



Session B49

IMS Debugging in the 64-bit World

Jeff Maddix- (408)463-4956

IMS
Technical Conference

Sept. 27-30, 2004

Orlando, FL



IMS Debugging in a 64-Bit World - Regarding this Presentation

**This procedure is intended to help systems programmers diagnose IMS
This presentation documents information that is Diagnosis Information
provided by IMS.**

**Attention: Do not use this Diagnosis Information as a programming
interface.**

Copyright IBM Corporation 2004



IMS Debugging in a 64-Bit World Agenda

▲ Agenda

- **Section 1: 64-Bit Terminology**
- **Section 2: Recognizing z/Architecture in Dumps**
- **Section 3: z/Architecture Related Changes**
- **Section 4: MVS Formatting Changes**
- **Section 5: Verify Buffers Backed Above the Bar**
- **Section 6: ABEND0C4 RC38 Example**
- **Section 7: Appendix - IMS z/Architecture Related APARs**



Section 1: 64-Bit Terminology

Section 1:

64-Bit Terminology



Hardware Terminology

▲ New hardware terminology

- Unofficial prerelease term "Freeway", "T-Rex"
- z/Architecture servers
 - z900, z990 series of processors
 - ▶ Constraint relief for workloads limited by 2GB real storage
 - ▶ Supports both ESA/390 and 64-bit z/Architecture
- Device Type
 - D/T2064
 - D/T2084
 - ▶ Actual keyword used in RETAIN database APARs to describe this hardware



Software Terminology

▲ New software terminology

- **z/Architecture**
 - The next step in the evolution from IBM's ESA/390 Architecture including 64-Bit general/control registers and a 64-bit addressing mode
- **z/Architecture Mode**
 - IPL with ARCHLVL 2 in LOADxx SYS1.PARMLIB member
 - Default for z/OS systems if running on zSeries processor
- **Extended Addressing Mode or 64-Bit Addressing Mode**
 - Addressing mode may be entered via:
 - ▶ SAM64, LPSWE, stacking PC/PR, BASSM/BSM, Interruptions
 - ▶ See z/Architecture Principles of Operation
 - ◆ Document Number SA22-7832



Software Terminology, Cont.

▲ Software Terminology, Cont.

- "Above the Bar"
 - Reference to storage above 2GB
- ESAME
 - Enterprise Systems Architecture Modal Extensions
 - This term is obsolete and replaced by z/Architecture but you may spot it used in system control blocks and in some messages



64-Bit Operating Systems

▲ 64-Bit Capable Operating Systems

- **OS/390 V2R10 - Planned to remain in service through Sept 2004**
 - Minimum release to run in 64-Bit z/Architecture mode
 - IPL with ARCHLVL 2 in LOADxx SYS1.PARMLIB member
- **z/OS V1R1 GA March 2001**
- **z/OS V1R2 GA Sept 2001**
- **z/OS V1R3 GA March 2002**
- **z/OS V1R4 GA September 2002**
- **z/OS V1R5 GA March 2004**
- **z/OS V1R6 GA September 2004**



64-Bit Operating Systems, Continued

- ▲ **Note:** z/OS V1R2/3/4 Bimodal Migration Accommodation for z/Architecture Servers.
- Allows for a 6 month "fallback" option to 31-bit mode when first migrating to z/OS on a z/Architecture server
 - Previously, z/OS had to run in 64-bit z/Architecture mode (ARCHLVL 2) on a z/Architecture server
 - See the following for details:
<http://www.ibm.com/servers/eserver/zseries/zos/downloads/>



Section 2: Recognizing z/Architecture in Dumps

Section 2:

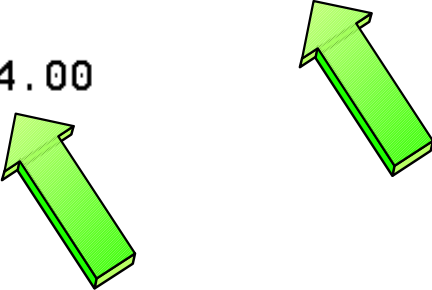
Recognizing z/Architecture in Dumps



Use Same release of IPCS when possible

- △ Dump initialization message BLS18223I tells you the z/OS level starting in z/OS 1.2
- △ Message BLS21001I tells you the level of IPCS being used

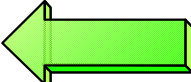
```
BLS18122I Initialization in progress for DSNAME('IMSDUMP.X020805.Y120942.PM7196
BLS18124I TITLE=IMS6 FOR IBM
BLS18223I Dump written by z/OS 01.02.00 SVC dump - level differs from IPCS level
BLS21001I IPCS for z/OS 01.04.00
BLS18222I ESA mode system
```





Recognizing the MVS Level

▲ Select Option 1 (BROWSE) from the IPCS Primary Option Menu

```
----- IPCS PRIMARY OPTION MENU -----  
OPTION ==> 1_   
  
0  DEFAULTS      - Specify default dump and options      * USERID  
1  BROWSE        - Browse dump data set                  * DATE  
2  ANALYSIS     - Analyze dump contents                 * JULIAN  
3  UTILITY       - Perform utility functions            * TIME  
4  INVENTORY    - Inventory of problem data            * PREFIX  
5  SUBMIT       - Submit problem analysis job to batch * TERMINAL  
6  COMMAND      - Enter subcommand, CLIST or REXX exec * PF KEYS  
T  TUTORIAL     - Learn how to use the IPCS dialog      *  
X  EXIT         - Terminate using log and list defaults *  
*****
```

Enter END command to terminate IPCS dialog



Recognizing the MVS Level Continued...

▲ Hit <Enter> to advance to IPCS Pointers Panel

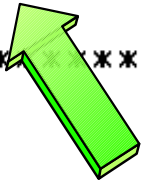
```
----- IPCS - ENTRY PANEL -----  
Command ==>  
  
CURRENT DEFAULTS:  
Source ==> DSNAME('IMSDUMP.X020805.Y120942.PM71967.M030776')  
Address space ==> ASID(X'0001')  
  
OVERRIDE DEFAULTS:                                     (defaults used for blank fields)  
Source ==> DSNAME('IMSDUMP.X020805.Y120942.PM71967.M030776')  
Address space ==>  
Password      ==>  
  
POINTER:  
Address      ==>                                     (blank to display pointer stack)  
Remark      ==>                                     (optional text)
```



Recognizing the MVS Level Continued...

▲ Enter line command: S

```
DSNAME('IMSDUMP.X020805.Y120942.PM71967.M030776') POINTERS -----  
Command ==> SCROLL ==> CSR  
ASID(X'0001') is the default address space  
PTR   Address   Address space           Data type  
s0001  00000000  ASID(X'0038')           AREA  
Remarks:  
***** END OF POINTER STACK *****
```





Recognizing the MVS Level Continued...

▲ Enter Command: L CVT

```

ASID(X'0038') STORAGE -----
Command ==> L CVT ← CMD
00000000 040C0000 814000E0 00000000 00000000
00000010 00FC97C0 00000000 070E0000 00000000
00000020 070C0000 85652522 070C4000 8566CE3A
00000030 00000000 00000000 070E0000 00000000
00000040 00000000 00000000 00000000 00FC97C0
00000050 00000000 00000000 040C0000 8134A8C0
00000060 040C0000 80FFE280 00080000 808A61E0
00000070 00080000 BD27D788 040C0000 813E2E80
00000080 00000000 00001005 00020038 00060011
00000090 00002001 00010000 00000000 00000000
000000A0 0C000000 013F2808 00000000 00000000
000000B0 00000000 00000000 00010965 02F5B638
000000C0 18000000 00000000 00000000 00000000
000000D0 00000000 15B05000 00000000 00000000
000000E0.:019F.--All bytes contain X'00'

          _ SCROLL ==> CSR
          |-----|
          |...a.\.....|
          |..p{.....|
          |...e.....e...|
          |.....|
          |.....p{
          |.....a.y{
          |.....S...../\
          |.....Ph...a...
          |.....|
          |.....5..
          |.....|
          |.....&.....|
    
```



Recognizing the MVS Level Continued...

▲ Enter Command: L CVT

```

ASID(X'0001') ADDRESS(FCC618.) STORAGE -----
Command ==> l cvt
00FCC5F0 E2D7F74B F04BF440 C8C2C2F7 F7F0F740 | SP7.0.4 HBB7707 |
00FCC600.:FCC60F.--All bytes contain X'40', C' '
00FCC610 00002064 F0F3F840 00000218 00FE66C4 | ...038 ...D |
00FCC620 00FCC594 00FCCC00 00000000 00FDE124 | ..Em..... |
00FCC630 00FEC25E 00FE4188 00FE3FBC 01857208 | ..B;...h...e |
00FCC640 814BCDB8 00FD3FD0 02150E30 00FE50B8 | a.....}.....& |
00FCC650 0104223F 00FC8CD0 00F3C000 00FD4860 | .....}.3{...- |
00FCC660 00FE66E8 00000000 0A0307FE 00FCC59C | ...Y.....E. |
00FCC670 00FC8B28 00000000 40C3E5E3 00FCD620 | ..... CVT..0. |
00FCC680 00FD2392 00FD23B2 00F3C2B8 9BFCF8B8 | ...k....3B...8. |
00FCC690 00620001 00FD4D48 00000000 81530250 | .....(.....a..& |
00FCC6A0 00FE6598 01868598 8A4B1000 00FC8CD0 | ...q.feq.....} |
00FCC6B0 00FE88A0 00D0D330 55000282 7FFFFFFF | ..h..}L...b"... |
00FCC6C0 00000000 00000000 00FCCC28 0000A320 | .....t. |
00FCC6D0 00FE4458 00FCC5D0 00FCCC60 80F870E0 | .....E}...-.8.\ |
00FCC6E0 00FC8A30 00000000 00000000 0A0D0A06 | ..... |
00FCC6F0 00000000 00FCC0D8 014F19E8 02722F44 | .....{Q.|.Y... |
00FCC700 00FE3610 00FCA860 00000000 00000000 | .....y-..... |
00FCC710 00F94CC0 00FCCB68 00F7C700 00000000 | .9<{.....7G.... |
00FCC720 80000000 01860288 00E70000 00FE5A00 | .....f.h.X....! |
00FCC730 814BDB90 00000000 00FFEEA8 234ED4B8 | a.....y.+M. |
00FCC740 00FC8DA8 00000000 FFFFC A5B 00000000 | ...y.....$. |
    
```





Recognizing the MVS Level Continued...

