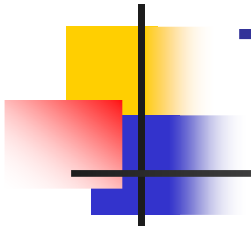
Introduction to IPCS for Application Programmers

Thomas Petrolino
IBM Poughkeepsie
tapetro@us.ibm.com



Trademarks

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

Language Environment

z/OS

CICS

* Registered trademarks of IBM Corporation

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.

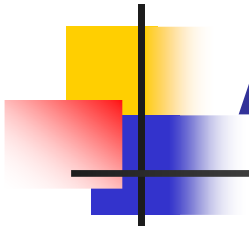
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations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

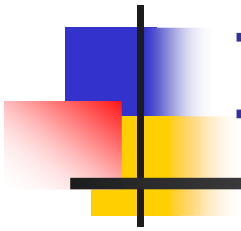
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products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

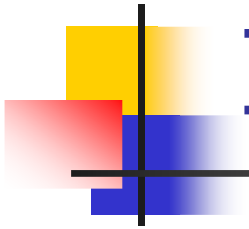


Agenda

- IPCS Overview
- Capturing a Dump
- A Guided Tour
  - Initializing A Dump
  - Status
  - Browsing Storage
  - Address Spaces and Tasks
  - Other Interesting Commands
- Sources of Additional Information

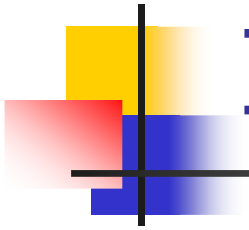


IPCS Overview



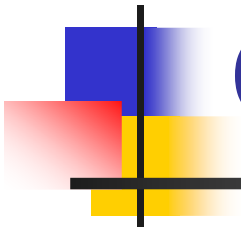
IPCS Overview

- Interactive Problem Control System (IPCS)
  - Formatting and analysis support for
    - Unformatted dumps (SVC, Console, SYSMDUMP, transaction, stand-alone)
    - Traces (GTF, component, master console, system)
  - Several different interfaces
    - Line / Batch / Dialog mode
      - Most popular is full screen (dialog) mode run under ISPF

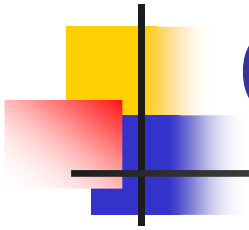


IPCS Overview…

- Advantages over Formatted Dump
  - Provides a more complete picture of the problem
  - Powerful commands and formatters available for dump analysis
- Disadvantages
  - Additional skills required for analysis
  - Application programmers may not have access to system dumps and/or IPCS

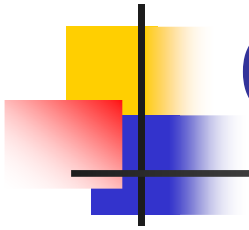


Capturing a Dump



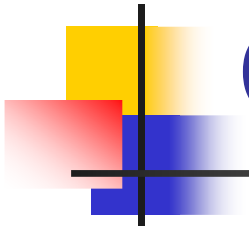
Capturing a Dump

- System Mechanisms
  - SLIP, SVC, Console Dumps
  - Not usually available to Application Programmers
- Language Environment Mechanisms
  - TERMTHDACT Run-time Option is used to request a dump for an unhandled condition of severity 2 or greater
    - Suboption UADUMP/UATRACE/UAONLY
  - Application can use CEE3ABD callable service to request a dump while terminating
    - Type of dump controlled using TERMTHDACT setting



Capturing a Dump…

- TERMTHDACT Run-time Option not Sufficient
  - Application must also either:
    - Allocate a SYSMDUMP DD
      - SYSMDUMP DD DSN=<dump name>, SPACE=(CYL,(200,200),RLSE),DISP=(NEW,DELETE, CATLG),DCB=(RECFM=FBS,DSORG=PS,LRECL=4160, BLKSIZE=24960),UNIT=SYSDA
      - Not always convenient to add to JCL or to execution environment
  - Or, specify the DYNDUMP Run-time Option…



Capturing a Dump…

- DYNDUMP run-time option
  - DYNDUMP(hlq,U4039-ABEND,U40xx-ABEND)
    - hlq
      - *USERID or *USERID.hlq
      - *TSOPREFIX or *TSOPRE
        - (also *TSOPREFIX.hlq or *TSOPRE.hlq)
      - Up to 26 characters of an MVS data set name

Capturing a Dump…

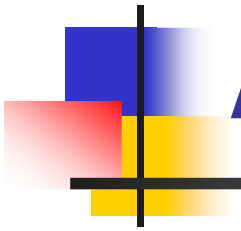
- DYNDUMP run-time option
  - DYNDUMP(hlq,U4039-ABEND,U40xx-ABEND)
    - U4039-ABEND
      - NODYNAMIC (default)
        - DYNDUMP turned off for U4039 ABENDs
      - DYNAMIC
        - DYNDUMP active for U4039 ABENDs if no SYSMDUMP, SYSUDUMP or SYSABEND DD.
      - FORCE
        - DYNDUMP active for U4039 ABENDs even with above DDs allocated
      - BOTH
        - DYNDUMP plus SYSMDUMP/SYSUDUMP/SYSABEND

Capturing a Dump…

- DYNDUMP run-time option
  - DYNDUMP(hlq,U4039-ABEND,U40xx-ABEND)
    - U40xx-ABEND
      - TDUMP (Default)
        - DYNDUMP is active for all U40xx ABENDs (other than U4039) which request a dump.
      - NoTDUMP
        - DYNDUMP is not active for U40xx ABENDs

