Introduction to IPCS

—

John C. Shebey III
IBM z/OS Support

IBM

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- **MVS**
- **OS/390®**
- **z/Architecture®**
- **z/OS®**

* Registered trademarks of IBM Corporation

For a complete list of IBM trademarks, see: http://www.ibm.com/legal/us/en/copytrade.shtml

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Table of Contents

IPCS Overview

What is IPCS?

- IPCS (**I**nteractive **P**roblem **C**ontrol **S**ystem)
  - Interactive tool provided with z/OS to aid in diagnosing software failures
  - Defined as a SUBCOMMAND command processor under TSO
  - Can be invoked from either FOREGROUND or BACKGROUND
  - Provides formatting and analysis support for dumps and traces produced by MVS, other program products, and applications that run on MVS

Output from IPCS

- IPCS produces three types of output
  - Formatted reports that user can view at terminal or print to MVS data set
  - Control block symbols and mappings of storage saved in dump directory
    - User dump directory is VSAM dataset containing information about dumps and traces viewed under IPCS
  - Symptom strings placed in headers of dump data sets
    - Symptom strings identify information about failure (e.g. abend code, failing component, failing routine)

Data Viewable in IPCS

- View Unformatted Dumps
  - SYSMDUMP
  - SVC Dump
  - SADump (Standalone Dump)
- Active Storage
  - Limited storage on active system (Source=ACTIVE)
- GTF/CTRACE Data Written to External Writer
  - IPCS Option 2.7 (ANALYSIS -> TRACES)
- DAE Data
  - IPCS Option 3.5 (UTILITY -> DAE)

Basic Setup Requirements for IPCS

- Define IPCS Dump Directory Data Set - VSAM Cluster
  - Refer to z/OS MVS IPCS User's Guide for more details
- Initialize the Dump Directory Data Set
  - IPCSDDIR command
- Concatenate SYS1.SBLSCLI0 into SYSPROC
  - IPCS REXX execs and CLISTs for dump analysis
- Allocate the FILE(IPCSDDIR) with Dump Directory Data Set
- Allocate the FILE(IPCSPRNT) for print output (Optional)

IPCS Basic Navigation

- Main menu when IPCS dialog invoked from ISPF
- Customizable like other ISPF dialogs
  - Panels shipped by default in SYS1.SBLSPNL0 pointed to by ISPPLIB

```
------------------------ 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
```

IPCS Option 0: DEFAULTS

- Choose which dump or trace data set to initialize
- Set default options that IPCS should use and press ENTER

```
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('ZOS.DEBUG.DUMP1')
Address Space    ==> ASID(X'0025')
Message Routing ==> NOPRINT TERMINAL NOPDS
Message Control ==> CONFIRM VERIFY FLAG(WARNING)
Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL NOALIGN
```

Initializing Dump or Trace

- Dump or trace data set initializes when IPCS formats or analyzes data as a result of:
  - Entering first IPCS dialog option
  - Issuing first IPCS subcommand

```
IKJ56650I TIME-09:59:37 PM. CPU-00:00:00 SERVICE-77283 SESSION-00:25:21 JANUARY 27,2020
BLS18122I Initialization in progress for DSNAME('ZOS.DEBUG.DUMP1')
BLS18124I TITLE=SHR2ESTA DETECTED ABEND S00C4, REASON 00000011
BLS18223I Dump written by z/OS 02.03.00-0 SVC dump - level same as IPCS level
```

- Initialization not automatic from IPCS Option 0 (DEFAULTS)

Dump Initialization: Summary Dump Data

- Summary dump data includes:
  - Storage dumped closest to time of error
  - Storage pointed to by PSW/registers at time of error
- While dump is initializing, message BLS18160D will be displayed

```
BLS18160D May summary dump data be used by dump access?
   Enter Y to use, N to bypass.
```

- Recommend replying ‘Y’ to message BLS18160D

When is Initialization Complete?

- When initialization of dump or trace is complete, you will see '***'

```
 IKJ56650I TIME-09:59:37 PM. CPU-00:00:00 SERVICE-77283 SESSION-00:25:21 JANUARY 27,2020
 BLS18122I Initialization in progress for DSNAME('ZOS.DEBUG.DUMP1')
 BLS18124I TITLE=SHR2ESTA DETECTED ABEND S00C4, REASON 00000011
 BLS18223I Dump written by z/OS 02.03.00-0 SVC dump - level same as IPCS level
 BLS18222I z/Architecture mode system
 BLS18160D May summary dump data be used by dump access?  Enter Y to use, N to bypass.
Y
 BLS18255I Dump Init    Elapsed Time          CPU Time
           Input I/O   00:00:00.346545     00:00:00.006364
           DDIR        00:00:00.022046     00:00:00.010970
 BLS18123I 39,888 blocks, 165,934,080 bytes, in DSNAME('ZOS.DEBUG.DUMP1')
 IKJ56650I TIME-10:11:58 PM. CPU-00:00:00 SERVICE-89234 SESSION-00:37:42 JANUARY 27,2020
 BLS18224I Dump of z/OS 02.03.00-0 - level same as IPCS level
 ***
```