▲ MVS Release to FMID relationship

OS/390 V2R8	HBB6608
OS/390 V2R9	JBB6609
OS/390 V2R10	HBB7703
z/OS 1.1	JBB7713
z/OS 1.2	HBB7705
z/OS 1.3	HBB7706
z/OS 1.4	HBB7707
z/OS 1.5	HBB7708
z/OS 1.6	HBB7709
	HBB7720

Note: Currently, starting with z/OS 1.1, the last 3 digits add up to the release

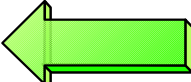
HBB7707 ==> $7+0+7 = 14$ ==> z/OS 1.4



Recognizing the Device Type

D/T 2064

▲ Select Option 1 (BROWSE) from the IPCS Primary Option Menu

```
----- IPCS PRIMARY OPTION MENU -----  
OPTION ==> 1_   
  
0  DEFAULTS      - Specify default dump and options      * USERID  
1  BROWSE        - Browse dump data set                  * DATE  
2  ANALYSIS     - Analyze dump contents                 * JULIAN  
3  UTILITY      - Perform utility functions            * TIME  
4  INVENTORY    - Inventory of problem data            * PREFIX  
5  SUBMIT       - Submit problem analysis job to batch * TERMINAL  
6  COMMAND      - Enter subcommand, CLIST or REXX exec * PF KEYS  
T  TUTORIAL     - Learn how to use the IPCS dialog      *  
X  EXIT         - Terminate using log and list defaults *  
*****
```

Enter END command to terminate IPCS dialog



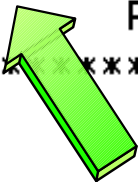
Recognizing the Device Type D/T 2064

▲ Enter line command: S

```

DSNAME('IMSDUMP.X010712.Y133100.SJFEVMX.MADDIX') POINTERS -----
Command ==>                                     SCROLL ==> CSR
ASID(X'0001') is the default address space
PTR  Address  Address space  Data type
S0001 00000000 ASID(X'00B8') AREA
Remarks:
***** END OF POINTER STACK *****

```





Recognizing the Device Type

D/T 2064

▲ Enter Command: L_CVT+42C?

```

ASID (X'00B8') STORAGE -----
Command ==> L_CVT+42C?  ← CMD
00000000  000A0000  000130E1  00000000  00000000  | ..... |
00000010  00FD1B18  00000000  00000000  00000000  | ..... |
00000020.:2F.--All bytes contain X'00'
00000030  00000000  00000000  470C0000  A1E7E9EE  | .....~XZ. |
00000040  00000000  00000000  00000000  00FD1B18  | ..... |
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  00000001  0133D108  00000000  0078B000  | .....J..... |
000000B0  00000000  00000000  0001369E  01F589E0  | .....5i\ |
000000C0  28000000  00000000  E0000000  00000000  | .....\..... |
000000D0.:012F.--All bytes contain X'00'
00000130  47041000  80000000  00000000  0A4282E2  | .....bS |
00000140  47851400  80000000  00000000  21CF0A66  | .e..... |
00000150  44040000  80000000  00000000  0134AE14  | ..... |
00000160.:016F.--All bytes contain X'00'
00000170  07060000  00000000  00000000  00000000  | ..... |
    
```



Recognizing the Device Type

D/T 2064

- ▲ You will see the machine information here at the CVTHID area
 - The device type can be found at offset x'1C'

```

ASID(X'0001') STORAGE -----
Command ==> _
016F2728          00000396    012064B6
016F2730    0478E100    00000000    00000000    00001800
016F2740    0003F0F0    F2F0F6F4    F1C3F5C9    C2D4F0F2
016F2750    F0F0F0F0    F0F0F0F4    F0F0F0F0    FFF00000
016F2760    C9D6E2E5    E2D9C3D2    F0F261F2    F661F0F1
016F2770    E4E6F7F7    F9F1F440    00000000    00000000
016F2780.:016F27AF.--All bytes contain X'00'
016F27B0    00000000    00000000    C9D6E2E5    C9E2C4E3
016F27C0    F0F261F2    F661F0F1    E4E6F7F7    F9F1F440
016F27D0    00081018    20283038    C9D6E2E5    C3C8D9C2
016F27E0    F0F261F2    F661F0F1    E4E6F7F7    F9F1F440
016F27F0    C3C8D9C2    00000000    00000000    00000000
016F2800    00000000    016F2820    00000000    81100BE8
016F2810    00000000    00000080    00000000    00000000
016F2820    E2D9C240    016F284C    00000000    00000000
016F2830.:016F283F.--All bytes contain X'00'
016F2840    00000000    00000000    00000000    E2D9C240
016F2850    016F2878    00000000    00000000    00000000
016F2860.:016F286F.--All bytes contain X'00'

```

	SCROLL ==>	CSR
	...	0...
	..	0020641C5IBM02
	00000000	40000.0..
	IOSVSI	02/26/01
	UW77914
	IOSVISDT
	02/26/01	UW77914
	IOSVCHRB
	02/26/01	UW77914
	CHRB.....	
?a..Y
	
	SRB .?<.....	
	SRB
	.?



Recognizing z/Architecture Mode IPL - Globally with CVT

▲ Enter Command: CBF CVT

```
----- IPCS PRIMARY OPTION MENU -----
OPTION ==> IP CBF CVT_  ← CMD
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   - M030776
* DATE     - 01/07/15
* JULIAN   - 01.196
* TIME     - 12:54
* PREFIX   - M030776
* TERMINAL - 3278
* PF KEYS  - 24
*****
```

Enter END command to terminate IPCS dialog



Recognizing z/Architecture Mode IPL - Globally with CVT

▲ Enter Command: F FLAG3

```

IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> F FLAG3_          SCROLL ==> CSR
***** TOP OF DATA *****

```



CVT: 00FD1B18

-0028	PRODN....	SP6.1.0	PRODI....	HBB7703	VERID....	
-0006	MDL.....	2064	RELNO....	038		
+0000	TCBP....	00000218	0EF00....	00FF1064	LINK.....	00FD4C7C
+000C	AUSCB....	00FD2100	BUF.....	00000000	XAPG.....	00FE40A0
+0018	0VL00....	00FF534E	PCNVT....	00FDD030	PRLTV....	00FDCE64
+0024	LLCB....	0170C3F0	LLTRM....	8118D620	XTLER....	00FDB1B8
+0030	SYSAD....	01FDE418	BTERM....	00FDEAE8	DATE....	0101172F
+003C	MSLT....	00FCF0A8	ZDTAB....	00F38000	XITP.....	00FF76B8
+0048	0EF01....	00FF1088	VSS.....	0000	VPSM.....	0000
+0050	EXIT....	0A03	BRET....	07FE	SVDCB....	00FD4C84
+0058	TPC.....	00FD2128	ICPID....	0000	CVT.....	40C3E5E3
+0064	CUCB....	00FCCC70	QTE00....	00FEF182	QTD00....	00FEF1A2
+0070	STB.....	00F496E0	DCB.....	9B	DCBA....	FD7438
+0078	SV76M....	00000000	IXAVL....	00FDD2D8	NUCB....	00000000
+0084	FBOSV....	8136FEE0	ODS.....	00FF0F38	ECVT....	016E9B58
+0090	DAIRX....	8A124000	MSER....	00FCF0A8	OPT01....	00FEA5B0



Recognizing z/Architecture Mode IPL - Globally with CVT

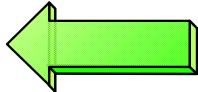
▲ FLAG3, CVT+x'17A' = x'80' bit is set if IPLed in z/Architecture Mode

```
IPCS OUTPUT STREAM ----- FOUND: LINE 39 C1
Command ==>                               SCROLL ==>
+0174 SYLK..... 00          SLID..... 000000          FLAG1.... 7E
+0179 FLAG2.... F8          FLAG3.... 80          ← IPLed in z/Arch Mode
+017C RT03..... 00FDED38   VLDWT.... 00000000   EXSNR.... 0118F6B8
+018C EXSNL.... 00          SPVLK.... 00          CTLFG.... 10
+018F APG..... 00          TRACE.... 07FB       TRAC2.... 07FB
+0194 RSCN..... 00FF0EB4   TAS..... 00000000   TRCRM.... 8B7B3BF8
+01A0 SHRVM.... 00800000   OVLO1.... 00FF5348   PPGMX.... 811915A8
+01AC GRSST.... 00          GVT..... 00FD35F0   ASCRF.... 021E4E00
+01B8 ASCRL.... 082F0200   PUTL..... 80DB8A3C   SRBRT.... 00FF0EF0
+01C4 OLT0A.... 00FD25B0   SMFEX.... 00FD2610   CSPIE.... 88A85A10
+01D0 PTGT..... 80DB8A42   SIGPT.... 1E          SPDMC.... 00
+01D6 DSSAC.... 00          STCK..... 80DB8A36   MAXMP.... 0007
+01DE BSM2..... 0B02       SCAN..... 80DC4000   AUTHL.... 7FFFF001
+01E8 BLDCP.... 00FDF220   GETCL.... 00FDF4E0   FRECL.... 00FDF670
+01F4 DELCP.... 00FDF828   CRMN..... 00FF7A6C   CRAS.... 81191A50
+0200 QSAS..... 812A2D3E   FRAS..... 81191A72   S1EE.... 00FF0D44
+020C PARS..... 80DB8A24   QUIS..... 01192D30   STXU.... 8A01BFF0
+0218 OPTE.... 00FD85E6   SDRM..... 8A04C000   CSRT.... 016F4EB8
+0224 AQTOP.... 016F6EB8   VVMDI.... 00000833   ASVT.... 00F9F0D0
```




Recognizing z/Architecture Mode IPL - Using PSA

△ Select Option 1 (BROWSE) from the IPCS Primary Option Menu

```
----- IPCS PRIMARY OPTION MENU -----  
OPTION ==> 1_   
  
0  DEFAULTS      - Specify default dump and options      * USERID  
1  BROWSE        - Browse dump data set                  * DATE  
2  ANALYSIS      - Analyze dump contents                 * JULIAN  
3  UTILITY       - Perform utility functions             * TIME  
4  INVENTORY     - Inventory of problem data            * PREFIX  
5  SUBMIT        - Submit problem analysis job to batch * TERMINAL  
6  COMMAND       - Enter subcommand, CLIST or REXX exec * PF KEYS  
T  TUTORIAL     - Learn how to use the IPCS dialog      *  
X  EXIT          - Terminate using log and list defaults *  
*****
```

Enter END command to terminate IPCS dialog



Recognizing z/Architecture Mode IPL - Using PSA

▲ Enter Line Command: S

```

DSNAME('IMSDUMP.X010712.Y133100.SJFEVMX.MADDIX') POINTERS -----
Command ==>                                     SCROLL ==> CSR
ASID(X'00B8') is the default address space
PTR   Address  Address space                               Data type
-----
s0001 00000000 ASID(X'00B8')                       AREA
Remarks:
***** END OF POINTER STACK *****

```





Recognizing z/Architecture Mode IPL

- Using PSA

▲ PSA+x'A3' = x'01' bit is set if IPLed in z/Architecture Mode

ASID (X'0001')	STORAGE	-----			SCROLL ==>	CSR
Command ==>						
00000000	000A0000	000130E1	00000000	00000000	
00000010	00FD1B18	00000000	00000000	00000000	
00000020.:2F.	--All bytes contain X'00'					
00000030	00000000	00000000	470C0000	A1E7E9EE~XZ.	
00000040	00000000	00000000	00000000	00FD1B18	
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	00000001	00000000	00000000	0078B000J.....	
000000B0	00000000	00000000	0001369E	01F589E05i\	
000000C0	28000000	00000000	E0000000	00000000\.....	
000000D0.:012F.	--All bytes contain X'00'					
00000130	47041000	80000000	00000000	0A4282E2bS	
00000140	47851400	80000000	00000000	21CF0A66	.e.....	
00000150	44040000	80000000	00000000	0134AE14	
00000160.:016F.	--All bytes contain X'00'					
00000170	07060000	00000000	00000000	00000000	