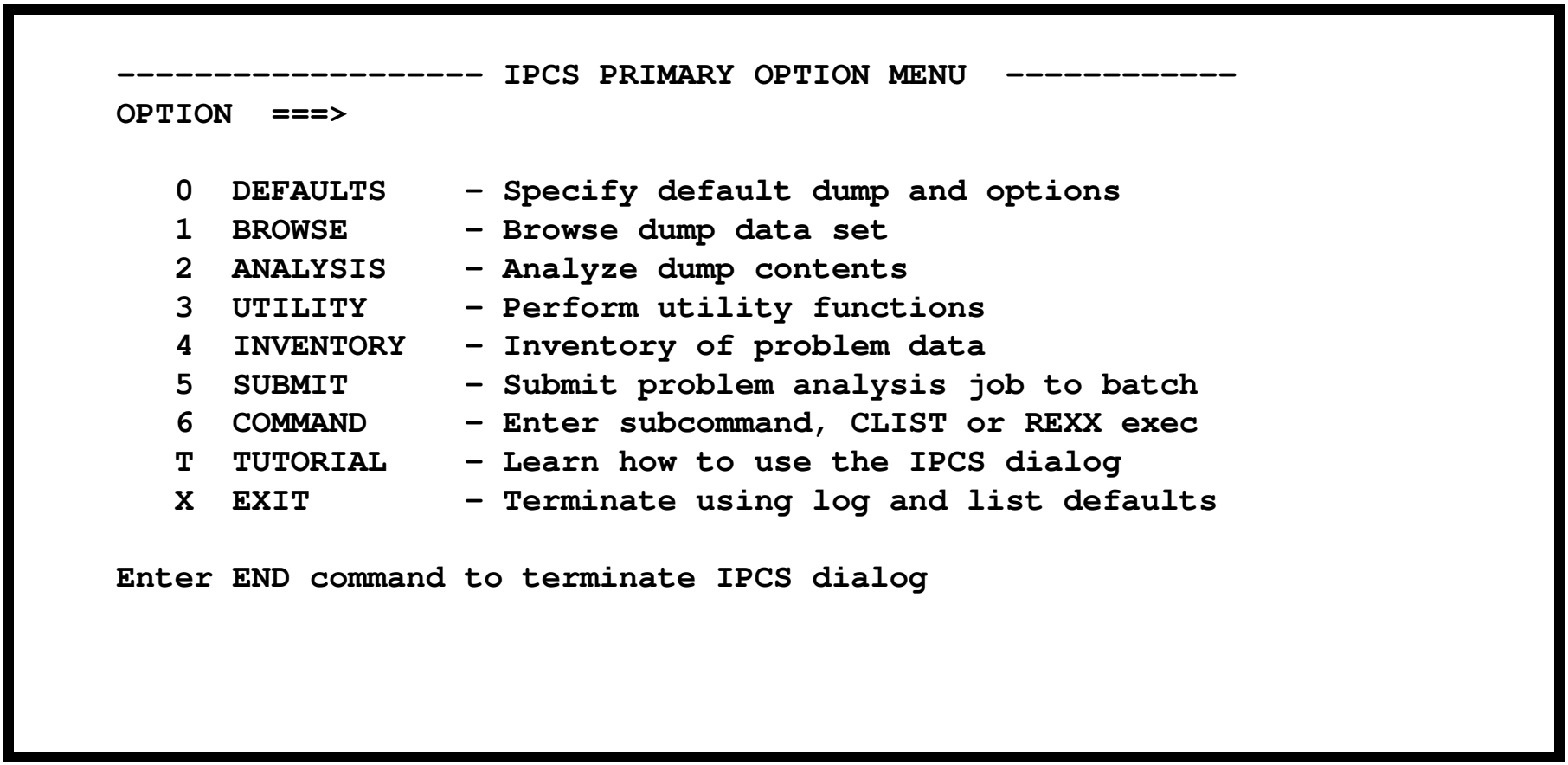
Capturing a Dump…

```
+CEE3798I ATTEMPTING TO TAKE A DUMP FOR ABEND U4039 TO
  DATA SET: PETRO.D194.T0904332.INTIPCS
IGD101I SMS ALLOCATED TO DDNAME (SYS00001)  084
   DSN (PETRO.D194.T0904332.INTIPCS                    )
   STORCLAS (STANDARD) MGMTCLAS (MIGONLY) DATACLAS (    )
   VOL SER NOS= SL7330
IGD104I PETRO.D194.T0904332.INTIPCS
  RETAINED,  DDNAME=SYS00001
IEA822I COMPLETE TRANSACTION DUMP WRITTEN TO
  PETRO.D194.T0904332.INTIPCS
+CEE3797I LANGUAGE ENVIRONMENT HAS DYNAMICALLY CREATED A
  DUMP.
```



A Guided Tour

The Main IPCS Panel



Initializing the Dump

- Use IPCS Option 0 to tell IPCS which dump to process

```
---------------------- 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('PETRO.D194.T0904332.INTIPCS')
  Address Space    ==>
  Message Routing ==> NOPRINT TERMINAL NOPDS
  Message Control ==> CONFIRM VERIFY FLAG(TERMINATING)
  Display Content ==> MACHINE REMARK REQUEST NOSTORAGE SYMBOL

Press ENTER to update defaults.
```

Initializing the Dump…

- Issue an IPCS command to cause IPCS to initialize the dump

```
------------------ IPCS PRIMARY OPTION MENU  ----------------------------
OPTION  ===> ip status faildata
   0  DEFAULTS     - Specify default dump and options
   1  BROWSE       - Browse dump data set
   2  ANALYSIS     - Analyze dump contents
```

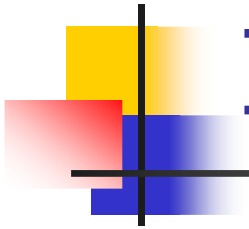
```
IKJ56650I TIME-10:11:37 AM. CPU-00:00:01 SERVICE-53266 SESSION-01:13:40 JULY
BLS18122I Initialization in progress for DSNAME('PETRO.D194.T0904332.INTIPCS')
BLS18124I TITLE=JOBNAME INTIPCS  STEPNAME G                  USER  4039
BLS18223I Dump written by z/OS 01.13.00 SYSMDUMP - level same as IPCS level
BLS18222I z/Architecture mode system
BLS18160D May summary dump data be used by dump access? Enter Y to use, N to
   bypass.
Y
BLS18123I 16,260 blocks, 67,641,600 bytes, in DSNAME('PETRO.D194.T0904332.INT
IKJ56650I TIME-10:11:44 AM. CPU-00:00:01 SERVICE-56368 SESSION-01:13:48 JULY
BLS18224I Dump of z/OS 01.13.00 - level same as IPCS level
 ***
```

IPCS Status Command

```
 *  *  *  DIAGNOSTIC DATA REPORT  *  *  *
SEARCH ARGUMENT ABSTRACT
  RIDS/CEEPLPKA#L RIDS/#UNKNOWN AB/U4039 PRCS/00000000 REGS/0B0DA REGS/05000

  Symptom              Description
  -------              -----------
  RIDS/CEEPLPKA#L      Load module name: CEEPLPKA
  RIDS/#UNKNOWN        Csect name: #UNKNOWN
  AB/U4039             User Abend code: 4039
  PRCS/00000000        Abend reason code: 00000000
  REGS/0B0DA           Register/PSW difference for R0B: 0DA
  REGS/05000           Register/PSW difference for R05: 000
   ...
```

