes contain X'00'
223C6040  00000000  00000000  C7C1E3C5  C8C5C1C4  | .....GATEHEAD |
223C6050  00000000  22288338  00000000  22288488  | .....c.....dh |
223C6060  00000120  D0000000  00000000  00000000  | .....}..... |
223C6070  00000000  2227A8F8  00000000  A227AA48  | .....y8....s... |
223C6080  00000770  54000000  00000000  00000000  | ..... |
223C6090  00000000  2227EF08  00000000  A227F058  | .....s.0. |
223C60A0  00000790  50000000  00000000  00000000  | .....&..... |
223C60B0  00000000  22288338  00000000  22288488  | .....c.....dh |
223C60C0  00000120  D0000000  00000000  00000000  | .....}..... |
223C60D0  00000000  22288338  00000000  22288488  | .....c.....dh |
223C60E0  00000120  D0000000  00000000  00000000  | .....}..... |
223C60F0  00000000  22288338  00000000  22288488  | .....c.....dh |
223C6100  00000120  D0000000  00000000  00000000  | .....}..... |
223C6110  00000000  22204458  00000000  A22045A8  | .....s..y |
223C6120  00000500  50000000  00000000  00000000  | .....&..... |
223C6130  00000000  22288338  00000000  22288488  | .....c.....dh |
223C6140  00000120  D0000000  00000000  00000000  | .....}..... |
223C6150  00000000  22288338  00000000  22288488  | .....c.....dh |
223C6160  00000120  D0000000  00000000  00000000  | .....}..... |
223C6170  00000000  22204458  00000000  A22045A8  | .....s..y |
223C6180  00000500  50000000  00000000  00000000  | .....&..... |
223C6190  00000000  22288A60  00000000  A2288BB0  | .....-....s... |
223C61A0  00000270  54000000  00000000  00000000  | ..... |
223C61B0  00000000  222775E8  00000000  A2277738  | .....Y....s... |
223C61C0  00000710  50000000  00000000  00000000  | .....&..... |

```

RETP – Retrieve command

- Retrieve allows you to find and rerun commands that you previously entered

```

ASID(X'0627') ADDRESS(223C6000.) STORAGE -----
C . Retrieve
2 Options Help 00001 00800000 DFHKE .....
2 00000 00000000 .....
2 ISPF Retrieve Panel 00000 00000000 .....
2 in X'00'
2 Select the command 1E3C5 C8C5C1C4 .....GATEHEAD
2 to be retrieved 00000 22288488 .....c.....dh
2 More: + 00000 00000000 .....}.....
2 1. 1 00000000_223C6000 00000 A227AA48 .....y8...s...
2 2. =1 00000 00000000 .....
2 3. sd 00000 A227F058 .....s.0.
2 4. 4 00000 00000000 .....&.....
2 5. 1 00000 22288488 .....c.....dh
2 6. I 00000 00000000 .....}.....
2 7. 3.4 00000 22288488 .....c.....dh
2 8. RUN 00000 00000000 .....}.....
2 9. 2 00000 22288488 .....c.....dh
2 10. NEW 00000 00000000 .....}.....
2 11. 3 00000 A22045A8 .....s..y
2 12. RUN 00000 00000000 .....&.....
2 13. NEW 00000 22288488 .....c.....dh
2 00000 00000000 .....}.....
223C6150 00000000 22288338 00000000 22288488 .....c.....dh
223C6160 00000120 D0000000 00000000 00000000 .....}.....
223C6170 00000000 22204458 00000000 A22045A8 .....s..y
223C6180 00000500 50000000 00000000 00000000 .....&.....
223C6190 00000000 22288A60 00000000 A2288BB0 .....-...s...
223C61A0 00000270 54000000 00000000 00000000 .....
223C61B0 00000000 222775E8 00000000 A2277738 .....Y...s...
223C61C0 00000710 50000000 00000000 00000000 .....&.....

```

1 BROWSE – pointer stack

```

DSNAME('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340') POINTERS -----
Command ==> SCROLL ==> PAGE
ASID(X'0627') is the default address space
PTR  Address          Address space          Data type
-----
00001 7000.             ASID(X'0627')         AREA
      Remarks: Kernel Anchor Block
00003 223C5000.         ASID(X'0627')         AREA
      Remarks: Domain Gate Table Header
00007 223C6000.         ASID(X'0627')         AREA
      Remarks: Domain Gate Table entry for KE
***** END OF POINTER STACK *****

```

- Press PF3 to return to the previous screen where you started the BROWSE. The addresses you used have been added to the pointer stack.
- You can type a remark for each of these addresses
- You can use normal ISPF line commands to manipulate the entries, e.g. d- delete, r – repeat, i - insert

4 INVENTORY

- INVENTORY shows a list of the processing sources in your IPCS directory
- You can open or close a dump, and work with a dump just as you did in DEFAULTS
- INVENTORY is a useful central place from which to manage the dumps