← IPLed in z/Arch Mode



Recognizing z/Architecture Mode IPL - Using Formatted PSA

▲ PSA formatted using Command: IP CBF PSA_n

- PSA AMDID at offset x'A3', x'01' bit is set if IPLed in z/Architecture Mode

```
IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> IP CBF PSA0_          SCROLL ==> CSR
***** TOP OF DATA *****
```

PSA: 00000000

+0010	CVT.....	00FCF098	CVT2.....	00FCF098	EPARM....	00000000
+0084	CPUAD....	0003	EICODE...	1202	SDATA....	00020030
+008C	PDATA....	00060011	PINFO....	00000000	MCNUM....	0000
+0096	PERCODE..	0000	PER.....	00000000	00000000	
+00A0	AID.....	0C	PERAID...	00	OPACID...	00
+00A3	AMDID....	01				
+00A8	TEID.....	00000000	00002001			
+00B0	MONCODE..	00000000	00000000		SSID.....	000101AF
+00BC	IOINTP...	00EF5A28	IOINTID..	28000000	PCFETO...	00000000
+00C8	FACLIST..	E0000000	MCIC.....	00000000	00000000	
+00F0	MCICE....	00000000	EDCODE...	00000000		
+00F8	FSA.....	00000000	00000000			
+0120	ROPSW....	00000000	00000000	00000000	00000000	
+0130	EOPSW....	07040000	80000000	00000000	04222D68	
+0140	SOPSW....	07040000	80000000	00000000	0421C182	
+0150	POPSW....	07044000	80000000	00000000	04223E62	

PSA Format Command

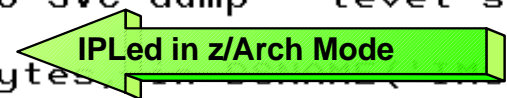
IPLed in z/Arch Mode



Recognizing z/Architecture Mode IPL - Dump Initialization

- △ z/OS 1.1 and below MSG BLS18222I will state "ESAME mode system" If z/Architecture Mode IPL
- △ z/OS 1.2 and higher MSG BLS18222I will state "z/Architecture mode system" If z/Architecture Mode IPL
- △ MSG BLS18222I will state "ESA mode system" if ARCHLVL 1, ESA mode IPL

```
IKJ56650I TIME-04:07:22 PM. CPU-00:00:04 SERVICE-533654 SESSION-01:03:29 AU
16,2004
BLS18122I Initialization in progress for DSNAME('IMSDUMP.X040811.Y131550.PM
7.FREDW')
BLS18124I TITLE=IMSSAGT ABEND SYS 0C4 USER 0000-011, DATE.TIME: 223.233124
LLER=FP , TCB=XFP, MODULE=UNKNOWN ,P
BLS18223I Dump written by z/OS 01.04.00 SVC dump - level same as IPCS level
BLS18222I z/Architecture mode system
BLS18123I 83,174 blocks, 346,003,840 bytes DUMP.X040811.Y131
PM10807.FREDW')
IKJ56650I TIME-04:08:08 PM. CPU-00:00:05 SERVICE-656260 SESSION-01:04:14 AU
16,2004
BLS18224I Dump of z/OS 01.04.00 - level same as IPCS level
***
```





Section 3: z/Architecture Related Changes

Section 3: z/Architecture Related Changes



z/Architecture Related Changes

▲ z/Architecture Related Changes

- 64-bit virtual storage management support starting z/OS V1R2
- 128-bit PSW
- New Program Interrupt Codes
- 163 new assembler instructions
- 64-bit General Purpose Registers
- 64-bit Control Registers
- 8K PSA
- Central storage to 256 GB

Note: Programs continue to be loaded and run below the 2GB Bar. RMODE support remains unchanged



z/Architecture Related Changes - 64-Bit Virtual Storage Management

▲ 64-bit virtual storage management support starting z/OS V1R2

- 64-bit data addressability within a single address space
 - Former limit of 2GB = $2^{31} = 2,147,483,648$ Bytes
 - New limit of $2^{64} = 18,446,744,073,709,551,616$ bytes
 - ▶ 18 and a half quintillion bytes
 - ▶ 16 exabytes = 16,777,216 terabytes = 17,179,869,184 GB
 - ▶ 8,589,934,592 times greater than the 31-Bit address space
 - An MVS/XA 31-Bit address space was only 128 times greater than the System 370 24-BIT address space



z/Architecture Related Changes - 64-Bit Virtual Storage Management, Continued

To download a
2 gigabyte
stand alone
dump at 1789
Kbytes/sec...



...it took 19.5
minutes

If the dump were 16
exabytes...

...it would have taken 319,302
years, 316 days, 12 hours and
45 minutes





z/Architecture Related Changes - 64-Bit Virtual Storage Management, Continued

▲ Storage above the bar is obtained via the IARV64 macro

- Smallest memory object size is 1 megabyte, largest to limit set by MEMLIMIT in IEFUSI. MEMLIMIT is also available at job or SMF level.
- SVC DUMPRGN=YES/NO parm in IARV64 macro is used to specify whether or not the virtual storage in the memory object is to be included when an SVC dump is requested using SDATA=(RGN)
 - The following options can be used to dump above the bar storage:
 - ▶ MVS DUMP command parm, STOR(beg,end)
 - ▶ SDUMPX macro SUMLIST64 or LIST64
 - ▶ MVS SLIP command parm, LIST
- Storage is freed explicitly via IARV64 macro or at task term
- GETMAIN/FREEMAIN, STORAGE, CPOOL, or callable cell pool services do not work on virtual storage above the bar



z/Architecture Related Changes - 64-Bit Virtual Storage Management, Continued

- ▲ **Storage above the bar is obtained via the IARV64 macro, Continued**
 - **Storage is freed explicitly via IARV64 macro or at task term**
 - **GETMAIN/FREEMAIN, STORAGE, CPOOL, or callable cell pool services do not work on virtual storage above the bar**
 - **See the following white paper for the IBM Storage plan reference:**
 - **IBM zSeries 900 z/OS 64-bit Virtual Storage Roadmap**
 - **<http://www-1.ibm.com/servers/eserver/zseries/library/whitepapers/gm130076.html>**



z/Architecture Related Changes - z/Architecture PSW Format, Continued

▲ PSA at address x'00000000' in address space showing 128-bit PSWs

ASID (X'0001')	STORAGE	-----			SCROLL ==>	CSR
Command ==>						
00000000	000A0000	000130E1	00000000	00000000	
00000010	00FD1B18	00000000	00000000	00000000	
00000020	:2F.--All bytes contain X'00'					
00000030	00000000	00000000	470C0000	A1E7E9EE~XZ.	
00000040	00000000	00000000	00000000	00FD1B18	
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	00000001	0133D108	00000000	0078B000J.....	
000000B0	00000000	00000000	0001369E	01F589E05i\	
000000C0	28000000	00000000	E0000000	00000000\	
000000D0	:12F.--All bytes contain X'00'					
00000130	47041000	80000000	00000000	0A4282E2bS	
00000140	47851400	80000000	00000000	21CF0A66	.e.....	
00000150	44040000	80000000	00000000	0134AE14	
00000160	:016F.--All bytes contain X'00'					
00000170	07060000	00000000	00000000	00000000	





z/Architecture Related Changes - z/Architecture PSW Format, Continued

▲ z/Architecture PSWs from PSA:

• PSA+x'130', EOPSW: 47041000 80000000 00000000 0A4282E2

– ESA PSW = 470C1000 8A4282E2

• PSA+x'140', SOPSW: 47851400 80000000 00000000 21CF0A66

– ESA PSW = 478D1400 A1CF0A66

• PSA+x'150', POPSW: 44040000 80000000 00000000 0134AE14

– ESA PSW = 440C0000 8134AE14

▲ The 128-bit z/Architecture PSW is converted to a z/OS apparent 64-bit (ESA/390-like) PSW in most control blocks



z/Architecture Related Changes - z/Architecture Mode PSW Example

▲ 64-bit mode PSW example:

- PSA+x'130', EOPSW: 04045001 80000000 00000000 2A4282E2
- ESA/390 PSW = 040C5001 AA4282E2

▲ PSW Bit 31 and 32 trimodal addressing chart:

Bit-31	Bit-32	Addressing Mode
0	0	24-bit
0	1	31-bit
1	1	64-bit
1	0	Specification Exception (PIC 6)



z/Architecture Related Changes - Program Interrupt Codes

▲ Program Interrupt Codes

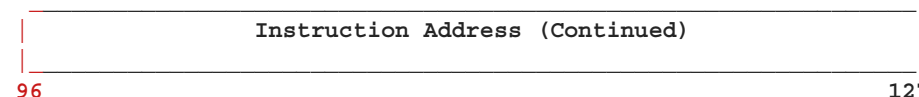
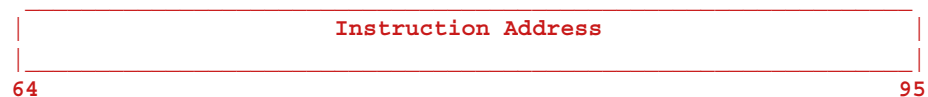
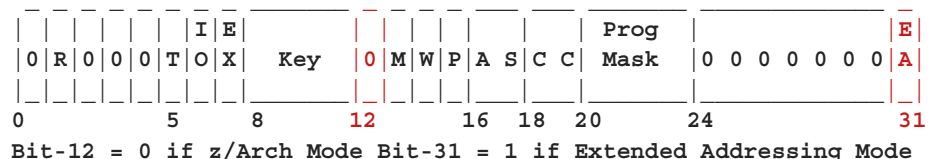
Program-Interruption Conditions	Hex PIC	Address mode	ABEND Code	Error PSW Points
Protection Exception	04	ALL	0C4	At Next Seq Instruction
Specification Exception	06	ALL	0C6	At Next Seq Instruction
Segment Translation	10	ALL	0C4	At Instruction
Page Translation	11	ALL	0C4	At Instruction
Special-Operation Except	13	ALL	0D3	At Next Seq Instruction
ASTE (AS Second Table Entry) Validity Exception	2B	ALL	0C4 : z/OS 1.2 and above 0E0 RC2B: 1.1 and below	At Instruction
ASCE (Address Space Control Element) -Type *	38	64Bit	0C4 : z/OS 1.2 and above 0E0 RC38: 1.1 and below	At Instruction
Region 1st Translation *	39	64Bit	0C4	At Instruction
Region 2nd Translation *	3A	64Bit	0C4	At instruction
Region 3rd Translation *	3B	64Bit	0C4	At instruction

* Unique to z/Architecture mode



z/Architecture Related Changes - PIC z/Architecture Summary

PICs	Hex PIC	Address mode	ABEND	Possible Reasons
Early Spec Except (occurs after operation is complete) Spec Exception	06	ALL	0C6	--- LPSW, LPSWE, PR, Interrupt - ILC = 0 --- Bit 31 of PSW =1, Bit 32 0 Bits 31/32=0 (24Bit Mode) 64-103 \rightarrow 0 Bit 31=0, Bit 32=1 (31Bit Mode) 64-96 \rightarrow 0 --- SAM24, SAM31 ILC=1 or 2 if EX target --- Bits 64-103 or 64-96 \rightarrow 0
Special-Operation	13	ALL	0D3	LRA w/Bits 0-32 of resulting real addr \rightarrow 0
ASCE -Type	38	64Bit	0C4 1.2^ 0E0 RC38	Branch to virtual address above the bar Access unallocated storage above the bar
Reg 1st Tran	39	64Bit	0C4	Access unallocated storage above the bar
Reg 2nd Tran	3A	64Bit	0C4	Access unallocated storage above the bar
Reg 3rd Tran	3B	64Bit	0C4	Access unallocated storage above the bar





z/Architecture Related Changes - Program Interrupt Codes - TEID

▲ The TEID (Translation Exception Identification), formerly known as TEA (Translation Exception Address), identifies the storage address that caused the PIC 10, 11, 38, 39, 3A, or 3B