- Press ENTER to proceed

IPCS Option 4: INVENTORY

- Dumps and traces initialized in IPCS Dump Directory

```
 IPCS INVENTORY - JSHEBEY.DDIR ------------------------------------------------
   -----------

AC Dump Source                                                      Status
SD DSNAME('ZOS.DEBUG.DUMP1') . . . . . . . . . . . . . . . . . . . ADDED
   Title=SHR2ESTA DETECTED ABEND S00C4, REASON 00000011
   Psym=RIDS/SHARE2#L RIDS/SHARE2 AB/S00C4 RIDS/SHR2ESTA#R VALU/HBF1FA06C REGS/80166
    PRCS/000000
__ DSNAME('ZOS.DEBUG.DUMP4') . . . . . . . . . . . . . . . . . . . ADDED
   Title=HUNG SYSTEM
   No symptoms
```

The following line commands (AC column of data source) are commonly used:

- SD - Establishes the source as both the local and global IPCS default
- DD - Deletes description of the source and, optionally, the source data set
- LD - Lists dump description with dumped storage summary

IPCS Option 6: COMMAND

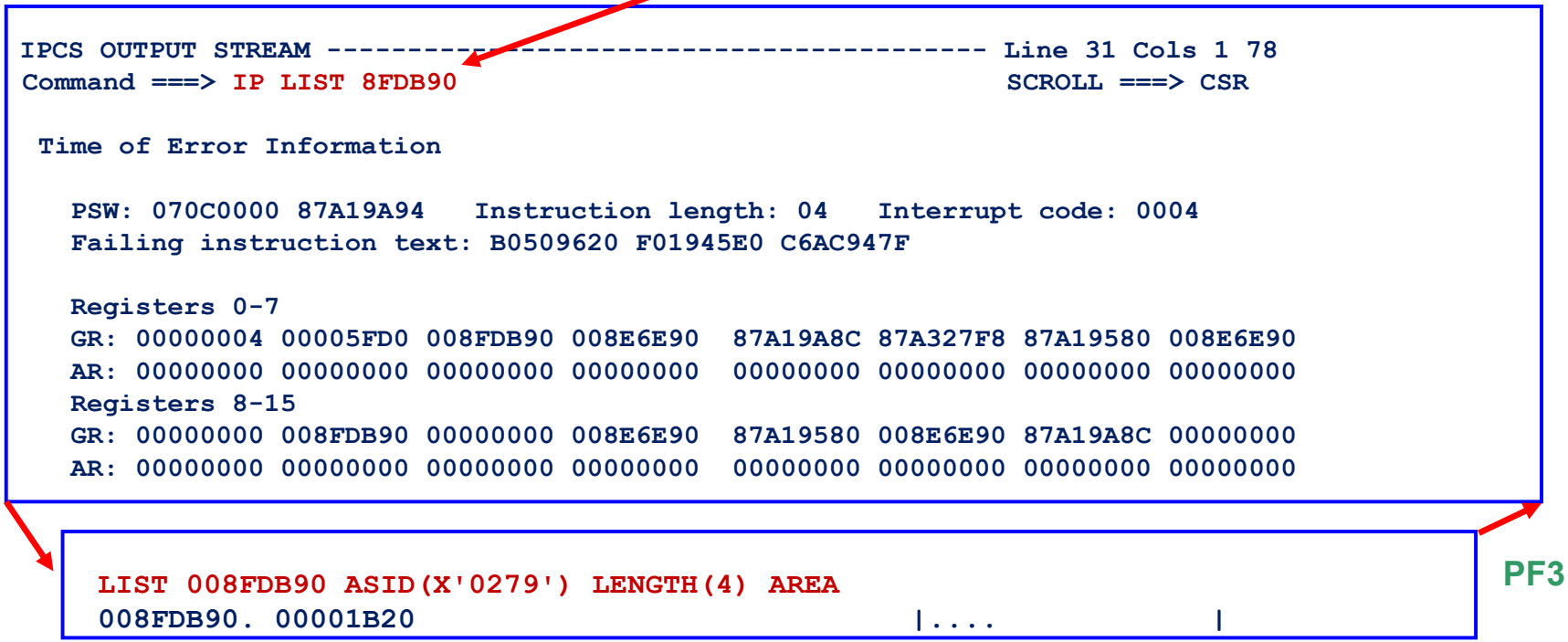
- Lists many of the available IPCS subcommands