IPCS Status Command…

```
 ...
Time of Error Information

  PSW: 07851000 80000000 00000000 20FEF8B2
  Instruction length: 02   Interrupt code: 000D
  Failing instruction text: 00181610 0A0D58D0 D00498EC

  Breaking event address: 00000000_00000000
  AR/GR 0-1    00000000/00000000_84000000  00000000/00000000_84000FC7
  AR/GR 2-3    00000000/00000000_2111D9A8  00000000/00000000_00040004
  AR/GR 4-5    00000000/00000000_20FE3FA0  00000000/00000000_20FEF8B2
  AR/GR 6-7    00000000/00000000_20F0E348  00000000/00000000_20F0B448
  AR/GR 8-9    00000000/00000000_2111D9A8  00000000/00000000_2111D53C
  AR/GR 10-11  00000000/00000000_2111E0AF  00000000/00000000_A0FEF7D8
  AR/GR 12-13  00000000/00000000_20F0FBE0  00000000/00000000_211201A8
  AR/GR 14-15  00000000/00000000_A0FE3070  00000000/7FFC8400_00000000

  Home ASID: 014B    Primary ASID: 014B    Secondary ASID: 014B
  PKM: 00C0          AX: 0000              EAX: 0000
  This Task's ASID/TCB: 014B/008D89F0
```

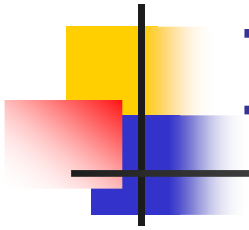


IPCS Browse

- Option 1 on the IPCS Primary Options Panel
  - Allows user to browse raw storage in the dump
  - Maintains a handy list of user-defined pointers
  - To access, specify 1 on the Primary Options Panel, and then hit <enter> on the next panel

IPCS Browse...

```
DSNAME('PETRO.D194.T0904332.INTIPCS') POINTERS  ----------------------------
Command ===>                                                 SCROLL ===> CSR
ASID(X'014B') is the default address space
PTR   Address            Address space                          Data type
00001 00.                ASID(X'014B')                          AREA
      Remarks:
****************************** END OF POINTER STACK ******************************
```



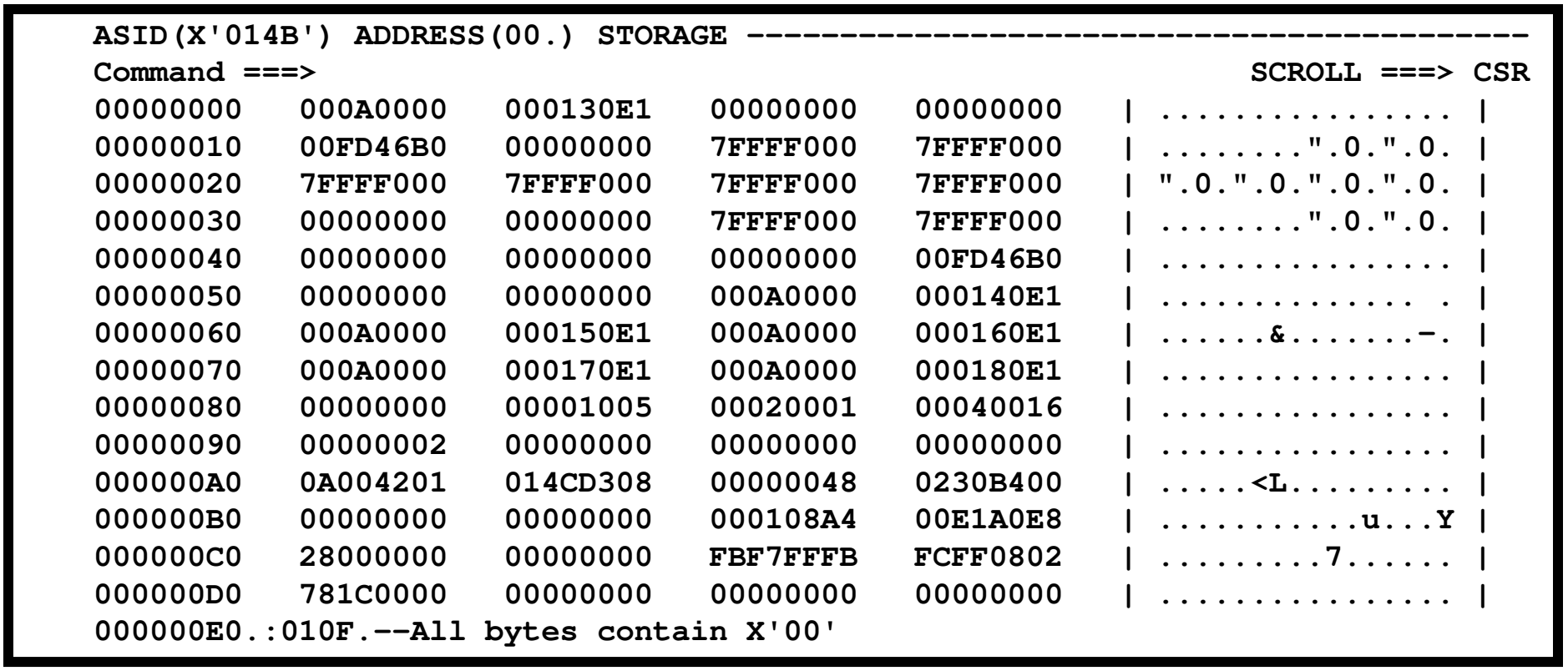
IPCS Browse...

- Pointer Stack Panel – Line Commands
  - Entered by typing over pointer number
    - S – Select a pointer entry for browsing
    - F – Format pointer entry storage
    - I – Insert a pointer entry
    - D – Delete a pointer entry
    - R – Repeat a pointer entry
    - E – Edit a pointer entry

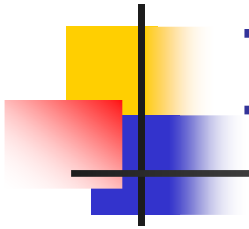
IPCS Browse...

```
DSNAME('PETRO.D194.T0904332.INTIPCS') POINTERS  ----------------------------
Command ===>                                                 SCROLL ===> CSR
ASID(X'014B') is the default address space
PTR   Address            Address space                          Data type
s0001 00.                ASID(X'014B')                          AREA
      Remarks:
****************************** END OF POINTER STACK ******************************
```

IPCS Browse...



```
ASID(X'014B') ADDRESS(00.) STORAGE ----------------------------------------
Command ===>                                                 SCROLL ===> CSR
00000000   000A0000   000130E1   00000000   00000000   | ................ |
00000010   00FD46B0   00000000   7FFFF000   7FFFF000   | ........".0.".0. |
00000020   7FFFF000   7FFFF000   7FFFF000   7FFFF000   | ".0.".0.".0.".0. |
00000030   00000000   00000000   7FFFF000   7FFFF000   | ........".0.".0. |
00000040   00000000   00000000   00000000   00FD46B0   | ................ |
00000050   00000000   00000000   000A0000   000140E1   | ............... . |
00000060   000A0000   000150E1   000A0000   000160E1   | ......&.......-. |
00000070   000A0000   000170E1   000A0000   000180E1   | ................ |
00000080   00000000   00001005   00020001   00040016   | ................ |
00000090   00000002   00000000   00000000   00000000   | ................ |
000000A0   0A004201   014CD308   00000048   0230B400   | .....<L......... |
000000B0   00000000   00000000   000108A4   00E1A0E8   | ...........u...Y |
000000C0   28000000   00000000   FBF7FFFB   FCFF0802   | .........7...... |
000000D0   781C0000   00000000   00000000   00000000   | ................ |
000000E0.:010F.--All bytes contain X'00'
```



IPCS Browse...