- Stored at PSA+x'A8', Label TEID, 2 words (64Bit value)
- Also available in Logrec entries in SDWARC4 + x'148'
- Full value not currently available in RTM2WA, SDWA, or IPCS STATUS

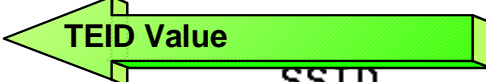
```

IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> IP CBF PSA0_          SCROLL ==> CSR
***** TOP OF DATA *****
    
```



```

PSA: 00000000
+0010  CVT..... 00FCF098  CVT2..... 00FCF098  EPARM.... 00000000
+0084  CPUAD.... 0003      EICODE... 1202      SDATA.... 00020030
+008C  PDATA.... 00060011  PINFO.... 00000000  MCNUM.... 0000
+0096  PERCODE.. 0000      PER..... 00000000  00000000
+00A0  AID..... 0C        PERAID... 00        OPACID... 00
+00A3  AMDID.... 01        MPL..... 0132E588
+00A8  TEID.... 00000D30  00000000
+00B0  MONCODE.. 00000000  00000000  SSID..... 000101AF
    
```



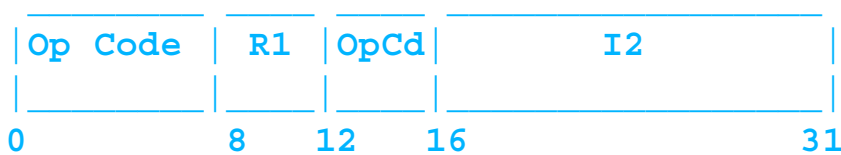


z/Architecture Related Changes - 163 New Assembler Instructions

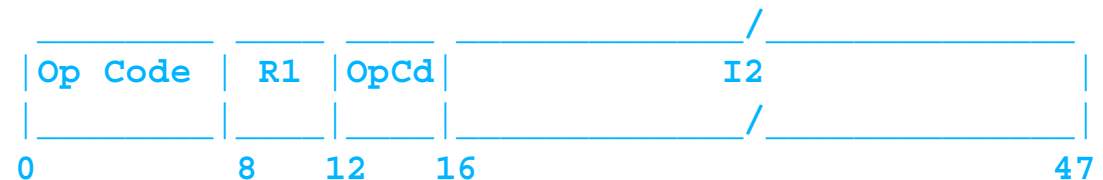
▲ 163 new assembler instructions -

- 141 for z/Architecture only, 22 for both ESA/390 and z/Architecture
- IMS uses some of the new ESA/390 instruction set in V8
 - OS/390 V2R10 and necessary hardware already were a prereq
 - z/Architecture only instructions not used by IMS to avoid necessary checks for instruction availability
- Three new instruction formats: RI, RIL, RIE

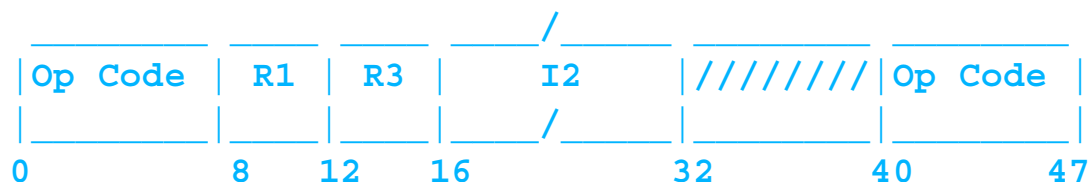
RI Format



RIL Format



RIE Format





z/Architecture Related Changes - 163 New Assembler Instructions

▲ 163 new assembler instructions -

- Use IPCS OPCODE command to decipher opcodes.

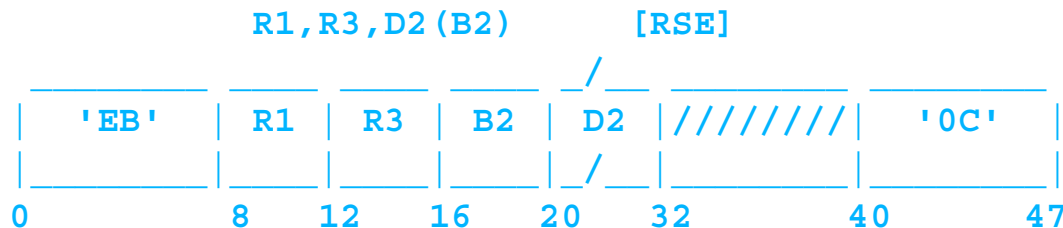
– Suggested to always provide 6 bytes of data to avoid possible confusion of OPCODE value

```

IPCS OUTPUT STREAM -----
Command ==> ip opcode EB345800C80C_  ← IPCS OPCODE COMMAND          SCROLL
***** TOP OF DATA *****
Mnemonic for X'EB345800C80C' is SRLG ← IPCS OPCODE RESULTS
***** END OF DATA *****
    
```

▲ (SRLG) SHIFT RIGHT SINGLE LOGICAL

- Mnemonics for 64Bit instructions contain a "G" (GRANDE)





Section 4: MVS Diagnostic Changes

Section 4: MVS Diagnostic Changes



TCB/RB Formatting Changes

▲ Select option 6 (COMMAND) from the IPCS Primary Option Menu

```
----- IPCS PRIMARY OPTION MENU -----
OPTION  ==> 6_  ←
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      - M030776
*  DATE        - 01/07/15
*  JULIAN      - 01.196
*  TIME        - 21:53
*  PREFIX      - M030776
*  TERMINAL    - 3278
*  PF KEYS    - 24
*****
```

Enter END command to terminate IPCS dialog



TCB/RB Formatting Changes

▲ Issue Command: SUMMARY FORMAT

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

==> SUMMARY FORMAT_ 

----- IPCS Subcommands and Abbreviations -----

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



TCB/RB Formatting Changes

▲ Issue Command: F 'TCB: '

- Repeat the find until the failing TCB is found vis TCB CMP field

```
IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> F 'TCB: ' _ ← CMD SCROLL ==> CSR
***** TOP OF DATA *****
COULD NOT ACCESS PSA      AT 00F8A000
COULD NOT ACCESS PSA      AT 00FBD000
COULD NOT ACCESS PSA      AT 00FB6000
COULD NOT ACCESS PSA      AT 00FAE000

* * * * F O R M A T * * * *

GLOBAL SERVICE MANAGER QUEUE
QUEUE IS EMPTY

LOCAL SERVICE MANAGER QUEUE
QUEUE IS EMPTY
```




TCB/RB Formatting Changes

IPCS OUTPUT STREAM ----- Line 3394 Cols 1 78

Command ==>

SCROLL ==> CSR

TCB: 007C92E8

+0000	RBP.....	007FD6C0	PIE.....	00000000	DEB.....	007970CC	
+000C	TIO.....	007B4000	CMP.....	840C4000	PLGS.....	00000000	
+0018	MSS.....	7F73F250	PKF.....	80	LLS.....	01000000	00
+0022	LMP.....	FF	DSP.....	FF			
+0028	JLB.....	007C9558	JPQ.....	007933F0			

Register values

0-3	0A460E1C	0078EF58	085A1018	7F696580
4-7	007C8000	00000000	00000001	7F696EB0
8-11	7F696990	00FBFA30	0078E000	0226F206
12-15	0A460E1C	0078EF78	50DB5F28	00EB18C0

64-Bit GPRs from TCB/STCB

0-1	00000000_0A460E1C	00000000_0078EF58
2-3	00000000_085A1018	00000000_7F696580
4-5	00000000_007C8000	00000000_00000000
6-7	00000000_00000001	00000000_7F696EB0
8-9	00000000_7F696990	00000000_00FBFA30
10-11	00000000_0078E000	00000000_0226F206
12-13	00000000_0A460E1C	00000000_0078EF78
14-15	00000000_50DB5F28	FFFFFFFF_00EB18C0


64-Bit GPRs are included with TCB Formatting




TCB/RB Formatting Changes

▲ Enter Command: F RB:

- This will take you to the first RB. Note the RBP field so you know when to stop

IPCS OUTPUT STREAM ----- Line 3393 Cols 1 78
 Command ==> F RB: _  SCROLL ==> CSR

TCB: 007C92E8

+0000	RBP.....	007FD6C0		00000000	DEB.....	007970CC	
+000C	TIO.....	007B4000	CMP.....	840C4000	TRN.....	00000000	
+0018	MSS.....	7F73F250	PKF.....	80	FLGS.....	01000000	00
+0022	LMP.....	FF	DSP.....	FF	LLS.....	007933D0	
+0028	JLB.....	007C9558	JPQ.....	007933F0			

Register values

0-3	0A460E1C	0078EF58	085A1018	7F696580
4-7	007C8000	00000000	00000001	7F696EB0
8-11	7F696990	00FBFA30	0078E000	0226F206
12-15	0A460E1C	0078EF78	50DB5F28	00EB18C0

64-Bit GPRs from TCB/STCB

0-1	00000000_0A460E1C	00000000_0078EF58
2-3	00000000_085A1018	00000000_7F696580
4-5	00000000_007C8000	00000000_00000000
6-7	00000000_00000001	00000000_7F696EB0
8-9	00000000_7F696990	00000000_00FBFA30



TCB/RB Formatting Changes

- Error RB

IPCS OUTPUT STREAM ----- FOUND: LINE 4092 COL 3
 Command ==> F RB: _ SCROLL ==> CSR

PRB: 007FE020

-0020	XSB.....	7FFFE10	FLAGS2...	80	RTPSW1...	478D0000
-0014		800EA39A	RTPSW2...	00040004	00000000	
-0008	FLAGS1...	02000004	WLIC.....	00040004		
+0000	RSV.....	00000000	00000000		SZSTAB...	00110082
+000C	CDE.....	007FAF08	OPSW.....	478D0000	800EA39A	
+0018	SQE.....	00000000	LINK.....	007C92E8		
+0020	GPR0-3...	FD000056	00006FA8	00000040	007D2CD4	
+0030	GPR4-7...	007D2CB0	007FAD90	007B3FF8	FD000000	
+0040	GPR8-11..	007FAF88	007F6108	00000000	007FAD90	
+0050	GPR12-15.	00E8F85A	00006F58	80E8FE2E	007FAFB8	

64-Bit GPRs from the RB/XSB


0-1	00000001_FD000056	00000000_00006FA8
2-3	00000000_00000040	00000000_007D2CD4
4-5	00000000_007D2CB0	00000000_007FAD90
6-7	00000000_007B3FF8	00000000_FD000000
8-9	00000000_007FAF88	00000000_007F6108
10-11	00000000_00000000	00000000_007FAD90
12-13	00000000_00E8F85A	00000000_00006F58
14-15	00000000_80E8FE2E	00000000_007FAFB8