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

===>  ST FAILDATA

---------------------- IPCS Subcommands and Abbreviations --------------------
ADDDUMP             | 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
ASMCHECK, ASMK      | EQUATE,   EQU, EQ  | LITERAL            | SETDEF,    SETD
CBFORMAT, CBF       | FIND,     F        | LPAMAP             | STACK
CBSTAT              | FINDMOD,  FMOD     | MERGE              | STATUS,    ST
```

- IPCS subcommands can also be entered from any IPCS command line with prefix of IP

Stacking IPCS Subcommands

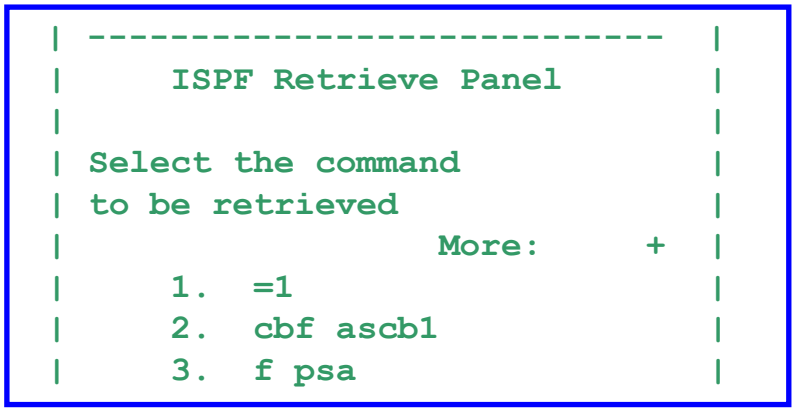
Issue subcommand on Command line prefixed by IP



PF3

Retrieving IPCS Commands

- RETP (enter on command line without IP prefix)
  - Retrieve panel with up to last 25 commands issued



- KEYS (enter on command line without IP prefix) – setup PF keys
  - Next to PF12, type RETRIEVE
    - PF12 key will then cycle through most recent commands.

Approach to Dump Debug

Type of Dump: STATUS SYSTEM

- Use this command to verify type of dump
- Program producing dump can be:
  - SVCDUMP, SLIPDUMP, SYSMDUMP, SADUMP

```
SYSTEM STATUS:
  Nucleus member name: IEANUC01
   I/O configuration data:
     IODF data set name: SYS9.IODF52
     IODF configuration ID: CONFIG00
     EDT ID: 00
Sysplex name: LOCAL
  TIME OF DAY CLOCK: D73FEE59 5A789400  12/30/2019 16:48:27.916169 local
  TIME OF DAY CLOCK: D74023FE 43789400  12/30/2019 20:48:27.916169 GMT
  Program Producing Dump: SVCDUMP
  Program Requesting Dump: SVCDUMP
  Incident token: LOCAL    SP7I     12/30/2019 20:48:27.809774 GMT
```

LIST TITLE

- Use to get an idea of what the dump represents
- For SVC dump:
  - Recovery dumps typically have a COMPID= and other recovery information, depending on the recovery routine that requests the dump
  - Console dumps have a title of whatever the user puts in COMM= as the dump title
  - SVC dumps taken by SLIP have SLIP trap id in title
  - Any program can issue the SDUMP macro and generate an SVC dump with a title of its choosing
- Example:

```
TITLE
LIST 00. LITERAL LENGTH(X'58') CHARACTER
00000000 | COMPON=BPX,COMPID=SCPX1,ISSUER=BPXMIPCE,MODULE=BPXPRSRB+????,ABE |
00000040 | ND=S00C6,REASON=00000006                                        |
```

Use of Dump Title

- May provide component and job(s) involved, as well as indication of problem for which dump was taken
  - Dump taken from SLIP trap with ID=S0C1

```
TITLE
LIST 00. LITERAL LENGTH(X'11') CHARACTER
00000000 | SLIP DUMP ID=S0C1 |
```

  - Console dump taken for hung BANKER job

```
TITLE
LIST 00000000 LITERAL LENGTH(19) CHARACTER
00000000 | JOB BANKER IS HUNG |
```

LIST SLIPTRAP

- Use if you have reason to believe that a SLIP trap was used to produce the dump
- If a SLIP trap was used to produce dump, you will see output similar to the following:

```
SLIPTRAP
LIST 00. LITERAL LENGTH(X'54') CHARACTER
00000000 | SLIP SET,ID=S0C1,C=0C1,AL=(H,P,S),A=SVCD,SDATA=(ALLNUC,CSA,SQA,L |
00000040 | PA,PSA,RGN,TRT,GRSQ)                                             |
```

Check z/OS Release and CPU: CBF CVT

- Can be used to verify system release level on which dump was taken, and also to verify CPU model
- Example:

```
HBB77B0 = z/OS 2.3
HBB77C0 = z/OS 2.4
HBB77D0 = z/OS 2.5
```

```
CVT: 00FDA0E0
   -0028  PRODN.... SP7.2.2    PRODI.... HBB77B0
          MDL...... 3906
   -0004  RELNO.... 038
```

- In above example, dump was taken of z/OS 2.3 system running on a 3906 (z14) CPU

STATUS WORKSHEET

- Provides environmental information about dump and system
  - Dump title
  - Original dump data set name
    - For SVC dump
  - Date and time dump was taken
  - System name
- Displays active CPUs on system at time of dump
  - For a SADump, displays PSW address running on each CPU
    - Useful for identifying possible loops
      - Check for same/similar PSW address on multiple CPUs
- Shows which SDATA areas are included in dump

ST WORKSHEET: Example

```
                                MVS Diagnostic Worksheet

Dump Title: WAIT 084 SYSTEM CRASH STANDALONE DUMP


CPU Model 2964 Version FF Serial no. 111111 Address 01
Date: 10/27/2019     Time: 11:22:31.983229 Local

Wait State Message Issued at 11:21:20 on Day 300 of 2019:
 IEA020W AN FRR STACK POINTER FOR CPU 01 IS DAMAGED, THE ERROR MASK IS
 1111111111
 1111111111.