```

IPCS INVENTORY - EJG.DDIR -----
Command ==>                                     SCROLL ==> PAGE

AC Dump Source                                     Status
-----
DSNAME('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340') . . . . . OPEN
Title=CICS DUMP: SYSTEM=CCVT53I  CODE=SR0001  ID=1/0006
Psym=PIDS/5655Y0400 LVLS/700 MS/DFHSR0001 RIDSD/DFHSRP PTFS/HCI7000 AB/S00C7
DSNAME('FUNDIP.DUMP.D0627.H01.FTS1.CCVT53I.S00343') . . . . . CLOSED
Title=CICS DUMP: SYSTEM=CCVT53I  CODE=SR0001  ID=1/0002
Psym=PIDS/5655Y0400 LVLS/700 MS/DFHSR0001 RIDSD/DFHSRP PTFS/HCI7000 AB/S00C4
DSNAME('FUNDIP.DUMP.D0627.H02.FTS1.CCVT53I.S00353') . . . . . CLOSED
Title=CICS DUMP: SYSTEM=CCVT53I  CODE=SR0001  ID=1/0001
Psym=PIDS/5655Y0400 LVLS/700 MS/DFHSR0001 RIDSD/DFHSRP PTFS/HCI7000 AB/S00C7
DSNAME('FUNDIP.DUMP.D0627.H02.FTS1.CCVT53I.S00354') . . . . . CLOSED
Title=CICS DUMP: SYSTEM=CCVT53I  CODE=SM0102  ID=1/0003
Psym=PIDS/5655Y0400 LVLS/700 MS/DFHSM0102 RIDSD/DFHSMAR PTFS/HCI7000 PRCS/000
DSNAME('FUNDIP.DUMP.D0627.H02.FTS1.CCVT53I.S00355') . . . . . ADDED
Title=CICS DUMP: SYSTEM=CCVT53I  CODE=SM0102  ID=1/0004
Psym=PIDS/5655Y0400 LVLS/700 MS/DFHSM0102 RIDSD/DFHSMAR PTFS/HCI7000 PRCS/000
***** END OF IPCS INVENTORY *****

```

6 COMMAND

Option 6 allows you to enter IPCS subcommands

- VERBEXIT is a very useful command that enables you to analyze CICS dumps
- You can run a 'baboooshka' IPCS instance within IPCS
- LIST displays storage from one or more addresses in the current dump

```

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

===> VERBX DFHPD700 'KE=1'

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

Formatting dumps

```
TIME-08:12:18 AM. CPU-00:00:00 SERVICE-11459 SESSION-00:15:55 JUNE 24,2016  
Initialization in progress for DSNAME('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340')
```

```
TITLE=CICS DUMP: SYSTEM=CCVT53I CODE=SR0001 ID=1/0006  
Dump written by z/OS 02.02.00-0 SVC dump - level same as IPCS level z/Architecture mode system  
May summary dump data be used by dump access? Enter Y to use, N to bypass.
```

Y

```
229,668 blocks, 955,418,880 bytes, in DSNAME('FUNDIP.DUMP.D0624.H04.FTS1.CCVT53I.S00340')
```

```
TIME-08:12:32 AM. CPU-00:00:01 SERVICE-40563 SESSION-00:16:09 JUNE 24,2016  
Dump of z/OS 02.02.00-0 - level same as IPCS level  
***
```

IPCS 'babushka'

- You can recursively call IPCS within the output of an IPCS command that you have already run
- Useful with the LIST command as you can display an address, then press PF3 and return to original page

```

IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> IPCS DFHPD700 'AP=1'          SCROLL ==> CSR
***** TOP OF DATA *****

* * * * * CICS 7.0.0 - IPCS EXIT * * * * *

CICS700 OPERANDS:

KE=1

=== SUMMARY OF ACTIVE ADDRESS SPACES

      ASID(hex):          JOBNAME:
      0627                CCVT53I

==KE: Licensed Edition

-- DFHPD0121I FORMATTING CONTROL BLOCKS FOR JOB CCVT53I

ADDRESS SPACE ASID NUMBER (HEX) = 0627

=== DUMP SUMMARY

DUMPID:    1/0006

DUMPCODE:  SR0001

DATE/TIME: 24/06/16 04:54:04 (LOCAL)

MESSAGE:   DFHSR0001 CCVT53I An abend (code 0C7/AKEA) has occurred at offse
  
```


CICS prerequisites

CICS prerequisites for VERBEXIT

- **IPCS exit control table**

To enable CICS to supply additional VERBEXITs to IPCS, you need this additional line in member BLSCECT of SYS1.PARMLIB

```
IMBED MEMBER(DFHIPCSP) ENVIRONMENT(ALL)
```

- **Access to CICS PARMLIB**

Member DFHIPCSP of current SDFHPARM with entries:

```
EXIT EP(DFHPD660 VERB(CICS660) ABSTRACT('V4 R1' analysis')
EXIT EP(DFHPD670 VERB(CICS670) ABSTRACT('V4 R2' analysis')
EXIT EP(DFHPD680 VERB(CICS680) ABSTRACT('V5 R1' analysis')
EXIT EP(DFHPD690 VERB(CICS690) ABSTRACT('V5 R2' analysis')
EXIT EP(DFHPD700 VERB(CICS700) ABSTRACT('V5 R3' analysis')
```