- Many ISPF-like Browse commands work
  - PF7|PF8 page up|down
  - PF3 return
  - UP|DOWN n scroll up|down n lines
  - FIND (more later)
  - PF5 Repeat Find

IPCS Browse...

- Locate command – Used to jump to another address

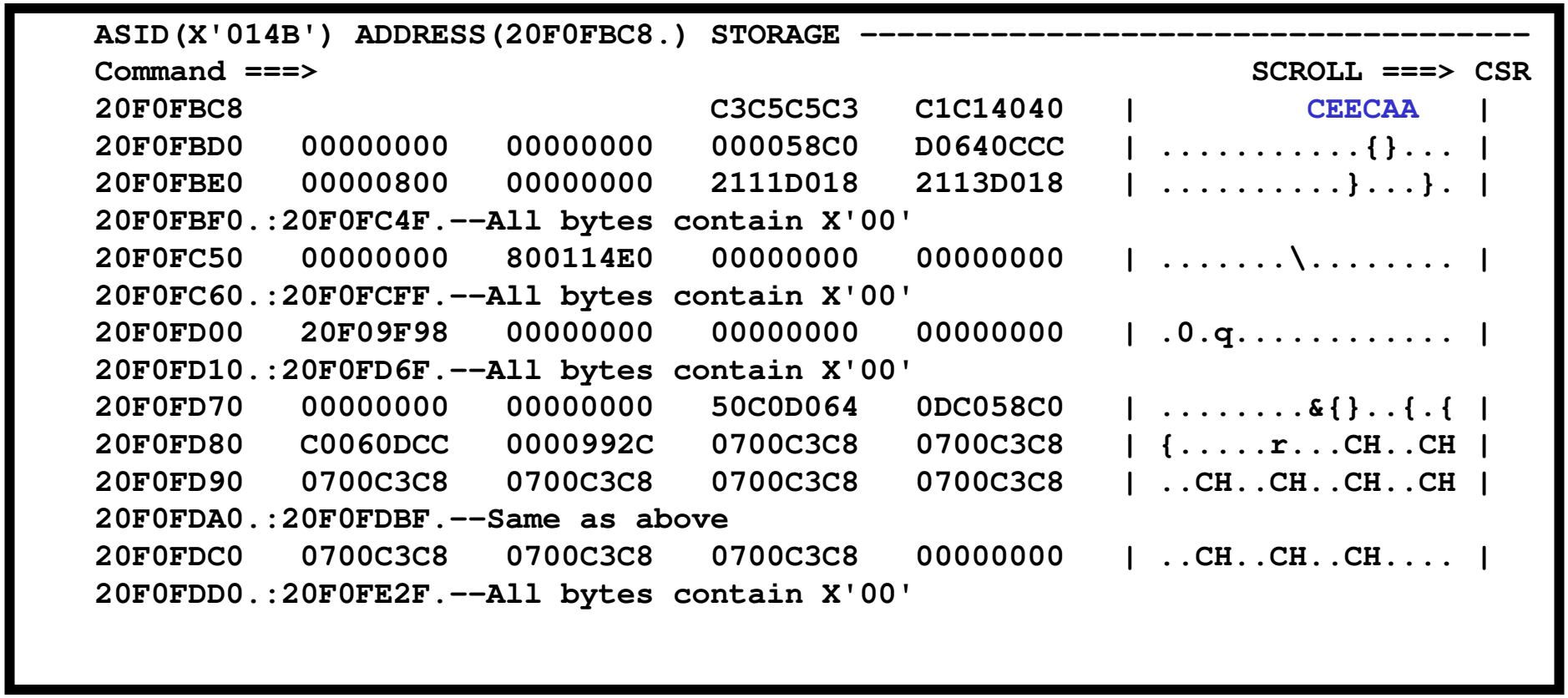
```
ASID(X'014B') ADDRESS(00.) STORAGE ----------------------------------------
Command ===> 1 20F0FBE0                                    SCROLL ===> CSR
00000000   000A0000   000130E1   00000000   00000000   | ................ |
00000010   00FD46B0   00000000   7FFFF000   7FFFF000   | ........".0.".0. |
00000020   7FFFF000   7FFFF000   7FFFF000   7FFFF000   | ".0.".0.".0.".0. |
00000030   00000000   00000000   7FFFF000   7FFFF000   | ........".0.".0. |
00000040   00000000   00000000   00000000   00FD46B0   | ................ |
00000050   00000000   00000000   000A0000   000140E1   | ............... . |
00000060   000A0000   000150E1   000A0000   000160E1   | ......&.......-. |
00000070   000A0000   000170E1   000A0000   000180E1   | ................ |
00000080   00000000   00001005   00020001   00040016   | ................ |
00000090   00000002   00000000   00000000   00000000   | ................ |
000000A0   0A004201   014CD308   00000048   0230B400   | .....<L......... |
000000B0   00000000   00000000   000108A4   00E1A0E8   | ...........u...Y |
000000C0   28000000   00000000   FBF7FFFB   FCFF0802   | .........7...... |
000000D0   781C0000   00000000   00000000   00000000   | ................ |
000000E0.:010F.--All bytes contain X'00'
```

IPCS Browse...

- X – Symbol representing current location in dump

```
 ASID(X'014B') ADDRESS(20F0FBE0.) STORAGE ----------------------------------
Command ===> l x-18                                          SCROLL ===> CSR
20F0FBE0   00000800   00000000   2111D018   2113D018   | .........}...}. |
20F0FBF0.:20F0FC4F.--All bytes contain X'00'
20F0FC50   00000000   800114E0   00000000   00000000   | .......\........ |
20F0FC60.:20F0FCFF.--All bytes contain X'00'
20F0FD00   20F09F98   00000000   00000000   00000000   | .0.q............ |
20F0FD10.:20F0FD6F.--All bytes contain X'00'
20F0FD70   00000000   00000000   50C0D064   0DC058C0   | ........&{}..{.{ |
20F0FD80   C0060DCC   0000992C   0700C3C8   0700C3C8   | {.....r...CH..CH |
20F0FD90   0700C3C8   0700C3C8   0700C3C8   0700C3C8   | ..CH..CH..CH..CH |
20F0FDA0.:20F0FDBF.--Same as above
20F0FDC0   0700C3C8   0700C3C8   0700C3C8   00000000   | ..CH..CH..CH.... |
20F0FDD0.:20F0FE2F.--All bytes contain X'00'
20F0FE30   00000000   00000000   00000000   20F09760   | ............0p- |
20F0FE40   00000000   00000000   8000E420   8000E340   | .........U...T  |
20F0FE50.:20F0FE7F.--All bytes contain X'00'
```

IPCS Browse...