 SYSTEM RELATED DATA

CVT  SNAME (154) SY1        VERID (-18)

     CUCB   (64) 00FD5B00   PVTP  (164) 00FFB208    GDA  (230) 02109278

     RTMCT (23C) 00F50B20   ASMVT (2C0) 00FD7F98    RCEP (490) 01906D18
```

ST WORKSHEET: Example (cont)

```
CSD

     Alive Mask: C0000000 00000000 00000000 00000000 #CPUs:     2
     CP    Mask: 00000000 00000000 00000000 00000000 #CPUs:     2
     zAAP  Mask: 00000000 00000000 00000000 00000000 #CPUs:     0
     zIIP  Mask: 00000000 00000000 00000000 00000000 #CPUs:     0

PROCESSOR RELATED DATA

 NAME                OFFSET   |  CPU 01     CPU 02
--------------------------+-----------------------------------------
 PSW at time of dump      |  00020000   00020000
                          |  80000000   80000000
                          |  00000000   00000000
                          |  00004084   00004084
 CR0 Interrupt mask       |  00000000   00000000
 CR6 I/O class mask       |  00          00
------ LCCA ---------------+-----------------------------------------
 IHR1   Recursion     208  |  00          00
 SPN1/2 Spin          20C  |  0000        0000
 CPUS   CPU WSAVT     218  |  00F8E000    00F93D80
 DSF1/2 Dispatcher    21C  |  0000        0000
 CRFL   ACR/LK flgs   2B4  |  00000000    00000000
------ PSA ---------------+-----------------------------------------
 TOLD   Curr TCB      21C  |  008FF2A0    00000000
 AOLD   Curr ASCB     224  |  00FA2380    00FD3900
```

ST WORKSHEET: Example (cont)

```
==> FLAGS SET IN SDUFLAG0:
 HDR/HDRADR specified.
 ECB specified.
 Set system non-dispatchable while dumping global storage.

==> FLAGS SET IN SDUFLAG1:
 SVC dump request.
 48+ byte parameter list.

==> FLAGS SET IN SDUSDATA:
 Dump all PSAs.
 Dump the nucleus.
 Dump SQA.
 Dump LSQA.
 Dump rgn-private area.
 Dump LPA mod. for rgn.
 Dump trace data.
 Dump CSA.
 Dump SWA.
 Dump summary dump data.
 Dump all nucleus.
```

```
SDATA=(ALLPSA,SQA,LSQA,
RGN,LPA,TRT,CSA,SWA,
SUM,ALLNUC)
```

List ASIDs dumped: CBF RTCT

- Display which ASIDs are included in SVC dump
  - F SDAS

```
RTCT: 00F59B20
   +0000  NAME..... RTCT      SAP...... 2FD0BE00  SUP...... 00100000  . . .
   +0014  MECB..... 808EC148  FASB..... 00000000  NAS...... 00000002  . . .
   +0028  SDDC..... 0001      MTCT..... 0000      DSV...... 00B96D80  . . .
   +0038  ADG1..... 00D3A530  ADG2..... 00D3A540  ADG3..... 00D3A4AE  . . .
   +004C  TABG..... 80C72190  TABQ..... 80C721AE  TABR..... 80C721F0  . . .
   +0060  DIRS..... 024D1370  SDAT..... 02603D20  SMOD..... 02365000  . . .
   +0074  RPAR..... 0183E118  BPXP..... 00000000  TABO..... 00BC47A8  . . .
   +009C  SDPL..... 023F2E28  FMT...... 00000000  MLCK..... 00000001  . . .

   ...

            ASTB

                 SDAS   SDF4   SDF5
                 ----   ----   ----
            001  000E   80     00
            002  0020   80     00
            003  0000   00     00
```

ASIDs dumped

Associate ASID with JOBNAME: SELECT

- Subcommand provides ASID/JOBNAME translation for address spaces
- Options:
  - ASID(X'nn')
  - JOBNAME(jjjjjjjj)
  - ALL
    - Determine all active address spaces on system
- Example: SELECT ASID(x'E',x'20')

```
ASID JOBNAME  ASCBADDR  SELECTION CRITERIA
---- -------- --------  ------------------
000E OMVS     00F57300  ASID
0020 ZFS      00FBF000  ASID
```

SVC Dump: Partial or Complete

- Phases of SVC dump processing:
  - Dump requested
  - Capture Phase
  - Write Phase

- Message IEA611I/IEA911E indicates whether dump is complete or partial
  - Message text includes SDRSN code bits indicating the reason why dump is partial
  - Covers all phases of dump processing

- The partial SDUMP reason codes (SDRSN) are available in the dump (only up through Data Capture) via the IPCS LIST subcommand:
  - **VERBX IEAVTSFS**

Example:

```
Capture phase partial dump reason codes (IHASDRSN):
  Are all zeros.
```

Dump complete if “zeros”.

Which CPUs are Active?
SYSTRACE STATUS TIME(LOCAL)

- Use to determine which CPUs are active at time of dump, and which (if any) are zIIP or zAAP processors
- Also indicates time that trace entries are available from all processors
- Example:

```
TRACE data reporting from all CPUs starts at 12/04/2019 09:41:59.262116 (CPU 0000)