- **Access to CICS Problem Determination Exits**

Members DFHPD*nnn* of current SDFHLINK library

CICS prerequisites for VERBEXIT

- To enable other programs such as CICS to supply additional VERBEXITs to IPCS, you need the line `IMBED MEMBER (DFHIPCSP) ENVIRONMENT (ALL)` in `SYS1.PARMLIB(BLSCECT)`
- Next, edit `SYS1.PARMLIB(DFHIPCSP)`. You can copy this line from CICS library `SDFHPARM`. It informs IPCS what the VERBEXIT command is called, e.g. `CICS690` or `CICS 700`. It also informs IPCS of the name of the corresponding load module that IPCS must run, e.g. `DFHPD690` or `DFHPD700`.
- Finally, make sure IPCS can find the load modules by moving them to an IPCS library or adding the CICS `SDFHLINK` library in the concatenation for IPCS.
- Your environment might require some variations, but this is the basic process.

VERBEXIT and TRS trace

VERBEXIT format

- VERBEXIT allows you to drive a CICS-supplied IPCS dump exit , DFHPDxxx where xxx = CICS release number, that can format and understand CICS-specified control blocks and data areas

Format:

```
VERBX CICS700 'parameters'
```

Where parameters can be:

```
[ JOB = {jobname | CURRENT } ]
```

```
  [ UPPERCASE ]
```

```
  [ ,DEF= { 0 | 1 | 2 | 3 } ]
```

```
  [ ,component-keyword [ = { 0 | 1 | 2 | 3 } ] ]
```

VERBEXIT parameters

- JOB specifies which job in the dump is to be formatted.

If the dump was produced by the SDUMP macro for a dump code defined as related, the dump might contain > 1 CICS job, so you should specify JOB. If the dump code is defined as local, there is 1 dump so omit this parameter.
- UPPERCASE specifies that you want the dump printed in uppercase only
- DEF specifies the level number defaults for formatting the dump data
 - 0 – Suppress. For those components not in a specified list of keywords, suppress all formatting
 - 1 – Summary. For those components not in a specified list of keywords, and where applicable, produce only a formatted summary from the control blocks. A summary is not available for all keywords.
 - 2 – Full. For those components not in a specified list of keywords, format all the control block information in full
 - 3 – Summary and full. For those components not in a specified list of keywords, format all control blocks and (where applicable) the summary information

VERBEXIT component keywords

General syntax, keyword=level

Levels:

0 Suppress all output from the component

1 Produce summary output only

Note: Not all components have a summary, in which case 1 acts as 0

2 Produce detail output only

3 Produce both summary and detail

For the IND keyword

1 block index sorted by address

2 sorted by block name

3 both summary and detail data

For the TR keyword

1. implies abbreviated trace

2 implies full trace

3 both abbreviated and full trace

CICS Verbx keywords

AI=0,2	Auto-install Model Manager	EC=0,1,2,3	Event Capture Domain.
AP=0,1,2,3	AP Domain	EM=0,1,2,3	Event Manager Domain
APS=<parms>	AP selection	EP=0,1,2,3	Event Processing Domain
BA=0,1,2,3	BAM Domain	FCP=0,2	File Control Program
BR=0,1,2,3	Bridge Domain	FT=0,1,2,3	Feature Table
CC=0,2	Catalog control blocks	ICP=0,2	Interval Control Program
CP=0,2	Common Programming Interface	IE=0,1,2,3	IP ECI domain
CQ=0,1,2,3	Console Queue Component	II=0,1,2,3	IIOP Domain
CSA=0,2	Common System Area	IND=0,1,2,3	block index
CV=0,1,2,3	CCSID conversion interface	IS=0,1,2,3	IP Interconnectivity Domain
DB2=0,1,2,3	CICS/DB2 interface	JCP=0,2	The Journal Control Area
DD=0,1,2,3	Directory Manager Domain	KE=0,1,2,3	Kernel Domain
DH=0,1,2,3	DH Domain	LD=0,1,2,3	Loader Domain
DLI=0,2	DL/I control blocks	LG=0,1,2,3	Log Manager Domain
DM=0,1,2,3	Domain Manager	LM=0,1,2,3	Lock Manager
DP=0,1,2,3	Debug Profile	ME=0,2	Message Domain control blocks
DS=0,1,2,3	Dispatcher Domain	ML=0,1,2,3	Markup Language Domain
DU=0,2	Dump domain control blocks	MN=0,1,2,3	Monitoring

APS and TRS VERBX keywords

The APS and TRS selection keywords allow you to select which task or trace records will be displayed

```
VERBX DFHPD700
```

```
'TR=3, TRS=<TASKID=038>, PG=1,  
AP=3, APS=<TASKID=38>'
```

NEW TK and TKS Keyword

CICS TS 5.3+

[Keyword to control block map](#)