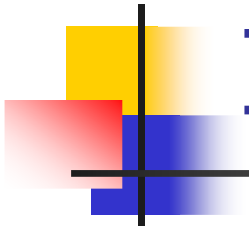
```
ASID(X'014B') ADDRESS(20F0FBC8.) STORAGE ----------------------------------
Command ===>                                                SCROLL ===> CSR
20F0FBC8                          C3C5C5C3   C1C14040   |          CEECAA    |
20F0FBD0   00000000   00000000   000058C0   D0640CCC   | ..........{}...  |
20F0FBE0   00000800   00000000   2111D018   2113D018   | .........}...}.  |
20F0FBF0.:20F0FC4F.--All bytes contain X'00'
20F0FC50   00000000   800114E0   00000000   00000000   | .......\........ |
20F0FC60.:20F0FCFF.--All bytes contain X'00'
20F0FD00   20F09F98   00000000   00000000   00000000   | .0.q............ |
20F0FD10.:20F0FD6F.--All bytes contain X'00'
20F0FD70   00000000   00000000   50C0D064   0DC058C0   | ........&{}..{.{ |
20F0FD80   C0060DCC   0000992C   0700C3C8   0700C3C8   | {.....r...CH..CH |
20F0FD90   0700C3C8   0700C3C8   0700C3C8   0700C3C8   | ..CH..CH..CH..CH |
20F0FDA0.:20F0FDBF.--Same as above
20F0FDC0   0700C3C8   0700C3C8   0700C3C8   00000000   | ..CH..CH..CH.... |
20F0FDD0.:20F0FE2F.--All bytes contain X'00'
```

IPCS Browse...

- Stack command – saves address on the pointer list

```
 DSNAME('PETRO.D194.T0904332.INTIPCS') POINTERS ----------------
Command ===>  ip stack 20F0FBE0                          SCROLL ===> CSR
ASID(X'014B') is the default address space
PTR   Address          Address space                    Data type
00001 00.              ASID(X'014B')                    AREA
      Remarks:
************************ END OF POINTER STACK ************************
```

```
 DSNAME('PETRO.D194.T0904332.INTIPCS') POINTERS  ---------------
Command ===>                                             SCROLL ===> CSR
ASID(X'014B') is the default address space
PTR   Address          Address space                    Data type
00001 00.              ASID(X'014B')                    AREA
      Remarks:
00002 20F0FBE0.        ASID(X'014B')                    AREA
      Remarks: CAA Pointer
************************ END OF POINTER STACK ************************
```



IPCS Browse...

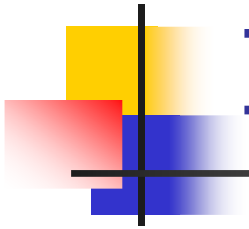
- Word Selection when browsing storage
  - Selection codes used with the storage panel allows a user to treat storage contents as addresses
    - L - Interpret the word as a 24-bit address and stack it
    - H - Interpret the word as a 31-bit address and stack it
    - % - Interpret the word as a 24-bit address, stack it, and display the addressed storage
    - ? - Interpret the word as a 31-bit address, stack it, and display the addressed storage
    - ! - Interpret the double word as a 64-bit address, stack it, and display the addressed storage

IPCS Browse...

```
 ASID(X'014B') ADDRESS(20F0FBE0.) STORAGE ----------------------------------
Command ===>                                                  SCROLL ===> CSR
20F0FBE0   00000800   00000000 ? 2111D018   2113D018   | ..........}...}. |
20F0FBF0.:20F0FC4F.--All bytes contain X'00'
```

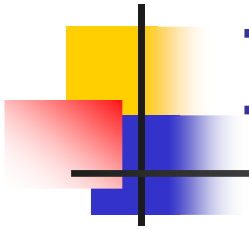
```
 ASID(X'014B') ADDRESS(2111D018.) STORAGE ----------------------------------
Command ===>                                                  SCROLL ===> C
2111D018                        E2E3D2E4   20F10454   |          STKU.1.. |
2111D020   20F10454   00020000   00000000   00000000   | .1.............. |
2111D030   00000000   20F10688   2111D4B0   A110761A   | .....1.h..M.~... |
```

```
DSNAME('PETRO.D194.T0904332.INTIPCS') POINTERS  ---------------------------
Command ===>                                                  SCROLL ===> CSR
ASID(X'014B') is the default address space
...
00002 20F0FBE0.          ASID(X'014B')                          AREA
      Remarks: CAA Address
00003 2111D018.          ASID(X'014B')                          AREA
      Remarks:
```



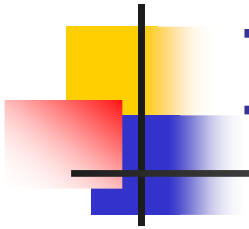
IPCS Browse...

- Finding specific values in storage
  - Use the FIND command
    - Entered on the command line of any storage browse panel
    - find ccc – locates an EBCDIC string
      ex: find ceecaa
    - find x'xxxx' – locates a hexadecimal value
      ex: find x' 47f0f014'
    - find * - uses the same find value as the previous find command
    - find first / last / next / prev
    - find nobreak (or nobr) – tells IPCS to to continue processing if it cannot retrieve storage from the dump
    - Can also specify storage boundary, column boundary, mask value, data length, ASCII data…



IPCS Browse...

- EQUATE command - Allows a user to create a symbol with an associated address and attributes
  - Syntax: EQUATE/EQU/EQ name <addr>
    ex: ip equ caa 20F0FBE0
  - Symbol can be used in places where an address may be specified
    ex: l caa



IPCS Browse...