------+------+-----------------------------+--------------------------
CPU   | Type | SYSTRACE First Local Time   | SYSTRACE Last Local Time
------+------+-----------------------------+--------------------------
0000  | CP   |  12/04/2019 09:41:59.262116 | 12/04/2019 09:42:00.390857
0001  | zAAP |  12/04/2019 09:33:42.270150 | 12/04/2019 09:42:00.389228
0002  | zIIP |  12/04/2019 09:33:42.277965 | 12/04/2019 09:42:00.389263
```

Browsing Storage

Viewing Storage In a Dump

- Quick Method:
  - IP LIST aaaaaaaa LEN(X‘nn') ASID(X‘nn’)
    - Issued on any command line to see storage
    - PF3 allows you to get back to previous display

- Scrollable Method:
  - ‘=1’ on any command line in IPCS takes you to the BROWSE menu (IPCS Option 1) that can be used to browse storage
    - Example on next pages will allow us to verify PSA eye catcher is at virtual address x'200'

IPCS Option 1: BROWSE

- Browsing Storage: IPCS Option 1 (=1 on any IPCS command line) brings up the following panel. Hit <ENTER> to get to POINTER STACK on next slide.

```
------------------------ IPCS - ENTRY PANEL  --------------------------------
 CURRENT DEFAULTS:
  Source ==> DSNAME('MVSSPT.S2822.DUMP1A')
  Address space ==> ASID(X'0065')

 OVERRIDE DEFAULTS:                          (defaults used for blank fields)
  Source ==> DSNAME('MVSSPT.S2822.DUMP1A')
  Address space ==>
  Password      ==>

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

IPCS Option 1: BROWSE Example

```
DSNAME('MVSSPT.S2822.DUMP1A') POINTERS
---------------------------------------
ASID(X'0065') is the default address space
PTR    Address    Address space                Data type
s0001  00000200   ASID(X'0065')                AREA
       Remarks:
**************** END OF POINTER STACK ************************
```

PF7 – scroll up
PF8 – scroll down

Hit <ENTER> to browse storage at address x'00000200':

```
ASID(X'0065') ADDRESS(0200.) STORAGE ------------------------------------------
Command ===> L 07E00F04. ASID(x'65')                        SCROLL ===> CSR
00000200 ! D7E2C140   00010041   00F44008 ? 030A6008   | PSA .....4 ...-. |
00000210   00F83000   030E5000 % 005EC120   005EC120   | .8....&..;A..;A. |
00000220   00FC5A80   00F9B400   00000000   00000000   | ..!..9.......... |
00000230   00000000   00000000   00000000   0000000D   | ................ |
```

Indirect addressing: ? - 31-Bit addr, % - 24-Bit addr, ! - 64-Bit addr

Browsing Storage – LOCATE Example

- Suppose we are currently at PSA storage location x'200', and we want to display storage at PSA location x'220'

```
ASID(X'0065') ADDRESS(0200.) STORAGE ----------------------------------------
Command ===> EQ PSAID 200.                                  SCROLL ===> CSR
00000200   D7E2C140   00010041   00F44008   030A6008   | PSA .....4 ...-. |
00000210   00F83000   030E5000   005EC120   005EC120   | .8....&..;A..;A. |
00000220   00FC5A80   00F9B400   00000000   00000000   | ..!..9.......... |
00000230   00000000   00000000   00000000   0000000D   | ................ |
```

L 0220 → L X+20 → L PSAID+20

```
ASID(X'0065') ADDRESS(0200.) STORAGE ----------------------------------
 Command ===>
 00000220   00FC5A80   00F9B400   00000000   00000000   | ..!..9.......... |
```

X - indicates current storage location
EQ - equates symbol to address

Error Information

Error Information in SVC Dump

- Recovery routine requests dump
  - ST FAILDATA
    - PSW and registers at time of error (from SDWA)
- Dump requested from operator console
  - No SDWA = No ST FAILDATA
  - ST REGS
    - PSW and registers at time console dump was requested
      - Normally not very useful
- SLIP generated dump
  - No SDWA = No ST FAILDATA
  - ST REGS
    - PSW and registers at time SLIP matched

STATUS FAILDATA

- Displays PSW/registers at time of dump
  - General Purpose Registers
  - Access Registers
- Provides diagnostic information from SDWA
  - Instruction length
  - Interrupt code
  - Translation exception address (TEA)
  - Breaking event address (BEAR)

ST FAILDATA: Example

```
SEARCH ARGUMENT ABSTRACT

  RIDS/SHARE2#L RIDS/SHARE2 AB/S00C4 PRCS/00000011 REGS/80166 RIDS/SHR2ESTA#R

  Symptom              Description
  -------              -----------
  RIDS/SHARE2#L        Load module name: SHARE2
  RIDS/SHARE2          Csect name: SHARE2
  AB/S00C4             System abend code: 00C4
  PRCS/00000011        Abend reason code: 00000011
  REGS/80166           Register/PSW difference for R08:-0166
  RIDS/SHR2ESTA#R      Recovery routine csect name: SHR2ESTA

SERVICEABILITY INFORMATION NOT PROVIDED BY THE RECOVERY ROUTINE
  Program id
  Recovery Routine Label
  Date Assembled
  Module Level
  Subfunction
```