64-Bit GPRs are included with the RB Formatting here. These registers are from the time the RB was first given control. Registers at the time of the interrupt are in the next RB. Enter "F RB:" to find those registers




TCB/RB Formatting Changes

- Registers for Error RB

IPCS OUTPUT STREAM ----- Line 4148 Cols 1 78
 Command ==> F RB:  SCROLL ==> CSR

```
SVRB: 007FD5C8
-0020 XSB..... 7FFFC2B0  FLAG52... 00      RTPSW1... 00000000
-0014      00000000  RTPSW2... 00000000  00000000
-0008 FLAGS1... 22010000  WLIC..... 00020033
+0000 RSV..... 00000000  00000000  SZSTAB... 001ED022
+000C CDE..... 00000000  OPSW..... 470C1000  8A583446
+0018 Q..... 00000000  LINK..... 007FE020
+0020 GPR0-3... 00000000  00121AD5  00121088  00000000
+0030 GPR4-7... 000E87B0  00121088  21E95498  40404040
+0040 GPR8-11.. FFFF0F0  000ADC38  000E77B0  000E95BE
+0050 GPR12-15. 000E773C  0002CB30  800EA02C  00000000
```

 64-Bit GPRs from the RB/XSB

Left halves of all registers contain zeros

```
0-3  00000000  00121AD5  00121088  00000000
4-7  000E87B0  00121088  21E95498  40404040
8-11 FFFF0F0  000ADC38  000E77B0  000E95BE
12-15 000E773C  0002CB30  800EA02C  00000000
```

Note that the 64-Bit GPRs are the same as the 32-bit GPRs and the left halves are not presented if they contain zeros. These are the registers from the time of the interrupt in the prior RB - in this case, the prior interrupt was an ABEND0C4 program check



TCB/RB Formatting Changes

- Last RB in the TCB/RB Flow

▲ This is the last RB in the TCB/RB flow

- TCBRBP from the TCB+x'00" contained 007FD6C0

IPCS OUTPUT STREAM ----- Line 4198 Cols 1 78
 Command ==> _ SCROLL ==> CSR

SVRB: 007FD6C0



-0020	XSB.....	7FFFC3A8	FLAGS2...	00	RTPSW1...	478C0000
-0014		8A42C45A	RTPSW2...	00020004	00000000	
-0008	FLAGS1...	02000000	WLIC.....	0002000C		
+0000	RSV.....	00000000	00000000		SZSTAB...	001ED022
+000C	CDE.....	00000000	OPSW.....	470C1000	8A45FAE2	
+0018	Q.....	00000000	LINK.....	007FD5C8		
+0020	GPR0-3...	81323EE8	7FF871B0	7F6B2518	7F73ACEC	
+0030	GPR4-7...	00F68540	0A58371F	8A582720	00F68580	
+0040	GPR8-11..	7F696EB0	000007B0	085A1018	0078CFC0	
+0050	GPR12-15.	7F73AB80	7F73AB80	085A1018	00000000	

64-Bit GPRs from the RB/XSB

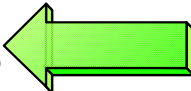
Left halves of all registers contain zeros

0-3	81323EE8	7FF871B0	7F6B2518	7F73ACEC
4-7	00F68540	0A58371F	8A582720	00F68580
8-11	7F696EB0	000007B0	085A1018	0078CFC0
12-15	7F73AB80	7F73AB80	085A1018	00000000



IPCS STATUS Output

▲ Select Option 2 (ANALYSIS) from the IPCS Primary Option Menu

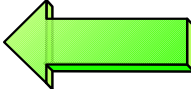
```
----- IPCS PRIMARY OPTION MENU -----  
OPTION  ==> 2_   
  
0  DEFAULTS      - Specify default dump and options  
1  BROWSE       - Browse dump data set  
2  ANALYSIS     - Analyze dump contents  
3  UTILITY      - Perform utility functions  
4  INVENTORY   - Inventory of problem data  
5  SUBMIT      - 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      - M030776  
* DATE        - 01/07/16  
* JULIAN      - 01.197  
* TIME        - 10:17  
* PREFIX      - M030776  
* TERMINAL    - 3278  
* PF KEYS     - 24  
*****
```

Enter END command to terminate IPCS dialog



IPCS STATUS Output

▲ Select Option 2 (STATUS) from the IPCS Analysis Menu

```
----- IPCS MVS ANALYSIS OF DUMP CONTENTS -----  
OPTION ==> 2_ 
```

To display information, specify the corresponding option number.

1	SYMPTOMS	- Symptoms	*****
2	STATUS	- System environment summary	* USERID - M030776
3	WORKSHEET	- System environment worksheet	* DATE - 01/07/16
4	SUMMARY	- Address spaces and tasks	* JULIAN - 01.197
5	CONTENTION	- Resource contention	* TIME - 10:22
6	COMPONENT	- MVS component data	* PREFIX - M030776
7	TRACES	- Trace formatting	* TERMINAL - 3278
			* PF KEYS - 24

Enter END command to terminate MVS dump analysis.



IPCS STATUS Output

▲ You will now see the output from the STATUS formatter

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

SYSTEM STATUS:

Nucleus member name: IEANUC01

I/O configuration data: Not Available

Sysplex name: RETEST

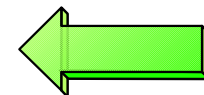
TIME OF DAY CLOCK: B6044DE4 00A5EE64 06/21/2001 00:54:06.598750 local

TIME OF DAY CLOCK: B6048388 E9A5EE64 06/21/2001 04:54:06.598750 GMT

Program Producing Dump: SYSMDUMP

Program Requesting Dump: #UNKNOWN

* * * DIAGNOSTIC DATA REPORT * * *



SEARCH ARGUMENT ABSTRACT



IPCS STATUS Output

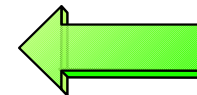
▲ The Diagnostic Data Report summarizes the error information

```
IPCS OUTPUT STREAM ----- Line 15 Cols 1 78
Command ==> _                SCROLL ==> CSR
                * * * DIAGNOSTIC DATA REPORT * * *
```

SEARCH ARGUMENT ABSTRACT

```
RIDS/#UNKNOWN#L RIDS/#UNKNOWN AB/S00C4 PRCS/00000004 REGS/0E36E REGS/0BDDC
```

Symptom	Description
-----	-----
RIDS/#UNKNOWN#L	Load module name: #UNKNOWN
RIDS/#UNKNOWN	Csect name: #UNKNOWN
AB/S00C4	System abend code: 00C4
PRCS/00000004	Abend reason code: 00000004
REGS/0E36E	Register/PSW difference for ROE: 36E
REGS/0BDDC	Register/PSW difference for ROB: DDC



SERVICEABILITY INFORMATION NOT PROVIDED BY THE RECOVERY ROUTINE



IPCS STATUS Output

▲ The Time of Error Information is also provided in the output

```
IPCS OUTPUT STREAM ----- Line 40 Cols 1 78
Command ==> _ SCROLL ==> CSR
Time of Error Information
```

```
PSW: 478D0000 800EA39A Instruction length: 04 Interrupt code: 0004
Failing instruction text: 926C92E8 30005850 92204140
```

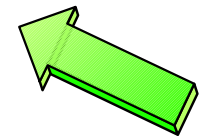
Registers 0-7

```
GR: 00000000 00121AD5 00121088 00000000 000E87B0 00121088 21E95498 40404040
AR: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Registers 8-15

```
GR: FFFFFFF0 000ADC38 000E77B0 000E95BE 000E773C 0002CB30 800EA02C 00000000
AR: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```
Home ASID: 00B8 Primary ASID: 00B8 Secondary ASID: 00B8
PKM: 00C0 AX: 0016 EAX: 0000
```



```
RTM was entered because of a program check interrupt.
The error occurred while an enabled RB was in control.
No locks were held.
No super bits were set.
```

The registers would have been 64-Bit if PSW was in 64-Bit mode

STATUS FROM THE RB WHICH ESTABLISHED THE ESTAE EXIT



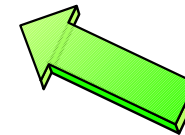
IPCS STATUS Output

IPCS OUTPUT STREAM ----- Line 92 Cols 1 78
Command ==> _ SCROLL ==> CSR

General purpose register values

Left halves of all registers contain zeros

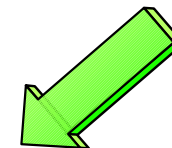
0-3	00000000	00121AD5	00121088	00000000
4-7	000E87B0	00121088	21E95498	40404040
8-11	FFFFFF0F0	000ADC38	000E77B0	000E95BE
12-15	000E773C	0002CB30	800EA02C	00000000



Access register values

0-3	00000000	00000000	00000000	00000000
4-7	00000000	00000000	00000000	00000000
8-11	00000000	00000000	00000000	00000000
12-15	00000000	00000000	00000000	00000000

The registers can be found in their extended form at the end of the IPCS STATUS output



Control register values

0-1	00000000_5F29FE50	00000001_ADBDC003
2-3	00000000_615F7D80	00000000_00C000B8
4-5	00000000_001600B8	00000000_01E58E00
6-7	00000000_FE000000	00000001_ADBDC003
8-9	00000000_00000000	00000000_20000000
10-11	00000000_20A0E158	00000000_20A0FFFE
12-13	00000000_7C7B5F0F	00000001_ADBDC003
14-15	00000000_DF881E53	00000000_7F6E9010



MVS System Trace Table

▲ Please note that the display of PSWs in the system trace table is not consistent. Some displayed PSWs are ESA/390-like while others display bit-12 off.

```

IPCS OUTPUT STREAM -----
Command ==> IP SYSTRACE ASID(x'116')_                               SCROLL ==> CSR
005A9A68 00
03-0116 005ACA68 DSP      070C0000 AB1C00B0 00000000 005ACA68 00598298
03-0116 005ACA68 SVC      23 07741000 80111FD8 80111FD8 00000000 000152E8
03-0116 005ACA68 SVC      78 07040000 841C1064 0000E500 00000000 00000000
03-0116 005ACA68 SVCR     78 070C0000 841C1064 00000000 00000000 00000000
03-0116 005ACA68 SVC      78 07042000 841C1190 0000E500 00000000 00000000
03-0116 005ACA68 SVCR     78 070C2000 841C1190 00000000 00000000 00000000
03-0116 005ACA68 SVC      38 07040000 841C3876 00000000 00000000 00000000
03-0116 005ACA68 PC        ... 0 00000000 00291EEC
03-0116 005ACA68 PT        ... 0 00000000 0129123C
03-0116 005ACA68 SVCR     38 070C1000 841C3876 00000000 00000000 01400116
03-0116 005ACA68 SVC        B 07042000 841C13EC 00000000 00000000 00000002
03-0116 005ACA68 SVCR        B 070C2000 841C13EC 00000000 18185560 0101198F
03-0116 005ACA68 PGM      011 070C2000 841C3D56 00040011 00585000

03-0116 005ACA68 SVC      3C 07041000 841C3E10 841C3D10 00000100 00585B70
03-0116 005ACA68 SVCR     3C 070C1000 841C3E10 00000000 00000000 00585B70
03-0116 005ACA68 PGM      011 070C2000 83DE29D4 00040011 00586000
    