- EQUATE command...
  - Use "equate *name*" without providing an address to assign the value of "X" (current location) to the specified symbol name
  - Use LISTSYM command to show all defined equates
    - The list will be long, since IPCS defines many equates on its own
  - Use "dropsym *name* purge" to delete a symbol

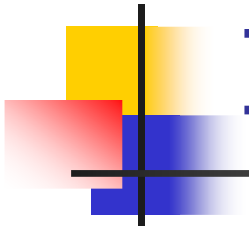
IPCS Browse...

```
IPCS OUTPUT STREAM ---------------------------------------- Line 336 Cols 1 7
Command ===>                                                 SCROLL ===> CSR
 ASCB335         F75880. ASID(X'0001') LENGTH(X'0180') STRUCTURE(Ascb) NODROP
 ASVT            FABDA8. ASID(X'0001') POSITION(X'+01E0') LENGTH(X'2074')
                         STRUCTURE(Asvt) NODROP
 ASXB331         8FDB60. ASID(X'014B') LENGTH(X'0300') STRUCTURE(Asxb) NODROP
 CAA           20F0FBE0. ASID(X'014B') LENGTH(X'04') AREA DROP
 COMMON          900000. ASID(X'0001') LENGTH(X'700000') AREA(Common) NODROP
 COMPONENTID        00. LITERAL LENGTH(X'09') CHARACTER NODROP
 CVT             FD46B0. ASID(X'0001') POSITION(X'-28') LENGTH(X'0528')
                         STRUCTURE(Cvt) NODROP
 CVTXTNT2        FD4C10. ASID(X'0001') LENGTH(X'84') STRUCTURE(Cvtxtnt2) NODROP
 DAESYMPTOMS        00. LITERAL LENGTH(X'54') CHARACTER NODROP
 DSA001        211201A8. ASID(X'014B') LENGTH(X'04') AREA DROP
 DSA002        2111D0B0. ASID(X'014B') LENGTH(X'04') AREA DROP
 DSA003        2111D030. ASID(X'014B') LENGTH(X'04') AREA DROP
 DSA004        20F10688. ASID(X'014B') LENGTH(X'04') AREA DROP
 DSA005            6008. ASID(X'014B') LENGTH(X'04') AREA DROP
 DUMPINGPROGRAM     00. LITERAL LENGTH(X'08') CHARACTER NODROP
```

IPCS Browse...

- List command - Displays storage in the dump
  - Boring by itself, but by adding INST attribute, IPCS will disassemble instructions at the given address
  - ip list 20f00072 len(24) inst :

```
IPCS OUTPUT STREAM ----------------------------------
Command ===>
 ***************************************************
 LIST 20F00072. ASID(X'014B') LENGTH(X'18') INSTRUCTION
 ASID(X'014B') ADDRESS(20F00072.) KEY(88)
 20F00072 | 1812           | LR      R1,R2
 20F00074 | C0B0 FFFF FFC6 | LARL    R11,*-X'74'
 20F0007A | 1744           | XR      R4,R4
 20F0007C | 5010 4000      | ST      R1,X'0'(,R4)
 20F00080 | 4110 B08C      | LA      R1,X'8C'(,R11)
 20F00084 | 58F0 B0A0      | L       R15,X'A0'(,R11)
 20F00088 | 05EF           | BALR    R14,R15
 ***************************************************
```



IPCS Summary Command

- Produces information associated with an address space
  - Defaults to current ASID, but user can identify other(s) to work with using ASIDLIST, JOBLIST, JOBNAME keywords
- summary format (or summ format)
  - produces detailed report of major control blocks in the address space
    - ASCB / ASSB
    - TCBs / STCBs / RBs
    - RTM2WA
    - Load Lists / Job Pack Queue
    - Linkage stacks / Save Areas
    - TCB Summary

IPCS Summary Command...

- TCB Summary
  - Found at the end of the Summary output

```
 * * * *   T C B   S U M M A R Y   * * * *

JOB INTIPCS  ASID 014B ASCB 00F75E80 FWDP 00F75D00 BWDP 00F76100 PAGE
00000017
    TCB AT    CMP      NTC      OTC      LTC      TCB      BACK     PAGE
   008FE040 00000000 00000000 00000000 008FF890 008FD0C0 00000000 00000057
   008FD0C0 00000000 00000000 008FE040 00000000 008FF890 008FE040 00000063
   008FF890 00000000 008FD0C0 008FE040 008FF260 008FF260 008FD0C0 00000067
   008FF260 00000000 00000000 008FF890 008D89F0 008D89F0 008FF890 00000074
   008D89F0 84000FC7 00000000 008FF260 00000000 00000000 008FF260 00000082
```

IPCS Summary Command...

- TCB/STCB/RB Information

```
TCB: 008D89F0
   +0000  RBP..... 008FD548  PIE..... 000060A0  DEB..... 008C9048
   +000C  TIO..... 008CFFD0  CMP..... 84000FC7  TRN..... 00000000
   +0018  MSS..... 7F4BE958  PKF..... 80        FLGS.... 00000000  00
   +0022  LMP..... FF        DSP..... FF        LLS..... 008FDF90
   +0028  JLB..... 008D98E8  JPQ..... 008FFA18

   General purpose register values
     0-3  7F4A9840  000007C0  086782D0  7F43EE70
     4-7  7F458640  00003B5F  00001040  7F440E60
     8-11 7F43FC50  00020003  7F456DE8  00003B5E
    12-15 0B3F8AA0  00000008  03C4C000  000007C0

          64-Bit GPRs from TCB/STCB
    0-1  00000000_7F4A9840  00000000_000007C0
    2-3  00000048_086782D0  00000000_7F43EE70
    4-5  00000048_7F458640  00000000_00003B5F
    6-7  00000000_00001040  00000000_7F440E60
    8-9  00000000_7F43FC50  00000048_00020003
```

IPCS Summary Command...

- RTM2WA Information

```
                              RTM2WA SUMMARY
                              --------------
+001C  Completion code                        84000FC7
+008C  Abending program name/SVRB address GO
+0094  Abending program addr                  20F00000

       GPRs at time of error
     0-3  84000000  84000FC7  2111D9A8  00040004
     4-7  20FE3FA0  20FEF8B2  20F0E348  20F0B448
     8-11 2111D9A8  2111D53C  2111E0AF  A0FEF7D8
    12-15 20F0FBE0  211201A8  A0FE3070  00000000

+06D8  PSW at time of error: 07851000 80000000 00000000 20FEF8B2
+0084  Instruction Length Code: 0002  Interruption Code: 000D
+06C8  Translation Exception Identification: 00000000 00000000
+00DC  SDWACOMP                    00000000
+00E8  Return code from recovery routine-00
   Continue with termination-implies percolation
+00E0  Retry Address returned from recovery exit 00000000
```

IPCS Summary Command...