CICS Verbx keywords

MQ=0,1,2,3 MQ Interface	SO=0,1,2,3 Sockets Domain
MRO=0,1,2,3 Multiregion Operation blocks	SSA=0,2 Static Storage Areas
NQ=0,1,2,3 Enqueue Manager	ST=0,1,2,3 Statistics Domain
OT=0,1,2,3 OT Domain	SZ=0,2 FEPI Interface
PA=0,2 Parameter Domain	TCP=0,1,2,3 Terminal Control
PCT=0,2 Transaction definitions	TDP=0,1,2,3 Transient Data
PG=0,1,2,3 Program Manager	TI=0,1,2,3 Timer
PI=0,1,2,3 Pipeline Domain	TMP=0,2 Table Manager Program
PR=0,2 Partner Resource blocks	TR=0,1,2,3 Trace domain
PT=0,1,2,3 Partner Domain	TRS=<parms> Trace selection
RD=0,2 Resource definition	TS=0,1,2,3 Temporary Storage
RL=0,1,2,3 Resource Life Cycle Domain	UEH=0,2 User Exit Handler
RM=0,1,2,3 Recovery Manager Domain	US=0,1,2,3 User Domain
RS=0,1,2,3 Region Status Domain	WB=0,1,2,3 Web Domain
RX=0,1,2,3 Recoverable Resource Domain	WU=0,1,2,3 CICS Management Client I/F
RZ=0,1,2,3 Request Stream Domain	W2=0,1,2,3 The Web 2.0 Domain
SH=0,1,2,3 Scheduler Services Domain	XM=0,1,2,3 Transaction Manager
SJ=0,1,2,3 JVM Domain	XRF=0,2 Extended Recovery
SM=0,1,2,3 Storage Manager	XS=0,1,2,3 Security Domain

APS example

```
VERBX DFHPD700  
'APS=<TASKID=6376>'
```

DUMP SUMMARY always appears
regardless of the VERBEXIT keyword

```
* * * * * CICS 7.0.0 - IPCS EXIT * * * * *
```

```
CICS700 OPERANDS:
```

```
APS=<TASKID=06376>
```

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

```
ASID(hex):      JOBNAME:  
0627            CCVT53I
```

```
==KE: Licensed Edition
```

```
-- DFHPD0121I FORMATTING CONTROL BLOCKS FOR JOB CCVT53I
```

```
ADDRESS SPACE ASID NUMBER (HEX) = 0627
```

```
=== DUMP SUMMARY
```

```
DUMPID: 1/0006
```

```
DUMPCODE: SR0001
```

```
DATE/TIME: 24/06/16 04:54:04 (LOCAL)
```

```
MESSAGE: DFHSR0001 CCVT53I An abend (code 0C7/AKEA) has occurred at offse
```

APS example

AP selective keyword allows you to view control blocks of an individual task

SYSEIB.06376 226A0B88 System EXEC Interface Block

```

-0008          5CE2E8E2 C5C9C240 *          *SYSEIB *      226A0B80

0000 0045402F 0116176F D7C1E8F1 0006376C *.. ....?PAY1...%* 226A0B88
0010 C3D7F2F0 0000017E 040C7D06 02000000 *CP20...=...'.....* 226A0B98
0020 000000D7 C1E8D9D6 D3D34000 00000000 *...PAYROLL .....* 226A0BA8
0030 000000D7 C1E8D9D6 D3D34000 00000000 *...PAYROLL .....* 226A0BB8
0040 00000000 00000000 00000000 00000000 *.....* 226A0BC8
0050 00000000 00          *.....* 226A0BD8
  
```

EIUS.06376 25700008 EXEC Interface User Structure

```

0000 00E86EC4 C6C8C5C9 E4E24040 40404040 *.Y>DFHEIUS *      25700008
0010 25700878 00000000 2570D5D0 00000000 *.....N}....* 25700018
0020 00000000 00000000 00000000 00000000 *.....* 25700028
0030 00000000 25700100 2570B628 25712D60 *.....-* 25700038
0040 00000000 00000000 00000000 A2E233D4 *.....sS.M* 25700048
0050 00027C88 2247D000 238FC56C 238FC4AC *..@h..}...E%..D.* 25700058
  
```

...

EIB.06376 25700100 EXEC Interface Block

```

-0010 00656EC4 C6C8C1D7 6DC4C6C8 C5C9C25C *..>DFHAP_DFHEIB** 257000F0

0000 0045402F 0116176F D7C1E8F1 0006376C *.. ....?PAY1...%* 25700100
0010 C3D7F2F0 0000017E 040C7D06 02000000 *CP20...=...'.....* 25700110
0020 000000D7 C1E8D9D6 D3D34000 00000000 *...PAYROLL .....* 25700120
0030 000000D7 C1E8D9D6 D3D34000 00000000 *...PAYROLL .....* 25700130
0040 00000000 00000000 00000000 00000000 *.....* 25700140
0050 00000000 00          *.....* 25700150
  