ST FAILDATA: Example (cont)

```
Time of Error Information

 PSW: 07042000 80000000 00000000 0890011A
 Instruction length: 02   Interrupt code: 0011
 Failing instruction text: 100DBFF8 802E0E0E BF1FA06C
 Translation exception address: 00000000_07A1D400

 Breaking event address: 00000000 0184F6B0
 AR/GR 0-1    00000000/00000000_07A1D000  00000000/00000000_FFFEEFFF
 AR/GR 2-3    FFFFFFFF/FFFFFFFF_08900166  FFFFFFFF/FFFFFFFF_006DBD6C
 AR/GR 4-5    FFFFFFFF/FFFFFFFF_006DBD48  FFFFFFFF/FFFFFFFF_006F8588
 AR/GR 6-7    FFFFFFFF/FFFFFFFF_006CCFC8  FFFFFFFF/FFFFFFFF_00F97280
 AR/GR 8-9    FFFFFFFF/FFFFFFFF_08900280  FFFFFFFF/FFFFFFFF_006F8190
 AR/GR 10-11  FFFFFFFF/FFFFFFFF_7F540E10  FFFFFFFF/FFFFFFFF_00000000
 AR/GR 12-13  FFFFFFFF/FFFFFFFF_0593C388  FFFFFFFF/00000000_7F540E18
 AR/GR 14-15  00000000/00000000_089013AD  00000C01/00000000_EE000000

 Home ASID: 0025    Primary ASID: 0025    Secondary ASID: 0025

 This Task's ASID/TCB: 0025/006FE9A8
```

STATUS REGS

- Displays PSW/registers at time of dump
  - General Purpose Registers
  - Access Registers
  - Control Registers
- Provides more information about the PSW
  - AMODE, Key, Enabled/Disabled
- However, failing instruction text, instruction length, and interrupt code are not provided (as in ST FAILDATA)
  - No SDWA

ST REGS: Example

```
CPU STATUS:
PSW=07042000 80000000 00000000 0890011A
     (Running in PRIMARY, key 0, AMODE 31, DAT ON, SUPERVISOR STATE)
     Disabled for PER
    ASID(X'0025') 0890011A. AREA(Subpool252Key00)+011A IN EXTENDED PRIVATE
    ASID(X'0025') 0890011A. SHARE2+011A IN EXTENDED PRIVATE
   ASCB37 at F97280, JOB(TESTJOB), for the home ASID
   ASXB37 at 6FD000 for the home ASID. TCB261G at 9A5E88 for the home ASID
   HOME ASID: 0025 PRIMARY ASID: 0025 SECONDARY ASID: 0025

   General purpose register values
      0-1  00000000_07A1D000  00000000_FFFEEFFF
      2-3  FFFFFFFF_08900166  FFFFFFFF_006DBD6C
      4-5  FFFFFFFF_006DBD48  FFFFFFFF_006F8588
      6-7  FFFFFFFF_006CCFC8  FFFFFFFF_00F97280
      8-9  FFFFFFFF_08900280  FFFFFFFF_006F8190
     10-11 FFFFFFFF_7F540E10  FFFFFFFF_00000000
     12-13 FFFFFFFF_0593C388  00000000_7F540E18
     14-15 00000000_089013AD  00000000_EE000000
```

Where Does Failing PSW Point? WHERE

- Use to map address to specific offset within CSECT or Load Module (collection of CSECTs)
  - WHERE maps local storage address to offset within private loadmod
  - WHERE maps global storage address as follows:
    - **Nucleus:** Maps address to offset within CSECT
    - **LPA:** Maps address to offset within LPA Loadmod

- Example: WHERE 1BE5D800 ASID(X'40')

```
*********************************************************************************
    ASID(X'0040') 1BE5D800. EAGRTALT+2780 IN EXTENDED PRIVATE
    ASID(X'0040') 1BE5D800. AREA(Subpool252Key00)+2800 IN EXTENDED PRIVATE
*********************************************************************************
```

Recent Events

System Trace History: SYSTRACE

- Displays System Trace activity at time of dump, including:
  - Dispatches, interrupts, and recovery actions
- Options:
  - IP SYSTRACE ASID(x'nn') TCB(x'tttttt') TIME(LOCAL)
    - Formats only trace records associated with the requested ASID/TCB
  - IP SYSTRACE ALL TIME(LOCAL)
    - Formats trace records for all active address spaces in the system
- Refer to z/OS MVS Diagnosis: Tools and Service Aids manual, Chapter 8, for more details on System Trace