```

The hardware PSW bit-12 is actually off in z/Architecture mode



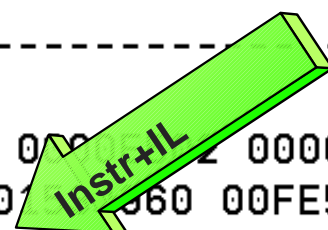
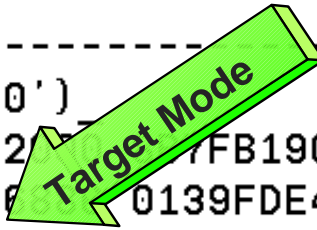
MVS System Trace Table

▲ The System Trace Table will show addressing mode changes.

- MODE trace entry PSWs point after the instruction (SAMxx - Set Address Mode, BSM -Branch and Set Mode, PC - Program Call) that caused the mode switch.
- MOBR trace entry PSWs contain the target branch address from the instruction (BASSM - Branch and Save and Set Mode or RP - Resume Program) that caused the mode switch

```

IPCS OUTPUT STREAM -----
Command ==> IP SYSTRACE ASID(x'1F0') SCROLL ==> C
00 01F0 008BDCF0 SVC 78 47042000 8B7FB190 00000000 00000068 00000000
00 01F0 008BDCF0 BR 014C0000 0139FDE4 014C0000 00FE590C 014F7398
00 01F0 008BDCF0 MODE ... 64 015353C2
00 01F0 008BDCF0 MODE ... 24 OR 31 00_015353E0
00 01F0 008BDCF0 MODE ... 64 015353F8
00 01F0 008BDCF0 MODE ... 24 OR 31 00_015353FE
00 01F0 008BDCF0 SVCR 78 470C2000 8B7FB190 00000000 00000068 7F80FF98
00 01F0 008BDCF0 SVC 38 47041000 8B7FD876 00000000 00000068 008B2FC8
00 01F0 008BDCF0 BR 011C5838 00FECF00 00FE5A0C 00FED000 011C4B50
00 01F0 008BDCF0 PC ... 0 011C4B8C 00101
00 01F0 008BDCF0 BR 011C6178 00FE5A0C 011C6990 09E01640 011C6A1A
00 01F0 008BDCF0 BR 011C707E 011C771E 00FE5F00 00FE26F8 011C5FC4
    
```





MVS System Trace Table

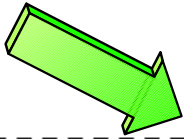
▲ Mode System Trace Table entry from prior page:

- 00 01F0 008BDCF0 MODE ... 64 015353C2
- Go to browse mode and select instruction address minus instruction length

ASID (X'01F0')	ADDRESS (015353C0.)	STORAGE	IS SAM64
Command ==>	OPCODE 010E		SCROLL ==> CSR
015353C0	010EE3E0	D2A00004	..T\K.....
015353D0	B40E3	B90B0011	.+.T....x9...\..
015353E0	E3	00045840	T.K.... K.x8.8..
015353F0	4300	14F0010E0.....1....
01535400	C018B2	00AA18A7	{..+...x..K... .
01535410	58B0D264	50B0A014	..K.&.... K.j...
01535420	A7E40006	9602A00A	xU..o...x4..m...
01535430	55580240	A77400AA	... x....xN...{.
01535440	A774007C	9108A010	x..@j...x....0..
01535450	582F04AC	58320008&.K....<
01535460	1233A774	000F5832	..x.....x...
01535470	58320018	5030D234&.K.x4..&.K.
01535480	5850004C	10550774	..x....

← IPCS CMD

Mode Instr





MVS Linkage Stack Entry

Command ==> IP SUMMARY FORMAT;F 'STACK ENTRY' _ ← **IPCS CMD** SCROLL ==>

LINKAGE STACK ENTRY 01 LSED: 7FFCE138

LSE: 7FFCE018

GENERAL PURPOSE REGISTER VALUES

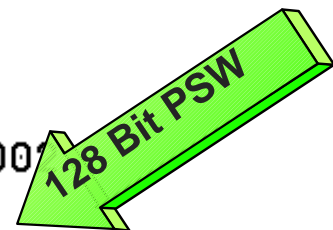
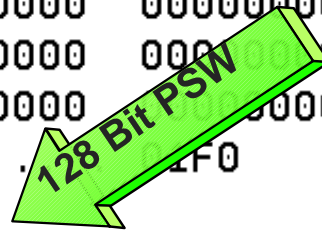
00-01....	00000000	0B82EFEE	00000000	3B6A5C40
02-03....	00000000	022EC310	00000000	01EC866C
04-05....	00000000	00000000	00000000	00000000
06-07....	00000000	00000040	00000000	01EC7678
08-09....	00000000	00F8ADD8	00000000	8B82E87C
10-11....	00000000	00000000	00000000	00000000
12-13....	00000000	00FBC480	00000000	7FFB9730
14-15....	00000000	8B82EFE8	00000000	8BEC0B60

ACCESS REGISTER VALUES

00-03....	00000000	00000000	00000000	00000000	
04-07....	00010004	00000000	00000000	00000000	
08-11....	00000000	00000000	00000000	00000000	
12-15....	00000000	00000000	00000000	00000000	
PKM.....	8040	SASN.....	00000000	EAX.....	00000000
PASN.....	01F0				
PSW.....	47042000	80000000		PSWE.....	00000000 0B82EFE8
TARG.....	00000000	8BEC0B84		MSTA.....	00000000 00000000
TYPE.....	8C				



Linkage stack entries will contain the full 128-Bit PSW and 64-Bit GPRs





Section 5: Verify Buffers Backed Above the Bar

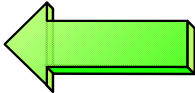
Section 5:

Verify Buffers Backed Above the Bar



Finding "Above the Bar" Log Buffers

▲ Select Option 2 (ANALYSIS) from the IPCS Primary Option Menu

```
----- IPCS PRIMARY OPTION MENU -----  
OPTION ==> 2_   
  
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   - M030776  
* DATE     - 01/07/18  
* JULIAN   - 01.199  
* TIME     - 12:59  
* PREFIX   - M030776  
* TERMINAL- 3278  
* PF KEYS  - 24  
*****
```

Enter END command to terminate IPCS dialog



Finding "Above the Bar" Log Buffers

▲ Select Option 6 (COMPONENT) from the IPCS Analysis Menu

```
----- IPCS MVS ANALYSIS OF DUMP CONTENTS -----  
OPTION  ==> 6_  ←
```

To display information, specify the corresponding option number.

1	SYMPTOMS	- Symptoms	*****
2	STATUS	- System environment summary	* USERID - M030776
3	WORKSHEET	- System environment worksheet	* DATE - 01/07/18
4	SUMMARY	- Address spaces and tasks	* JULIAN - 01.199
5	CONTENTION	- Resource contention	* TIME - 13:06
6	COMPONENT	- MVS component data	* PREFIX - M030776
7	TRACES	- Trace formatting	* TERMINAL - 3278
			* PF KEYS - 24

Enter END command to terminate MVS dump analysis.



Finding "Above the Bar" Log Buffers

▲ Select DFSAAMPR from the IPCS Component Analysis Menu

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

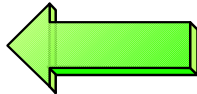
To display information, specify "S option name" or enter S to the left of the option desired. Enter ? to the left of an option to display help regarding the component support.

<u>S</u>	<u>Name</u>	<u>Abstract</u>
_	COUPLE	XCF Coupling analysis
_	CTRACE	Component trace summary
_	DAE	DAE header data
_	DB2DATA	DB2 analysis
<u>s</u>	<u>DFSAAMPR</u>	IMS Interactive Dump Formatter
_	DIVDATA	Data in virtual storage
_	DLFDATA	Data Lookaside Facility data
_	DLFTRACE	Data Lookaside Facility trace
_	ELXDATA	IMS Transport Manager Formatting
_	GRSDATA	ENQ/DEQ resources
_	IOSCHECK	Active input/output requests
_	IPCSDATA	IPCS control data
_	IRLM	IMS Resource Lock Manager analysis



Finding "Above the Bar" Log Buffers

▲ Select option 0 (INIT) from the IMS Dump Formatting Primary Menu

```
----- IMS DUMP FORMATTING PRIMARY MENU -----  
OPTION  ==> 0_   
  
 0  INIT          - IMS formatting initialization and content summary  
 1  BROWSE       - Browse Dump dataset (IPCS norm)          *****  
 2  HI-LEVEL    - IMS Component level formatting          *USERID   - M030776  
 3  LOW-LEVEL   - IMS ITASK level formatting              *DATE     - 01/07/18  
 4  ANALYSIS    - IMS dump analysis                      *JULIAN   - 01.199  
 5  USER        - IMS user formatting routines           *TIME     - 13:15  
 6  OTHER COMP  - Other IMS components (BPE, CQS...)      *PREFIX   - M030776  
 7  OTHER PROD  - Other IMS-related products             *TERMINAL - 3278  
 E  EDA         - IMS Enhanced Dump Analysis            *PF KEYS  -  
 T  TUTORIAL    - IMS dump formatting tutorial           *****  
 X  EXIT        - Exit IMS dump formatting  
  
Enter END or RETURN command to terminate IMS component formatting.  
Use PFKeys to scroll up and down if needed.
```

```
* THIS PRODUCT CONTAINS "RESTRICTED MATERIALS OF IBM". 5655-B01 (C) *  
* COPYRIGHT IBM CORP. 1991,2000 LICENSED MATERIALS - PROPERTY OF IBM. *  
* ALL RIGHTS RESERVED. U.S. GOVERNMENT USERS RESTRICTED RIGHTS - USE *  
* DUPLICATION, OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT *  
* WITH IBM CORP. REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083. *
```



Finding "Above the Bar" Log Buffers

Initialize with the correct IMS JOBNAME in the CTL field

```
----- IMS DUMP CONTENT STATUS AND CONTROL -----  
COMMAND ==>
```

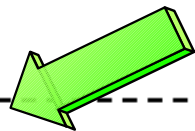
Enter the IMS CTL/BATCH or DL/I jobname to cause the IMS symbols to be set for this dump. Request subsystem list for possible IMS names.