```

APS example

AP selective keyword
 recursively issues a
 VERBEXIT to display LE
 control blocks

```
IPCS OUTPUT STREAM ----- FOUND: LINE 2539 COL 1
Command ===>                                SCROLL ===> CSR
```

```
VERBEXIT CEEERRIP ASID(0627) CAA(25711E88) DSA(25712D60) ALL
```

```
*WARNING LEDATA could not obtain the TCB address from location 0000021C
```

```
APS=<TASKID=06376>
```

```
*****
                        LANGUAGE ENVIRONMENT DATA
*****
```

```
Language Environment Product 04 V02 R02.00
```

```
TCB: 00000000          LE Level: 1B          ASID: 0627
```

```
Active Members: COBOL
```

```
CEECAA: 25711E88
+000000 FLAG0:00 LANGP:08 BOS:25712B20 EOS:25713B20
+000044 TORC:00000000 TOVF:800335B8 ATTN:2570DFA0
+00015C HLLEXIT:00000000 HOOK:50C0D064 0DC058C0 C0060DCC
+0001A4 DIMA:0002A8C4 ALLOC:0700C3C8 STATE:0700C3C8
+0001B0 ENTRY:0700C3C8 EXIT:0700C3C8 MEXIT:0700C3C8
+0001BC LABEL:0700C3C8 BCALL:0700C3C8 ACALL:0700C3C8
+0001C8 DO:0700C3C8 IFTRUE:0700C3C8 IFFALSE:0700C3C8
+0001D4 WHEN:0700C3C8 OTHER:0700C3C8 CGOTO:0700C3C8
+0001F0 CGENE:40404040 CRENT:00000000 CTHD:40404040
+000210 EDCV:00000000 CEDB:40404040 EDCOV:40404040
+000258 TCASRV_USERWORD:00000000 TCASRV_WORKAREA:2570D768
+000260 TCASRV_GETMAIN:00000000 TCASRV_FREEMAIN:00000000
+000268 TCASRV_LOAD:80030A68 TCASRV_DELETE:80030988
```

APS example

Traceback shows more than CICS trace as it shows COBOL dynamic call stack as well as CICS links

Traceback:

DSA	Entry	E Offset	Statement	Load Mod	Program
1	PAYBUS1	+0000BF4			PAYBUS1
2	CEECRINV	+0000306			CEECRINV
3	CEECRINI	+0000B62			CEECRINI

DSA	DSA Addr	E Addr	PU Addr	PU Offset	Comp Date	Compile Attr
1	25712D60	25511028	25511028	+0000BF4	20160624	COBOL
2	25712BB8	23ECA088	23ECA088	+0000306	20150304	CEL
3	25712B38	23E94F08	23E94F08	+0000B62	20160420	CEL

Control Blocks Associated with the Thread:

Thread Synchronization Queue Element (SQEL): 25711C68

+000000	25711C68	40404040	40404040	40404040	40404040	40404040	40404040
+000020	25711C88	-	+00005F	25711CC7			same as above

Information for enclave PAYPGM1

Information for thread 8000000000000000
 PCB Address: 25703648
 TCB Address: 00000000

The starting DSA was located from the EDB of the child enclave.

Traceback:

DSA	Entry	E Offset	Statement	Load Mod	Program
1		-40404002			
2		+70F6403E			
WARNING An invalid DSA pointer was found on traceback while processing D					

APS example

LE control blocks contain COBOL addresses that would otherwise be difficult to chain up to using only CICS control blocks

```
----- Control blocks for program PAYBUS1 -----
CLLE: 25713E70
+000000 PGMNAME:PAYBUS1  TGT_FLAGS:01      LANG_LST:00000000
+000010 INFO_FLAGS:9481  LOAD_ADDR:A5511028  TGT_ADDR:25713F50
+00001C LE_TOKEN:00000000  WSA_ADDR:00000000
+000030 THD_STAT:00000001  THD_CNT:00000000
+000038 OPEN_NON_EXT_FILES:00000000
```

```
COB Main Static Rent WSA PgmInit StgInit
```

```
      TGT: 25713F50
+000048 IDENT:3TGT  LVL:06      FLAGS:68030260  RUNCOM:25713B98
+00005C COBVEC:00066ECC  #FCBS:00000000  WS_LEN:000062E
+000070 SMG_WRK:00000000  CAA:25711E88   LEN:00000208
+00008C EXT_FCBS:00000000  OUTDD:SYSOUT   ABINF:25511726
+0000FC TESTINF:00000000  PGMADDR:25511028  1STFCB:00000000
+000114 WS_ADDR:25714228  1STEXTFCB:00000000
```

```
Main AM31 PgmInit LibTrm
Static RES InitPgmSetup Ans85
```

```
Exiting COBOL Environment Data
Exiting Language Environment Data
Control returned successfully to CICS Verbexit
```

```
-- DFHPD0122I END OF DUMP FOR JOB CCVT53I
```

```
***** END OF DATA *****
```

TRS trace parameters

TRS selection criteria include

- TASKID - task number
- TERMID - terminal ID
- TRANID - transaction ID

Additional TRS selection criteria

EXCEPTION - exception trace records only

TYPETR - selective trace records

The following example selectively prints trace records that match the supplied domain id and point id

```
TR=3, TRS=<TASKID=6376, TYPETR= (AP00E1, APE160-E161) >
```

Refer to IBM Knowledge Center topic

https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.5.0/reference/utilities/dfha65k.html for a complete list of trace selection parameters

TRS example

```
verbx dfhpd700
'tr=1, trs=<taskid=99,
typetr=(ap00e1, ape160-e161)>`
```