- Load List

```
EP....... CEEMENU3
ENTPT.... A1149000  RRBP..... 00000000  USE..... 0001      SP....... FC
Reenterable. Reusable.
LOADCNT.. 0001      SYSCT.... 0000
NRFAC.... 00000001  MSBAD.... 21149000  LNTH..... 0000C810
NAMEL.... 0008      ASID..... 014B      PROVIDI.. 00000002
PROVIDD.. 00000000  00D31200  E2E3C5D7  D3C9C240
EPTOKEN.. 00000C3E  014B001F

EP....... CEEPLPKA
ENTPT.... A0F11000  RRBP..... 00000000  USE..... 0001      SP....... FC
Reenterable. Reusable.
LOADCNT.. 0001      SYSCT.... 0000
NRFAC.... 00000001  MSBAD.... 20F11000  LNTH..... 00207110
NAMEL.... 0008      ASID..... 014B      PROVIDI.. 00000002
PROVIDD.. 00000000  05020800  E2E3C5D7  D3C9C240
EPTOKEN.. 00000C3D  014B004D
```

IPCS Summary Command...

- Linkage Stack Entry

```
LINKAGE STACK ENTRY  00  FROM TCB.  LSED: 7F495010
 LSEH: 7F495000
    FNXT..... 00000000  BSEA..... 00000000  TYPE..... 89
     HEADER ENTRY
    RFS...... 0FD0      NES...... 0128

LINKAGE STACK ENTRY  01  FROM TCB.  LSED: 7F495138
 LSE: 7F495018
    GENERAL PURPOSE REGISTER VALUES
    00-01.... 00000000  0B3A9068  00000000  A0F0B200
    02-03.... 00000000  A0F0B200  00000000  0000E670
      ...
PKM...... 00C0      SASN..... 014B      SINS..... 00000001
EAX...... 0000      PASN..... 014B      PINS..... 00000001
PSW...... 07043000  80000000            PSWE..... 00000000  0B3A61C4
TARG..... 00000000  8B3E1B42            MSTA..... 00000000  00000000
TYPE..... 8C
```

Where Command

- Identifies an area at a given address
  - where 20f00080

```
ASID(X'014B') 20F00080. AREA(Subpool251Key08)+80 IN EXTENDED PRIVATE
ASID(X'014B') 20F00080. GO+80 IN EXTENDED PRIVATE
```

  - where 20fef8b2

```
ASID(X'014B') 20FEF8B2. AREA(Subpool252Key00)+0DE8B2 IN EXTENDED PRIVATE
ASID(X'014B') 20FEF8B2. CEEPLPKA+0DE8B2 IN EXTENDED PRIVATE
```

Master Trace

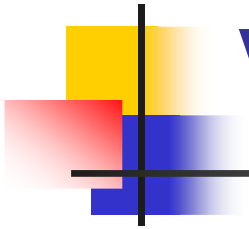
- Option 2.7.3
  - Formats the master trace table, containing the most recently issued console messages

```
21:07:26.84 TSU00010 00000090  $HASP100 WELLIE0  ON TSOINRDR
21:07:27.09 TSU00010 00000090  $HASP373 WELLIE0  STARTED
21:07:35.21 POSIXCON 00000090  slip
   set,a=svcd,c=0c4,enable,sdata=(csa,sum,trt,psa,nuc,sqa,grsq,rgn,lpa),end
21:07:35.22 POSIXCON 00000090  IEE727I SLIP TRAP ID=0001 SET
21:09:35.53 JOB00011 00000090  $HASP100 INTIPCS ON INTRDR PETRO FROM WELLIE0
21:09:35.56 JOB00011 00000290  IRR010I USERID WELLIE0 IS ASSIGNED TO THIS JOB.
21:09:35.80 JOB00011 00000090  ICH70001I WELLIE0  LAST ACCESS AT 21:07:26 ON
   WEDNESDAY, JULY 18, 2012
21:09:35.80 JOB00011 00000090  $HASP373 INTIPCS STARTED - INIT 1 - CLASS 2
   21:09:36.82          00000090  IEA045I AN SVC DUMP HAS STARTED AT
   TIME=21.09.36 DATE=07/18/2012 523
                523 00000090  FOR ASID (0019)
                523 00000090  QUIESCE = YES
```

System Trace

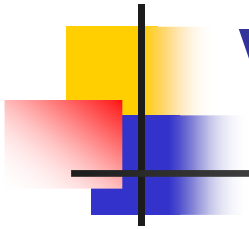
- Option 2.7.4 or SYSTRACE command
  - Formats the system trace table, containing information on significant system events
    - Sample trace entries:

```
00-014B 008D89F0 SVC  6D 00000000_20FB1B62  0000001C 00000004 2111D5A8 Espie
                         07850000 80000000
00-014B 008D89F0 SSRV 78          8AEF2CA0  0000FA12 00000160 000060A0 Getmain
                                            014B0000
00-014B 008D89F0 SVC  3C 00000000_20FB19EA  A0FB1720 00000100 A111D5A8 Estae
                         07851000 80000000
00-014B 008D89F0 PGM 004 00000000_20F00080  00040004 00000000          00000000
                         07850000 80000000            00000000          00000000
00-014B 008D89F0*SVC   D 00000000_20FEF8B2  00000000 84000000 84000FC7
                         07851000 80000000
```



VERBEXIT Command

- Calls an IBM or user-supplied verb exit
  - Optionally can provide parameters to tailor how the verb exit runs
  - Verb Exits useful for application debugging
    - LEDATA – Formats Language Environment diagnostic information as well as application information
    - OMVSDATA – Formats z/OS UNIX diagnostic information
    - DFHPDxxx – Formats CICS diagnostic information (and Language Environment info, too!)



VERBEXIT Command...

- LEDATA Parameters

Report type parameters:

SUMMARY | ALL

HEAP | STACK | SM

HPT(value)

CM

MH

CEEDUMP

COMP(value)

PTBL(value)

Control block selection parameters:

CAA(caa-address)

DSA(dsa-address)

TCB(tcb-address)

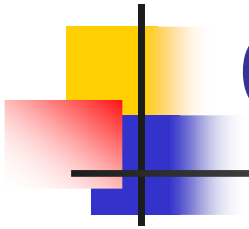
ASID(address-space-ID)

NTHREADS(value)

Additional Component Support

- Option 2.6 – over 50 components!

```
------------------ IPCS MVS DUMP COMPONENT DATA ANALYSIS --------------------
OPTION ===>                                                SCROLL ===> CSR

To display information, specify "S option name" or enter S to the left
of the option desired.  Enter ? to the left of an option to display
help regarding the component support.

S Name     Abstract
  ALCWAIT  Allocation wait summary
  AOMDATA  AOM analysis
  APPCDATA APPC/MVS Data Analysis
  ASCHDATA APPC/MVS Scheduler Data Analysis
  ASMCHECK Auxiliary storage paging activity
  ASMDATA  ASM control block analysis
  AVMDATA  AVM control block analysis
  CICS410  CICS Version 4 Release 1 analysis
  COMCHECK Operator communications data
  COUPLE   XCF Coupling analysis
  CSFDATA  ICSF control block analysis
  CTRACE   Component trace summary
  DAEDATA  DAE header data
  DB2DATA  DB2 analysis
  DIVDATA  Data in virtual storage
```



CBFORMAT Command

- Formats a control block
  - Syntax: CBF(ORMAT) <cbaddr> STR(<cbname>)
    - <cbaddr> can be address or symbol
    - STR(UCTURE) support provided by various components
      - IPCS Commands, Appendix D for MVS control blocks
      - Language Environment Debugging Guide for LE control blocks
        - CEExxx for AMODE 24/31 (CEECAA, CEECIBH, CEEDSA, CEEEDB, CEEHANC, CEESTKH, etc.)
        - CELxxxx for AMODE 64 (CELCIBH, CELDSA, CELEDB, CELLAA, CELLCA, CELSANC, etc.)