```
N <==== IMS SUBSYSTEM LIST DESIRED? (Y or N)  
N <==== FORMATTER REFRESH? (Y or N)
```

	JOBNAME	ID	ASID	DUMPED?
CTL	IMS71RP1	IMS1	0116	YES
DL/I	DLIERES		0118	YES
DBRC	DBRCEORS		0119	YES
IRLM	IRLMC1	IRL1	0115	YES
TMS		-		

ABEND CODE = SYS 0C3 USER 0
MODULE = UNKNOWN

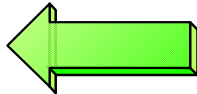
IMS SDWA ADDRESS - 00584918 IMS RELEASE - 710





Finding "Above the Bar" Log Buffers

▲ Select option 2 (HIGH-LEVEL) from the IMS Dump Formatting Primary Menu

```
----- IMS DUMP FORMATTING PRIMARY MENU -----  
OPTION  ==> 2_   
  
0  INIT          - IMS formatting initialization and content summary  
1  BROWSE        - Browse Dump dataset (IPCS norm)          *****  
2  HI-LEVEL      - IMS Component level formatting          *USERID   - M030776  
3  LOW-LEVEL     - IMS ITASK level formatting              *DATE     - 01/07/18  
4  ANALYSIS      - IMS dump analysis                      *JULIAN    - 01.199  
5  USER          - IMS user formatting routines           *TIME     - 13:20  
6  OTHER COMP    - Other IMS components (BPE, CQS...)       *PREFIX   - M030776  
7  OTHER PROD    - Other IMS-related products              *TERMINAL - 3278  
E  EDA           - IMS Enhanced Dump Analysis             *PF KEYS  -  
T  TUTORIAL      - IMS dump formatting tutorial                          *****  
X  EXIT          - Exit IMS dump formatting  
  
Enter END or RETURN command to terminate IMS component formatting.  
Use PFKeys to scroll up and down if needed.
```

```
* THIS PRODUCT CONTAINS "RESTRICTED MATERIALS OF IBM". 5655-B01 (C)      *  
* COPYRIGHT IBM CORP. 1991,2000 LICENSED MATERIALS - PROPERTY OF IBM.   *  
* ALL RIGHTS RESERVED. U.S. GOVERNMENT USERS RESTRICTED RIGHTS - USE    *  
* DUPLICATION, OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT     *  
* WITH IBM CORP. REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083.  *  
* * * * *
```



Finding "Above the Bar" Log Buffers

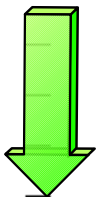
▲ Select LOG formatting from the IMS High-Level Dump Formatting Menu

```
----- IMS HIGH LEVEL DUMP FORMATTING OPTIONS Row 1 to 14 of 30
Command ==> Scroll ==> PAGE
```

```
N <====SPOOL OUTPUT? (Y or N)          N <====REFRESH FORMATTER? (Y or N)
      S = select      M = select,min      select choice(s) and hit enter
                                          to process or UP/DOWN to scroll
```

Additional IMS format requests==>

Cmd	Option	Description
—	AUTO	Internally determined options (by failing ITASK type)
—	ALL	All high level IMS dump formatting options
—	SUMMARY	PSW, regs, SAP, failing ITASK blocks at time of abend
—	SCD	SCD, SLX, FP ESCD, scheduler sequence queues
—	SAVEAREA	SAP, savearea, ECB prefix, UEHB (sorted by DSPNO)
—	DISPATCH	Dispatcher work areas, Dispatcher and Latch traces
—	SPST	System PSTs and subordinate blocks
—	RESTART	CHKPT ID table, SIDX, LCRE, RPST, RRE, EQEL, IEEQE, FRB
s	LOG	LCD, log buffer prefixes, log buffers (OLDS and MON)
=	DB	DDIRs, PDIRs, intent list, DLI/LOCK traces, DPSTs, DBT
=	DEDB	ALDS, DMCB, DMAC, XCRB, SRB, ESRB, FPT blocks





Finding "Above the Bar" Log Buffers

▲ Enter Command: F 'OLDS BUFFER'

```
IPCS OUTPUT STREAM ----- Line 0 Cols 1 78
Command ==> f 'OLDS BUFFER' _          SCROLL ==> CSR
***** TOP OF DATA *****
```



**FYI...For those of you familiar with IMS formatting commands,
we could have received this formatting through command:
IP VERBX IMSDUMP 'IMS71RP1 FMTIMS LOG'**

```
* : * * * * *
* * * * * I M S   C T L   R E G I O N   F O R M A T T I N G
**IMS
* * * * *
* * * * *
```

DATE= 2001/199 TIME= 13.32.27

EYECATCHER AREA

```
**LCD      LOG CONTROL DIR.
**LCDM     DC-MONITOR LCD
**LGT      DLOG TRACE
```




Finding "Above the Bar" Log Buffers

▲ Enter Command: F 'OLDS BUFFER' . Note Olds Buffer Address X'BABC00'

IPCS OUTPUT STREAM ----- FOUND: LINE 285 COL 4
Command ==> SCROLL ==> CSR

***OLDS BUFFER PREFIX (LBUFFER) AT 00BABC00



*LBUFFER	-	FCHAIN	00BABD00	FSCHN	00000000	FLAGS	80000000
		LCD	00BBA000	BUFFA	2AD4B000	SAVEA	001DA5E0
		BLKCNT	00000000	BLKOFF	00000000	DMYLOG	00000000
		RESV	00000000	AWEQ	00000000	AWET	00000000

LBECBP1

00030	0FC4E2D7	00000000	00000000	00000000	*.DSP.....*
00040	00030909	2B01F150	001BE148	2AF58120	*.....1.....5a.*
00050	00000000	00000000	00000000	00000000	*.....*
LINES	00060-0007F	SAME AS THE ABOVE			

LBECBP2-LBRESVW

00080	0FC4E2D7	00000000	00000000	00000000	*.DSP.....*
00090	00030909	2B01F150	001BE148	2AF58120	*.....1.....5a.*
000A0	00000000	00000000	00000000	00000000	*.....*
LINES	000B0-000EF	SAME AS THE ABOVE			

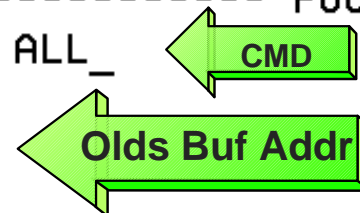


Finding "Above the Bar" Log Buffers

▲ Enter Command: IP RSMDATA VIRTPAGE RA(00BABC00) ALL

IPCS OUTPUT STREAM ----- FOUND: LINE 285 COL 4
Command ==> IP RSMDATA VIRTPAGE RA(00BABC00) ALL_ CMD SCROLL ==> CSR

***OLDS BUFFER PREFIX (LBUFFER) AT 00BABC00



```
*LBUFFER      - FCHAIN 00BABD00  FSCHN  00000000  FLAGS  80000000
                  LCD      00BBA000  BUFFA  2AD4B000  SAVEA  001DA5E0
                  BLKCNT  00000000  BLKOFF 00000000  DMYLOG 00000000
                  RESV    00000000  AWEQ   00000000  AWET   00000000
```

LBECBP1

```
00030  0FC4E2D7 00000000 00000000 00000000  *.DSP.....*
00040  00030909 2B01F150 001BE148 2AF58120  *.....1.....5a.*
00050  00000000 00000000 00000000 00000000  *.....*
LINES  00060-0007F  SAME AS THE ABOVE
```

LBECBP2-LBRESVW

```
00080  0FC4E2D7 00000000 00000000 00000000  *.DSP.....*
00090  00030909 2B01F150 001BE148 2AF58120  *.....1.....5a.*
000A0  00000000 00000000 00000000 00000000  *.....*
LINES  000B0-000EF  SAME AS THE ABOVE
```



Finding "Above the Bar" Log Buffers

▲ RSM Virtual Page Report shows "Above the Bar:" real address in "R LOC" field

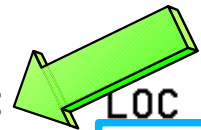
IPCS OUTPUT STREAM ----- Line 12 Cols 1 78
Command ==> _ SCROLL ==> CSR

R S M V I R T U A L P A G E R E P O R T

JOBNAME	DSP	NAME	PAGE	G	K	F	R	P	D	B	L	STAT	T	R	LOC	LOC	LOC2
PAGECOMM	-		00BAB000	Y	7	Y	E	N	N	N	N	REAL	V		0014288C		

Totals (in decimal) for job COMMON:

REAL	AUX	VIO
1	0	0
DSN	FREF	HIDE
0	0	0
SWAX		
0		
SOAI	SIAI	
0	0	



The "R LOC" field presents the real page address. Three zeros must be added to the end of the field.
Real Page Location =
x'1_4288C000'
Central storage to 128 GB =
x'20_00000000'



Section 6: ABEND0C4 RC038 Example

Section 6 ABEND0C4 RC38 Example



ABEND0C4 RC038 Example

Program-Interruption Conditions	Hex PIC	Address mode	ABEND Code	Reason
ASCE (Address Space Control Element) -Type *	38	64Bit	0C4 : z/OS 1.2 and above	Branch to virtual address above the bar

Problem: SYSTEM COMPLETION CODE=0C4 REASON CODE=00000038 ABEND0C4

TIME=11.13.37 APRIL 30 SEQ=46373 CPU=0000 ASID=0152

PSW AT TIME OF ERROR 070C1001 B45BC397 ILC 4 INTC 38

ACTIVE LOAD MODULE ADDRESS=345BC158 OFFSET=000002

NAME=DFSAERE0

DATA AT PSW 345BC391 - F0C23E0D EF185F12 BB4780C2

```
GR 0: 00000010    1: 345BC428
    2: 0000BAE8    3: 00000000
    4: 00000000    5: 00000000
    6: 345BC41C    7: 00B2FD80
    8: 0085F918    9: 00000000
    A: 0000000C    B: 0085FFB8
    C: B45BC158    D: 0000BAF8
    E: B45BC397   F: 00000000
```



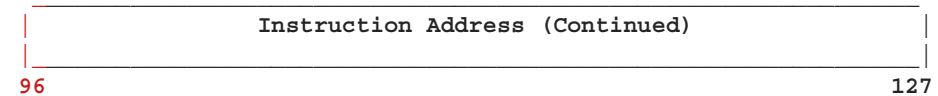
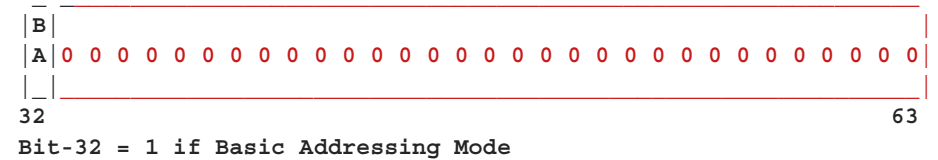
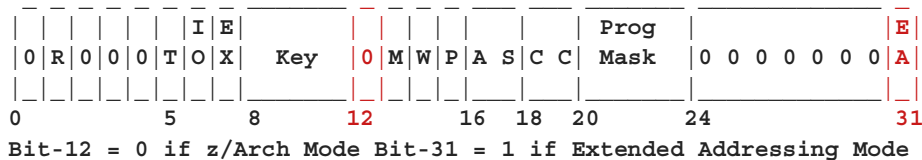
ABEND0C4 RC038 Example, Continued

▲ Error due to branch to address above the bar

- PSW = 070C1001 B45BC397 PIC=038

Bit-31	Bit-32	Addressing Mode
0	0	24-bit
0	1	31-bit
<u>1</u>	<u>1</u>	64-bit
1	0	Specification Exception (PIC 6)

Because we are in EA Mode, the instruction address being translated is B34BC396, which is above the line, resulting in PIC 38





ABEND0C4 RC038 Example, Continued

▲ Error indicates branch to address above the bar

- PSW = 070C1001 B45BC397
- Register 14 = B45BC397

Code from module called just prior to 345BC396 follows:

Low order bit on indicates Extended Addressing (64Bit) Mode is to be entered for BSM instruction