This allows you to profile just the CICS commands in a COBOL program

```
00099 QR    AP 00E1 EIP  ENTRY ASKTIME-ABSTIME
00099 QR    AP E160 EXEC  ENTRY ASKTIME                AT X'A570C448', COBOLII,0
00099 QR    AP E161 EXEC  EXIT  ASKTIME                3675984580368 AT X'A570C
00099 QR    AP 00E1 EIP  EXIT  ASKTIME-ABSTIME        OK
00099 QR    AP 00E1 EIP  ENTRY FORMATTIME
00099 QR    AP E160 EXEC  ENTRY FORMATTIME            3675984580368 AT X'2570C
00099 QR    AP E161 EXEC  EXIT  FORMATTIME            3675984580368 AT X'2570C
00099 QR    AP 00E1 EIP  EXIT  FORMATTIME            OK
00099 QR    AP 00E1 EIP  ENTRY GET-CONTAINER
00099 QR    AP E160 EXEC  ENTRY GET                    'request-cont    ' AT X'
00099 QR    AP E161 EXEC  EXIT  GET                    'request-cont    ' AT X'
00099 QR    AP 00E1 EIP  EXIT  GET-CONTAINER          OK
00099 QR    AP 00E1 EIP  ENTRY RECEIVE-MAP
00099 QR    AP E160 EXEC  ENTRY RECEIVE                'DATAM1 ' AT X'2570AED0'
00099 QR    AP E161 EXEC  EXIT  RECEIVE                'DATAM1 ' AT X'2570AED0'
00099 QR    AP 00E1 EIP  EXIT  RECEIVE-MAP            OK
00099 QR    AP 00E1 EIP  ENTRY PUT-CONTAINER
00099 QR    AP E160 EXEC  ENTRY PUT                    'request-cont    ' AT X'
00099 QR    AP E161 EXEC  EXIT  PUT                    'request-cont    ' AT X'
00099 QR    AP 00E1 EIP  EXIT  PUT-CONTAINER          OK
00099 QR    AP 00E1 EIP  ENTRY LINK
00099 QR    AP E160 EXEC  ENTRY LINK                    'DATABUS1' AT X'2570AEE8
00099 QR    AP 00E1 EIP  ENTRY ASSIGN
00099 QR    AP E160 EXEC  ENTRY ASSIGN                    COBOLII 00053
00099 QR    AP E161 EXEC  EXIT  ASSIGN                    0,0, COBOLII,00053
00099 QR    AP 00E1 EIP  EXIT  ASSIGN                    OK
00099 QR    AP 00E1 EIP  ENTRY GET-CONTAINER
00099 QR    AP E160 EXEC  ENTRY GET                    'request-cont    ' AT X'
00099 QR    AP E161 EXEC  EXIT  GET                    'request-cont    ' AT X'
00099 QR    AP 00E1 EIP  EXIT  GET-CONTAINER          OK
00099 QR    AP 00E1 EIP  ENTRY READ
00099 QR    AP E160 EXEC  ENTRY READ                    'PAYROLL ' AT X'25713AC0
00099 QR    AP E161 EXEC  EXIT  READ                    'PAYROLL ' AT X'25713AC0
00099 QR    AP 00E1 EIP  EXIT  READ                    OK
```

Some useful VERBX keywords 1

Dump Summaries

Can provide invaluable debugging information

Kernel Domain (KE)

The level 1 summary provides information all tasks in the system

Dispatcher Domain (DS)

The summary shows which task is running/waiting - resource information

Transaction Manager Domain (XM)

The summary gives detailed information about related transaction control blocks

Some useful VERBX keywords 2

Application Domain (AP)

The AP summary gives summary information on all transactions running. Use the APS parameter to limit the AP detail output to a single transaction

Trace Domain (TR)

If trace is turned on, can speed up debugging process

Program Manager Domain (PG)

The summary gives information on programs

Storage Manager Domain (SM)

The detailed information can be invaluable for solving storage problems

Output from VERBX

VERBX with no options produces a complete formatted dump

Use component keywords to narrow the focus

This output is for

```
VERBX CICS700
```

```
'XM=1,DS=1,KE=1,AP=1'
```

The output is listed in order of summary, and then alphabetical PARM sequence, not the order in which they were requested.

In this example: summary, dispatcher, kernel, user, and task manager

```
* * * * * CICS 7.0.0 - IPCS EXIT * * * * *
```

```
CICS700 OPERANDS:
```

```
XM=1,DS=1,KE=1,AP=1
```

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

ASID(hex):	JOBNAME:
0627	CCVT53I

```
==KE: Licensed Edition
```

```
-- DFHPD0121I FORMATTING CONTROL BLOCKS FOR JOB CCVT53I
```

```
ADDRESS SPACE ASID NUMBER (HEX) = 0627
```

```
=== DUMP SUMMARY
```

```
DUMPID: 1/0006
```

```
DUMPCODE: SR0001
```

```
DATE/TIME: 24/06/16 04:54:04 (LOCAL)
```

```
MESSAGE: DFHSR0001 CCVT53I An abend (code 0C7/AKEA) has occurred at offse
```

List, Find and other IPCS subcommands

IPCS List subcommand

Displays storage from one or more addresses in the current dump

```
{ LIST } address [ :address ] [ LENGTH (length) ]
{ L }
```