CBFORMAT Command…

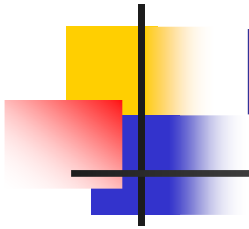
- ip cbf 211201A8 str(ceedsa)
  OR
  ip cbf dsa1 str(ceedsa)

```
IPCS OUTPUT STREAM ----------------------------------------------
Command ===>
 ************************************************************
      DSA:  211201A8
   +000000  FLAGS:0000  MEMD:0000   BKC:2111D0B0      FWC:21120280
   +00000C  R14:A10F5776      R15:A10F7818      R0:2111D594
   +000018  R1:2112023C       R2:00000000       R3:21120268
   +000024  R4:A0F11000       R5:00000000       R6:21120264
   +000030  R7:00000000       R8:00000001       R9:21120258
   +00003C  R10:2111D654      R11:A10F5558      R12:20F0FBE0
   +000048  LWS:00000000      NAB:21120248      PNAB:00000000
   +000064  RENT:00000000     CILC:00000000     MODE:00000000
   +000078  RMR:00000000
 ************************************************************
```

CBFORMAT Command…

- Can be used from Browse's Pointer Stack

```
DSNAME('PETRO.D194.T0904332.INTIPCS') POINTERS  ----------------------------
Command ===>                                                  SCROLL ===> CSR
ASID(X'014B') is the default address space
PTR   Address          Address space                        Data type
00001 00.              ASID(X'014B')                        AREA
      Remarks:
f0002 20F0FBE0.        ASID(X'014B')                        STRUCTURE(Ceecaa)
      Remarks: CAA Pointer
***************************** END OF POINTER STACK ******************************
```



RUNCHAIN Command

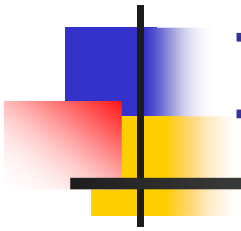
- Allows the user to process a chain of control blocks
- User provides:
  - Starting address
  - Link offset
  - Additional parameters to tell RUNCHAIN what to do with each control block / address

RUNCHAIN Command...

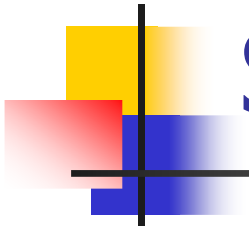
```
ip runc address(211201a8) link(4) display length(x'50`) name(dsa)

DSA001
LIST 211201A8. ASID(X'014B') LENGTH(X'50') AREA
ASID(X'014B') ADDRESS(211201A8.) KEY(88)
211201A8.                   00000000 2111D0B0 |        ......}.|
211201B0. 21120280 A10F5776 A10F7818 2111D594 |....~...~.....Nm|
211201C0. 2112023C 00000000 21120268 A0F11000 |.............1..|
211201D0. 00000000 21120264 00000000 00000001 |................|
211201E0. 21120258 2111D654 A10F5558 20F0FBE0 |......O.~....0.\|
211201F0. 00000000 21120248                   |........         |

DSA002
LIST 2111D0B0. ASID(X'014B') LENGTH(X'50') AREA
ASID(X'014B') ADDRESS(2111D0B0.) KEY(88)
2111D0B0. 0808CEE1 2111D030 211201A8 A0FE3070 |......}....y....|
2111D0C0. A0FEF7D8 2111D53C 2111D50C 20F0E7F8 |..7Q..N...N..0X8|
2111D0D0. 00000794 20FE3FA0 20F09F50 00000000 |...m.....0.&....|
2111D0E0. 20F0B448 A0FE2E9A 2111F0AE 2111E0AF |.0........0...\.|
2111D0F0. 20FDF188 20F0FBE0 00000000 211201A8 |..1h.0.\.......y|
```

RUNCHAIN Command...

```
ip runc address(211201a8) link(4) name(dsa) exec((cbf x str(ceedsa)))

LIST 211201A8. ASID(X'014B') LENGTH(X'04') AREA
ASID(X'014B') ADDRESS(211201A8.) KEY(88)
      DSA:  211201A8
   +000000  FLAGS:0000  MEMD:0000   BKC:2111D0B0      FWC:21120280
   +00000C  R14:A10F5776      R15:A10F7818      R0:2111D594
   +000018  R1:2112023C       R2:00000000       R3:21120268
   +000024  R4:A0F11000       R5:00000000       R6:21120264
   +000030  R7:00000000       R8:00000001       R9:21120258
   +00003C  R10:2111D654      R11:A10F5558      R12:20F0FBE0
   +000048  LWS:00000000      NAB:21120248      PNAB:00000000
   +000064  RENT:00000000     CILC:00000000     MODE:00000000
   +000078  RMR:00000000

LIST 2111D0B0. ASID(X'014B') LENGTH(X'04') AREA
ASID(X'014B') ADDRESS(2111D0B0.) KEY(88)
      DSA:  2111D0B0
   +000000  FLAGS:0808  MEMD:CEE1   BKC:2111D030      FWC:211201A8
   +00000C  R14:A0FE3070      R15:A0FEF7D8      R0:2111D53C
```

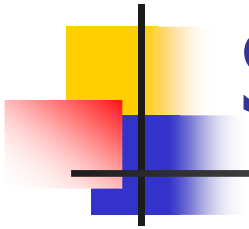


Sources of Additional Information



Sources of Additional Info

- All Language Environment documentation is available on the z/OS DVD collection and on the Language Environment website
  - Language Environment Debug Guide
  - Language Environment Runtime Messages
  - Language Environment Programming Reference
  - Language Environment Programming Guide
  - Language Environment Customization
  - Language Environment Migration Guide
  - Language Environment Writing ILC Applications
- Language Environment Web site
  - http://www-03.ibm.com/systems/z/os/zos/features/lang_environment/



Sources of Additional Info…

- IPCS documentation is available on the z/OS DVD collection
  - MVS IPCS Commands
  - MVS IPCS User's Guide