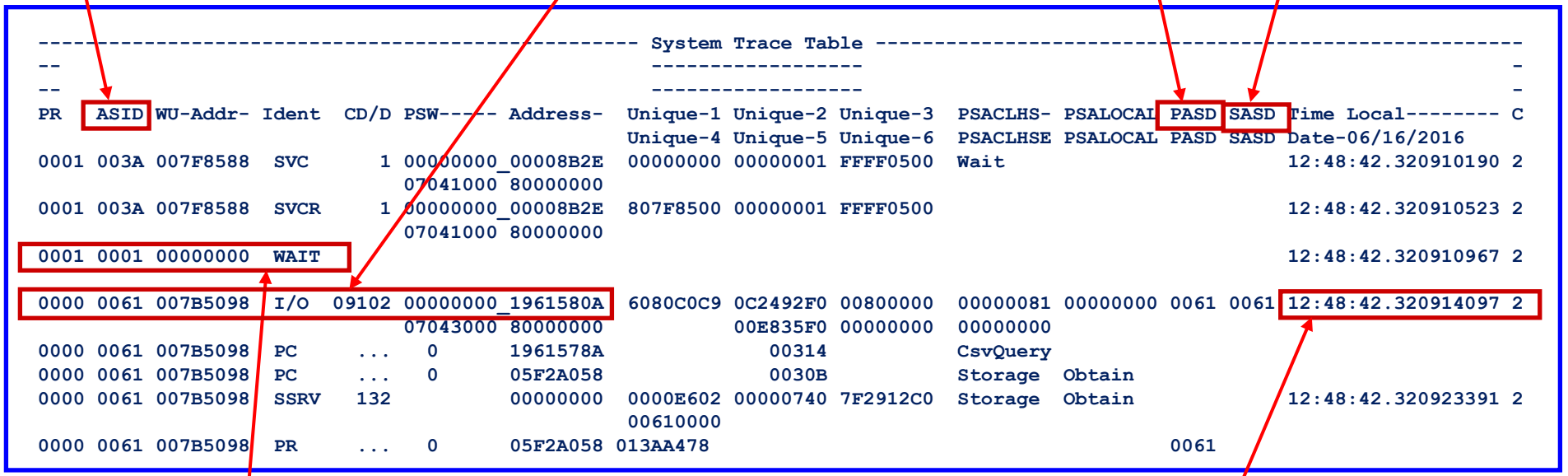
SYSTRACE: Example

HOME ASID

I/O interrupt on CPU 0 while TCB(x'7B5098') in ASID(x'61') was running. Does not mean that ASID(x'61') was doing I/O, but that's just what happened to be running at the time the I/O completion occurred.

PRIMARY ASID

SECONDARY ASID



```
------------------------------------------------ System Trace Table ------------------------------------------------
--                                               ------------------                                                -
--                                               ------------------                                                -
PR   ASID WU-Addr- Ident  CD/D PSW----- Address-  Unique-1 Unique-2 Unique-3  PSACLHS- PSALOCAL PASD SASD Time Local-------- C
                                                 Unique-4 Unique-5 Unique-6  PSACLHSE PSALOCAL PASD SASD Date-06/16/2016
0001 003A 007F8588  SVC      1 00000000_00008B2E 00000000 00000001 FFFF0500  Wait                        12:48:42.320910190 2
                               07041000 80000000
0001 003A 007F8588  SVCR     1 00000000_00008B2E 807F8500 00000001 FFFF0500                              12:48:42.320910523 2
                               07041000 80000000
0001 0001 00000000  WAIT                                                                                 12:48:42.320910967 2

0000 0061 007B5098  I/O  09102 00000000_1961580A 6080C0C9 0C2492F0 00800000  00000081 00000000 0061 0061 12:48:42.320914097 2
                               07043000 80000000          00E835F0 00000000  00000000
0000 0061 007B5098  PC      ...   0      1961578A             00314           CsvQuery
0000 0061 007B5098  PC      ...   0      05F2A058             0030B           Storage  Obtain
0000 0061 007B5098  SSRV    132          00000000  0000E602 00000740 7F2912C0  Storage  Obtain             12:48:42.320923391 2
                                                   00610000
0000 0061 007B5098  PR      ...   0      05F2A058 013AA478                                  0061
```

Processor no work WAIT on CPU 1.

Timestamp from TIME(LOCAL)

Message History: VERBX MTRACE

- Provides a snapshot of what is taking place in SYSLOG just before the dump
- Useful to see if a job was started, a message was issued, or a command was issued just prior to the problem
- May see messages on delayed issue queues that are not shown in SYSLOG (e.g., waitstate messages)

- Refer to z/OS MVS Diagnosis: Tools and Service Aids manual, Chapter 9, for more details on Master Trace

VERBX MTRACE: Example

```
*** NIP MESSAGES ON THE DELAYED ISSUE QUEUE ***

THERE ARE NO NIP MESSAGES ON THE DELAYED ISSUE QUEUE.

*** BRANCH-ENTRY MESSAGES ON THE DELAYED ISSUE QUEUE ***

   WQE ADDRESS  DATE    TIME      MESSAGE TEXT

      02518F08  2019300 06:12:05  *IEA020W AN FRR STACK POINTER FOR CPU 01 IS DAMAGED, THE
                                   ERROR MASK IS 11111111111111


*** MASTER TRACE TABLE ***

TAG  IMM DATA  |-------------------------- MESSAGE DATA --------------------------------|
0001 00000037  N 0200000 SY1      19300 06:12:04.80 STC00043 00000000  $HASP100 DESTRUCT ON STCINRDR
0001 00000037  N 0020000 SY1      19300 06:12:04.95 STC00043 00000200  IEF695I START DESTRUCT WITH
JOBNAME DESTRUCT IS ASSIGNED TO USER ++++++++
0001 00000037  N 4000000 SY1      19300 06:12:04.95 STC00043 00000000  $HASP373 DESTRUCT STARTED
```