```
LEAVE RC= (15) , LINKAGE=DEP , CHAIN=NO , CAP=RET
DS      0H
LR      15, (15)          SET RETURN CODE.
L       14, 12 (13)      A (RETURN TO CALLER)
LM      0, 12, 20 (13)   RESTORE REGS
OI      15 (13) , X'01'  SHOW RETURNED
BSM    0, 14           RETURN TO CALLER WITH MODE
```

Attempted to return to Module DFSAERE0 using a save area that has already been marked as residual (low order bit set on) by another task. Double use of save area - See APAR PQ76555. Would have been ABEND0C6 in ESA/390 mode.



Section 7: Appendix - IMS z/Architecture Related APARs

Section 7: Appendix IMS z/Architecture Related APARs



z/Architecture Related IMS APARs

- ▲ IMS will include keyword "64BITIMS" in APARs that involve z/Architecture software or hardware changes



z/Architecture Related IMS APARs, Continued

▲ ABENDU0762-08 AFTER EXECUTION OF THE LASP COMMAND DURING ISWITCH TO=XMDLI

– PQ91414/UQ91436 - R710 PDO 0433 ESO F ____

– PQ90002/UQ91194 - R810 PDO ____ ESO F ____

- **ERROR DESCRIPTION:** IMS abends because it receives condition code 1 or condition code 2 from the LASP instruction when it tries to set a new cross-memory state. It is receiving a bad condition code because it is passing an 8-byte area as the first argument to LASP. However, with ALRF (ASN-and-LX-reuse facility) enabled, the definition of the LASP instruction has changed, and LASP expects a 16-byte area for the first argument. The 8-byte area and the following 8 bytes passed by IMS are not in the required format for this new LASP instruction, and LASP fails either the primary or the secondary ASN translation.
- **USERS AFFECTED:** IMSFP R710/R810 under z/OS 1.6 with APAR OA07708 on a 2084 (z/990) or 2086 (z/890) at Driver 55 (and follow-on machines) which support the ASN-and-LX-reuse facility.
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

- ▲ **ABENDS0D3 PIC13 IN DBFVOCIO PREIO1B IN ABENDS0D3 PIC13 IN DBFVOCIO PREIO1B**
 - PQ78665/UQ80806 - R710 PDO 0342 ESO F310
 - PQ78716/UQ82876 - R810 PDO 0401 ESO F312
- **ERROR DESCRIPTION:** LRA instruction references beyond last page fixed WSA buffer.
 - In the z/Architecture environment, when pageable storage is getmained, it can be backed anywhere in real storage. The LRA instruction cannot handle translation of the 64-Bit real address if bits 0-32 of the real address are not all zeros, a special-operation exception (PIC 13) is recognized.
- **USERS AFFECTED:** IMSFP R710/R810 DEDB Local VSO users with z/OS running in z/Architecture mode.
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

▲ Various IMS control region abends and waits when running in z/Architecture mode PEd APAR/PTF PQ60202/UQ67132

- PQ63384/UQ67132 - R710 PDO 0242 ESO F210
- PQ73981/UQ78487 - R710 PDO 0330 ESO F307
- PQ65596/UQ70434 - R810 PDO 0241 ESO F210
- PQ74282/UQ78488 - R810 PDO 0330 ESO F307
- **ERROR DESCRIPTION:** IMS Control region ABENDU0403 in DFSAOS70, wait for I/O completion DFSAOS80, ABEND0C4 IOSVIRBA DFSAOS70, ABENDS0D3 after application termination. Due to problems with construction of CCW chains and related control blocks because of size changes resulting in overlays.
- **USERS AFFECTED:** All IMS R710 Users with PEd APAR/PTF PQ60202/UQ67132 (64-Bit abends and waits OSAM and Logger - PDO 0227 ESO F207) installed and running in z/Architecture mode
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

- ▲ **64-Bit exploitation for OSAM sequential buffers**
 - **PQ52333/UQ76745 - R710 PDO 0324 ESO F306**
 - **PQ66132/UQ76547 - R810 PDO 0321 ESO F305**
- **IMS sequential buffering exploitation of 64-Bit real storage.**
- **Virtual storage for these buffers remain below the 2GB bar and operate in 31-bit mode**



z/Architecture Related IMS APARs, Continued

▲ Abend0C4 in DFSUT060 running an MFSGEN in 64-Bit Mode

– PQ67953/UQ71988 - R710 PDO 0248 ESO F211

– PQ68031/UQ72143 - R810 PDO 0249 ESO F211

- **ERROR DESCRIPTION:** Abend0C4 in DFSUT060 running an MFSGEN batch job in 64 bit ESAME mode. Picked up value from low core which is non-zero in z/Architecture mode.
- **USERS AFFECTED:** ALL MFS users compiling MFS formats defined without a DPAGE statement in a 64-bit machine.
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

▲ ABENDU0403 IN DFSAOS70 AT +X'4C2' WHEN RUNNING 64-BIT

– PQ57234/UQ64497 - R710 PDO 0214 ESO F204

- **ERROR DESCRIPTION:** ABENDU0403 because global IOMA does not match local IOMA. Global IOMA pointer to local IOMA is overlaid with IDAL control blocks
- **USERS AFFECTED:** All IMS710 users of OSAM data bases in z/Architecture mode
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

▲ ABEND0D3 RC13 IN DFSVSPL0 +X'57E' WHEN OPENING VSO PRELOAD AREA.

– PQ57983/UQ64591 - R610 PDO 0219 ESO F205

– PQ58447/UQ64726 - R710 PDO 0216 ESO F204

- **ERROR DESCRIPTION:** The algorithm for page fixing the pages of the dataspace as the UOWs are processed fails. LRA instruction receives the ABEND0D3 RC13.
- **USERS AFFECTED:** IMSFP DEDB VSO running z/OS or S/390 R10 ARCHLVL=2 in z/Architecture mode
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

▲ 64-Bit exploitation for fastpath buffer pool

- PQ51419/UQ62396 - R710 PDO 0218 ESO F204 - **PE**
- PQ62953/UQ68629 - R710 PDO 0233 ESO F208
- IMS fastpath buffer pool exploitation of 64-Bit real storage
- Virtual storage for these buffers remain below the 2GB bar and operate in 31-bit mode. LOC=(31,64) for IMODULE GETMAIN



z/Architecture Related IMS APARs, Continued

- ▲ Various IMS control region abends and waits when running in z/Architecture mode PEd APAR/PTF PQ42127/UQ52990
 - PQ60202/UQ67132 **PE** - R710 PDO 0227 ESO F207
 - PQ63384/UQ67132 - Various control region waits and abends
- **ERROR DESCRIPTION:** IMS Control region ABENDU0403 in DFSAOS70, wait for I/O completion DFSAOS80, ABEND0C4 IOSVIRBA DFSAOS70, ABENDS0D3 after application termination. Due to problems with construction of CCW chains because of size changes resulting in overlays.
- **USERS AFFECTED:** All IMS R710 Users with PEd APAR/PTF PQ42127/UQ52990 (64-Bit exploitation for OSAM and Logger - PDO 0121 ESO F105) installed and running in z/Architecture mode
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

▲ Emergency Restart Ignores WADS Data when not Running in z/Architecture Mode with PEd APAR/PTF PQ42127/UQ52990

– PQ52887/UQ59347 - R710 PDO 0144 ESO F111

- **ERROR DESCRIPTION:** Users may lose log data during emergency restart. Restart will ignore WADS data when closing the OLDS. It reads the WADS but decides that the data is residual. (Exclusion: If DFS0616I OLDS BUFFERS SUCCESSFULLY PURGED is received at previous execution of XRF capable system).
- **USERS AFFECTED:** All IMS R710 Users with PEd APAR/PTF PQ42127/UQ52990 (64-Bit exploitation for OSAM and Logger - PDO 0121 ESO F105) installed and not running in z/Architecture mode.
- **RECOMMENDATION:** Install corrective PTF.



z/Architecture Related IMS APARs, Continued

▲ 64-Bit exploitation for OSAM and Logger

- PQ42127/UQ52990 **PE** - R710 PDO 0121 ESO F105
- PQ52887/UQ59347 - Log Data Loss During /ERE
- PQ60202/UQ67132 - Various Control Region Abends and Waits
- Buffers for OSAM databases and for IMS logs are page fixed in real storage above the 2GB bar if IPLed in z/Architecture mode.
- Virtual storage for these buffers remain below the 2GB bar and operate in 31-bit mode.



z/Architecture Related IMS APARs, Continued

▲ Performance problems with Log Archive utility used on D/T2064 zSeries processor

- PQ48811/UQ54912 - R510 PDO 0125 ESO F106**
- PQ46887/UQ54046 - R610 PDO 0120 ESO F105**
- PQ48050/UQ54047 - R710 PDO 0121 ESO F105**
- The IMS Log Archive Utility contains numerous instances of variables defined in close proximity to instructions that modify them. If the distance between variable and instruction is less than the amount of cached storage for instruction prefetch, the modification invalidates the cache, causing it to be refreshed.**
- Local subroutine variables have been moved to the general data area.**



z/Architecture Related IMS APARs, Continued

▲ **IMS V7 CQS loop calling IXLMG with ABEND0C4s in module IXLA1MG filling
SYS1.LOGREC**

– PQ45860/UQ51962 - R710 PDO 0110 ESO F103

- **Users affected are those with more than two links defined between the machine on which CQS is running and any coupling facility containing a CQS list structure, or**
- **Both the machine on which CQS is running and any coupling facility containing a CQS list structure or D/T2064 with any number of links.**



z/Architecture Related IMS APARs, Continued

▲ BPE Dump formatting access problems

- PQ44349/UQ50125 - R710 PDO 0103 ESO F101 - CQS**
- PQ44351/UQ50529 - R110 PDO 0105 ESO F102 - IMS Connect**
- PQ44351/UQ50244 - R110 PDO 0103 ESO F101 - ORS**
- OS/390 V2R10 and above IPCS changes affected the EVALSYM function which now returns a 16 byte value instead of an 8 byte value. Formatting fails when a dump formatting option is attempted.**



z/Architecture Related IMS APARs, Continued

- ▲ **ABEND0C4 in DFSRCP30 due to accessing PSA data in z/Architecture mode.
Also could result in overlay of CSA.**
 - **PQ42981/UQ48747 - R510 PDO 0048 ESO F011**
 - **PQ42917/UQ48701 - R610 PDO 0048 ESO F011**
 - **PQ42937/UQ48746 - R710 PDO 0048 ESO F011**
- **This failure was recognized in z/Architecture mode because the layout of the PSA has changed. What was a high value in ESA/390 mode, is a low value in z/Architecture mode. In ESA/390 mode, an invalid length was used rather harmlessly. Not so in z/Architecture mode.**



z/Architecture Related IMS APARs, Continued

▲ ABEND0D3 PIC x'13' on LRA (Load Real Address) instruction

- PQ41295/UQ48302 - R510 PDO 0045 ESO F011**
- PQ42408/UQ48303 - R610 PDO 0045 ESO F011**
- PQ42409/UQ48304 - R710 PDO 0046 ESO F011**
- In the z/Architecture environment, when pageable storage is getmained, it can be backed anywhere in real storage. The LRA instruction cannot handle translation of the 64-Bit real address if bits 0-32 of the real address are not all zeros, a special-operation exception (PIC 13) is recognized.**
- TPROT instruction substituted to check if page is loaded into real**



IMS Debugging in a 64-Bit World - Summary

▲ Summary

- **64-Bit Terminology**
- **Recognizing z/Architecture in Dumps**
- **z/Architecture Related Changes**
- **MVS Formatting Changes**
- **Verify Buffers Backed Above the Bar**
- **ABEND0C4 RC38 Example**
- **Appendix: IMS z/Architecture Related APARs**