- address can be:
 - Literal address, e.g. `L A773F0 . Length (x'200')` = list the data at address 00A773F0 for 512 bytes
 - Note:** If the initial digit is A-F, the literal address must end with a period
 - Relative address, e.g. `L +50` = display storage 50 bytes from current address
 - General-purpose register, e.g. `L 0r:15r` = display the contents of registers
 - Indirect address, e.g. `L 107FC0 . +C8?` = display storage addressed by offset C8 from address 107FC0. It is a 31-bit address.
- Length is optional. Can be specified in decimal (nnn), hexadecimal (X'xxx'), or binary (B'bbb') notation)

IPCS LIST example

```
VERBX DFPD700 'SM=3'
```

Using the IPCS 'baboooshka' method, you can issue an IPCS list to view storage

```
IPCS OUTPUT STREAM ----- FOUND: LINE 5616 COL 2
Command ==> ipcs list 25704570 length(x'420')          SCROLL ==> CSR
```

```
SCE.U0006376 00000050_4070B080 Storage Element Descriptor
```

```
0000 00000050 407089B0 00000000 22554ADC *...& .i.....%.* 4070B080
0010 2570D580 00008C00 2242FE80 00000000 *..N.....* 4070B090
0020 D0F0889C 0C8DD44E 0006376C D7C1E8F1 *}0h...M+...%PAY1* 4070B0A0
```

```
SCE.U0006376 00000050_407089B0 Storage Element Descriptor
```

```
0000 00000050 4070AA70 00000050 4070B080 *...& .....& ...* 407089B0
0010 25704990 00008BF0 2242FE80 00000000 *.....0.....* 407089C0
0020 D0F0889C 0985A032 0006376C D7C1E8F1 *}0h..e.....%PAY1* 407089D0
```

```
SCE.U0006376 00000050_4070AA70 Storage Element Descriptor
```

```
0000 00000050 4070B1A0 00000050 407089B0 *...& .....& .i.* 4070AA70
0010 25704570 00000420 2242FE80 00000000 *.....* 4070AA80
0020 D0F0889C 0983E1C0 0006376C D7C1E8F1 *}0h..c.{...%PAY1* 4070AA90
```

```
SCE.U0006376 00000050_4070B1A0 Storage Element Descriptor
```

```
0000 00000050 4070AE90 00000050 4070AA70 *...& .....& ...* 4070B1A0
0010 25700870 00003D00 2242FE80 00000000 *.....* 4070B1B0
0020 D0F0889C 09769DD8 0006376C D7C1E8F1 *}0h...Q...%PAY1* 4070B1C0
```

IPCS LIST example

- Storage check zones are intact. Press PF3 to return to the previous page

```

IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> SCROLL ==> CSR
***** TOP OF DATA *****

LIST 25704570. ASID(X'0627') LENGTH(X'0420') AREA
25704570. E4F0F0F0 F6F3F7F6 40404040 40404040 |U0006376 |
25704580 LENGTH(X'E0')==>All bytes contain X'40', EBCDIC C' ' |
25704660. 40404040 00000000 00004040 40404040 | ..... |
25704670 LENGTH(X'0310')==>All bytes contain X'40', EBCDIC C' ' |
25704980. 40404040 40404040 E4F0F0F0 F6F3F7F6 | U0006376|
***** END OF DATA *****

```

IPCS Find subcommand

- Locates literal values in a dump

```
{ FIND } [relational-operator]
```

```
{ F }
```

```
    [value]
```

```
    [data-desc]
```

```
    [BOUNDARY (bdy, index-range) ]
```

```
    [BREAK | NOBREAK]
```

```
    [FIRST | NEXT | PREVIOUS]
```

```
    [MASK (mask) | NOMASK]
```

IPCS Find subcommand

- relational operators: < | LT | <= | LE | ¬> | NG | = | EQ | >= | GE | ¬< | NL | > | GT | ¬ | NE
- data-desc specifies the data description operand, which consists of at least an address
- BOUNDARY requests that storage is partitioned into strings of *bdy* bytes in length
- BREAK specifies that FIND will stop processing if it cannot retrieve storage from the dump to continue the search. This happens if the required storage was not acquired through a GETMAIN or the required storage is not contained in the dump
- FIRST | LAST | NEXT | PREVIOUS defines where the search is to begin or proceed
- MASK defines a value that is logically ANDed with both operands before performing the comparison
- Example: {FIND} DFHSIT finds a character string of DFHSIT

Other IPCS subcommands

Command	Description
VERBX MTRACE	System log for the job
VERBX TRACE	System trace z/OS
VERBX NUCLMAP	Map the z/OS nucleus
LPAMAP	Map of the LPA and ELPA
SUMM FORMAT	Formats the MVS control blocks for the address space
DROPDUMP (DROPD)	Drops the information stored in the current dump
FINDMOD	Locates a module which has been z/OS loaded
STATUS (ST)	Status command – gives the ‘worksheet’ PSW/REGS
GTFTRACE	Formats GTF trace data
WHERE	Identifies where a particular address is located

Note: In option 6, enter IPCS HELP to get help for any IPCS command

SDUMP options

SDUMP options

The CICS dump exit is unable to successfully format a CICS dump unless these minimum SDUMP, SDATA parameter options are in effect when the dump is written