Error History: VERBX LOGDATA

- Provides history of ABENDs leading up to this dump
  - Most recent ABEND at the bottom of the output
- Useful elements:
  - ERRORID: contains sequence number, ASID and time of error
  - TIME OF ERROR INFORMATION: provides PSW and REGs
  - RECOVERY ROUTINE ACTION: indicates if dump was requested
- F 'SOFTWARE EDIT'
  - Scroll through SYMPTOM RECORDs and SOFTWARE RECORDs

VERBX LOGDATA: Example

```
TYPE:   SOFTWARE RECORD       REPORT:   SOFTWARE EDIT REPORT            DAY.YEAR
        (PROGRAM INTERRUPT)                                 REPORT DATE: 332.19
FORMATTED BY: IEAVTFDE  HBB77B0                             ERROR DATE:  320.19
                              MODEL:    4381                            HH:MM:SS.TH
                              SERIAL:   31F167                    TIME:  08:43:28.66

JOBNAME: PAYROLL     SYSTEM NAME: SP7I
ERRORID: SEQ=00011   CPU=0000   ASID=0025   TIME=08:43:28.3
```

VERBX LOGDATA: Example (cont)

```
SEARCH ARGUMENT ABSTRACT

  RIDS/SHARE2#L RIDS/SHARE2 AB/S00C4 PRCS/00000004 REGS/A1F0E
RIDS/SHR2ESTA#R

  SYMPTOM              DESCRIPTION
  -------              -----------
  RIDS/SHARE2#L        LOAD MODULE NAME: BANKMOD
  RIDS/SHARE2          CSECT NAME: BANKER2
  AB/S00C4             SYSTEM ABEND CODE: 00C4
  PRCS/00000004        ABEND REASON CODE: 00000004
  REGS/A1F0E           REGISTER/PSW DIFFERENCE FOR R0A:-1F0E
  RIDS/BANKESTA#R      RECOVERY ROUTINE CSECT NAME: BANKESTA
```

VERBX LOGDATA: Example (cont)

```
TIME OF ERROR INFORMATION

  PSW: 070D1000 00007848   INSTRUCTION LENGTH: 02   INTERRUPT CODE: 0001
   FAILING INSTRUCTION TEXT: B06C1311 0A01 4100 00061B11

   REGISTERS 0-7
   GR: 00000001 FFFF93FC 00005FF8 00006EB0  008F2848 008FDE28 008C5FF8 FD000000
   AR: 008FB01F 00000000 00000000 00000000  00000000 00000000 00000000 00000000
   REGISTERS 8-15
   GR: 008FD214 00006C78 008F35D8 00006B98  4000771E 00006B98 00006C0C 808FD040
   AR: 00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000

   HOME ASID: 00D4     PRIMARY ASID: 00D4     SECONDARY ASID: 00D4
   PKM: 8040           AX: 0000               EAX: 0000

   THIS TASK'S ASID/TCB: 0025/006F8240
```

VERBX LOGDATA: Example (cont)

```
RECOVERY ENVIRONMENT

  RECOVERY ROUTINE TYPE: ESTAE RECOVERY ROUTINE
   RECOVERY ROUTINE ENTRY POINT: 08900112
   USER REQUESTED NO I/O PROCESSING.

RECOVERY ROUTINE ACTION

  THE RECOVERY ROUTINE REQUESTED THAT TERMINATION PROCESSING CONTINUE.
  THE REQUESTED SVC DUMP WAS SUCCESSFULLY STARTED.
   NO LOCKS WERE REQUESTED TO BE FREED.
```

Summary

- IPCS is a valuable tool for reviewing unformatted dumps and trace data
- Commands can be entered manually from a command line or via panels
- The type of dump governs where error information can be found
- There are multiple ways to view storage in a dump
- The most recent error history and message history can be displayed in a dump

Appendix

System Trace Common Entries

- Entries indicating Dispatch of work
  - DSP – TCB Dispatch
  - SRB – Initial SRB Dispatch
  - SSRB – Suspended SRB Dispatch
  - WAIT – Dummy (No-work) Wait Dispatch
- Entries indicating execution of Cross Memory instructions
  - PC – Program Call
  - PR – Program Return
  - PT – Program Transfer
  - SSAR – Set Secondary Address Space Number
- Entries indicating an I/O operation has been performed
  - SSCH – Start Subchannel
  - MSCH – Modify Subchannel
  - HSCH – Halt Subchannel
  - RSCH – Resume Subchannel

System Trace Common Entries (cont)

- Entries indicating an Interrupt has occurred
  - SVC – SVC interrupt (or System Service entered via SVC)
  - I/O – I/O interrupt
  - CLKC, EMS, EXT, CALL, SS – External interrupt
  - PGM – Program Check Interrupt
  - MCH – Machine Check Interrupt
  - RST – Restart Interrupt
- Entries indicating an error has been encountered
  - RCVY – RTM has been entered
  - SVCE – SVC Error
- Miscellaneous entries
  - SVCR – SVC Return
  - SSRV – System Service entered via PC or Branch
  - SUSP – Suspension due to lock not available

Thank you

John C. Shebey III
IBM z/OS Support
—
jshebey@us.ibm.com
+1-845-435-7826
ibm.com