ALLPSA	prefix storage area for all processors
CSA	common storage area
GRSQ	global resource serialization queues
LPA	link pack area
NUC	non-page-protected areas of the DAT-on nucleus
RGN	private area of CICS address space
SQA	system queue area
SUMDUMP	summary dump function
TRT	GTF, system trace, and master trace data

Displaying SDUMP options

Run the MVS command, DISPLAY DUMP OPTIONS

- Enter `D D,O` to display the SDATA parameters
- Check the MVS console for message IEA611I, IEE711I or IEA911E

```
IEA611I PARTIAL DUMP
ON SYS1.P390.DMP00001
IEE711I SYSTEM DUMP
NOT TAKEN. A CRITICAL
AUXILIARY STORAGE
SHORTAGE EXISTS
```

MVS Console D D,O command

```
COMMAND INPUT ==> /D D,O
RESPONSE=SRV1
IEE857I 09.14.23 DUMP OPTION 118
  SYSABEND- ADD PARMLIB OPTIONS SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM) ,
            PDATA=(SA,REGS,LPA,JPA,PSW,SPLS)
  SYSUDUMP- ADD PARMLIB OPTIONS SDATA=(CB,ERR,SUM) , PDATA=(SA,REGS,
            LPA,JPA,PSW,SPLS)
  SYSMDUMP- ADD PARMLIB OPTIONS (NUC,SQA,LSQA,SWA,TRT,RGN,CSA,SUM)
  SDUMP- ADD OPTIONS (PSA,SQA,LSQA,RGN,TRT,CSA,SUMDUMP,GRSQ,YESDATA) ,
            BUFFERS=00000000K,MAXSPACE=00012288M,
            MSGTIME=99999 MINUTES,MAXSNDSP=015 SECONDS,
            AUXMGMT=OFF,DEFERTND=NO
            SYSFAIL NO STRLIST OPTIONS
  ABDUMP- TIMEENQ=0240 SECONDS
```

Setting SDUMP options

Ensure that the SDUMP options needed to support the CICS dump exit are in force at the time the dump is written

If the options are not set correctly, you can do one of the following:

- Enter MVS change dump command:

```
CD SET,SDUMP=(ALLPSA,SQA,SUMDUMP,NUC,RGN,LPA,TRT,CSA,GRSQ),ADD
```

- update member IEACMDxx in the SYS1.PARMLIB

Note: The SDUMP options are described under *z/OS MVS System Commands > MVS system commands reference > CHNGDUMP command* in the Knowledge Center

MAXSPACE and auxiliary storage

SDUMP dump sizes are increasing

When a storage dump is captured, it must be backed up by auxiliary storage. To determine the amount of page space, you need to consider MAXSPACE (default value 500M)

SDUMP processing from z/OS V1R11 has been enhanced

Available auxiliary storage is now monitored during SDUMP processing to detect excessive use. The MAXSPACE value restricts the virtual storage available to the DUMPSRV address space.

The general guideline is that the ratio between MAXSPACE and auxiliary storage should be 1:3.

For example, if MAXSPACE=5000 M is specified, 15000 M of paging space is required to ensure that there is enough auxiliary storage to back the captured storage dump

```
CD SET,SDUMP,MAXSPACE=xxxxxxxM
```

Slow Dump Process - consider removing the SDATA “GRSQ” option

This captures MVS-related “contention” information and may be unnecessary if your problem scope is limited to a single CICS region. Including “GRSQ” can significantly increase the time required to capture a system dump, depending on your system

MVS Console Dump

MVS Console Dump command

“The DUMP command requests a system dump of virtual storage (SVC dump).

The SVC dump is stored in a direct access data set.

The data set can be either a pre-allocated dump data set named SYS1.DUMPxx, or an automatically allocated dump data set named according to an installation-specified pattern.”

([DUMP Command](#) topic - IBM Knowledge Center)

MVS Console Dump command

```
/DUMP COMM=(DAC001)
*008 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
  R 08 ,JOBNAME=(DACTOR1) ,SDATA=(RGN,ALLNUC ,CSA ,LSQA ,PSA ,
SQA ,SUM ,SWA ,TRT ,COUPLE ,WLM ,LPA) ,END

IEA794I SVC DUMP HAS CAPTURED: 453
DUMPID=002 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=DAC001
```


References

- MVS Interactive Problem Control System (IPCS) Commands

SA23-1382-30

[https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3SA231382/\\$file/ieac500_v2r3.pdf](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3SA231382/$file/ieac500_v2r3.pdf)

- z/OS V2R1.0 MVS Interactive Problem Control System (IPCS) User's Guide

SA23-1384-30

[https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3sa231384/\\$file/ieac600_v2r3.pdf](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3sa231384/$file/ieac600_v2r3.pdf)

- CICS Transaction Server for z/OS V5.3: Operations and Utilities Guide

SC34-7420-00

<http://publibfp.dhe.ibm.com/epubs/pdf/dfha6h00.pdf>

Thank You for Attending!
Please remember to complete your evaluation of this session in the SHARE mobile app.

Introduction to IPCS for CICS People

