

How to Help IBM **AND YOU** Quickly Resolve IMS Problems.

Steve Nathan – snathan@us.ibm.com

Disclaimer

© Copyright IBM Corporation 2024. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and IMS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Agenda

- **Before the Problem Occurs**
- **While the Problem is Happening**
- **After the Problem – YOU**
- **After the Problem – Helping IBM**
- **Appendix – IMS Connect Recorder Trace Details**

Introduction

Abstract

We all know IMS is perfect. But on the rare occasion when there is a problem is must be resolved quickly. This presentation will document how to make that happen.

Introduction

▪ Acknowledgements

- Much of the setup information was prepared and presented several times by Jeff Maddix
 - “Making It Difficult for IMS Problems to Hide”
- Thanks to Kevin Stewart for explaining TIMEOUT and LOCKTIME and DEADLOK

Introduction

- **This presentation was prepared to show how to resolve those RARE times when there is a problem**
 - Setting up the z/OS and IMS environments properly first can help when there is a problem
 - You can take actions while the problem is happening
 - You can take actions to resolve the problem yourself
 - You can gather and prepare documentation that will help IMS support to resolve the problem most efficiently

Before the Problem Occurs

Before the Problem Occurs

- **Have the proper tools**

- You must have and be very familiar with one from each set
- Monitor IMS
 - BMC Mainview
 - Omegamon for IMS
 - TMON for IMS
- Analyze CPU
 - BMC STROBE
 - IBM Application Performance Analyzer

Before the Problem Occurs

- **Have the proper tools**

- Analyze logs and traces

- BMC AMI Log Analyzer for IMS
- IMS Performance Analyzer and IMS Problem Investigator

- Analyze IMS Connect

- BMC Energizer
- IMS Connect Extensions

Before the Problem Occurs

- **Become familiar DFSERA10**
 - File Select and Formatting Print Utility
 - [DFSERA10](#)
 - Can select records based on offset and value
 - Can select ALL records with string ABC or x'C1C2C3'
 - Can select records based on bit settings on or off
 - Can select records based on common data fields

Before the Problem Occurs

- **Become familiar DFSERA10**
 - Can be used to **PRINT** or **COPY**
 - Can be used with **ANY** file
 - Not just IMS logs
 - Including selection
 - Has STARTAFT and STOPAFT so you can print/copy a selected part of a file
 - Use STOPAFT=EOF to process the whole file if more than 16M records
 - The default STOPAFT is 16,777,215

Before the Problem Occurs

- **Become familiar DFSERA10**

- Exit routine DFSERA70 allows specialized record selection

PST=pst_number

Selects records for the PST number.

SYSID=system_id

Selects records for the system ID portion of recovery token.

TOKEN=token

Selects records for the hexadecimal token portion of recovery token.

PSB=psb_name

Selects records for the PSB name.

DBD=dbd_name

Selects records for the DBD name.

RBA=rba_value

Selects records for the RBA (lrecl).

BLOCK=block_rba

Selects records for the RBA (block).

Before the Problem Occurs

■ **Become familiar DFSERA10**

- Exit routine DFSERA70 allows specialized record selection

USERID=userid

Selects records for the userid.

KEY=ksds_key

Selects records for the key.

OFFSET=offset

Selects records that update a given offset of data in the buffer.

UNDO=undo_data

Selects records for backout data that matches the character string you specify.

The maximum length of the character string is 255 characters.

REDO=redo_data

Selects records with recovery data that matches the character string you specify.

The maximum length of the character string is 255 characters.

DATA=log_data

Selects records with data, including compressed data, anywhere in the record .

The maximum length of the character string is 255 characters.

Before the Problem Occurs

- **Become familiar with DFSDDLTO**
 - DL/I Test Program
 - [DFSDDLTO](#)
 - Run as batch or BMP
 - Make IMS DB and DC calls from SYSIN control cards
 - Issue any valid DL/I call against any database using
 - Any SSA or PCB, or both
 - Any SSA or AIB, or both
 - Repeat each call up to 9,999 times
 - Compare the results of a call to expected results
 - Print data only when the output is useful, such as after an unequal compare

Before the Problem Occurs

▪ **Become familiar with DFSDDLTO**

- Dump DL/I control blocks, the I/O buffer pool, or the entire region
- Punch selected control statements to an output file to create new tests
- Merge multiple input data sets into a single input data set
- Send messages to the z/OS® system console
 - With or without a reply
 - Great for running 2 tests - one of which depends on the other
 - Start A - WTOR to Hold A - Run B - Reply to WTOR and A continues

Before the Problem Occurs – z/OS

- **Ensure the z/OS System Trace Table is large enough to capture events**
 - The default is 1M (per CPU)
 - The recommendation is 3M
 - The z/OS command “TRACE ST,3M” can be specified in the z/OS COMMNDxx SYS1.PARMLIB member
 - The DISPLAY TRACE command will show current values

```
IEE843I 13.17.07 TRACE DISPLAY 479
      SYSTEM STATUS INFORMATION
      ST=(ON,3000K,12000K) AS=ON BR=OFF EX=ON MO=OFF MT=(ON,1000K)
```


Before the Problem Occurs – z/OS

- **If you want to have some fun practice reading the SYSTRACE from an SVC dump**
 - Command IP SYSTRACE TIME(LOCAL)
 - Here are some links to help you
 - [MVS Diagnosis: Tools and Service Aids](#)
 - [Reading system trace output](#)

Before the Problem Occurs – z/OS

■ SYSTRACE example

```

----- System Trace Table -----
--
--
PR  ASID WU-Addr- Ident  CD/D PSW----- Address- Unique-1 Unique-2 Unique-3 PSACLHS- PSALOCAL PASD SASD Time Local----- CP--
                                Unique-4 Unique-5 Unique-6 PSACLHSE PSALOCAL PASD SASD Date-03/21/2024
00D7 07C8AF00 SRB      00000000_0133D368 000000D7 07768600 0776862C      00          00D7 00D7 11:55:55.272696096 0003
                                47040000 80000000 009FE828 80
0000 00D7 07C8AF00 SSRV   2          80FEA67A 009D7DD0 7F000000 00000000 Post          11:55:55.272696943 0003
                                00000000
0000 00D7 009FE828 DSP    00000000_30E2E7C2 00000000 00000001 009D7DD0 00000000 00000000 00D7 00D7 11:55:55.272697414 0003
                                47043000 80000000
0000 00D7 009FE828 SSRV   78          B0E2F848 2010E670 00000F78 7FF5D088 Getmain          11:55:55.272698727 0003
                                00D70000
0000 00D7 009FE828 SVC     0 00000000_30E2E7B0 30F31600 009D7E10 009D7DE8 Excp          11:55:55.272699212 0003
                                47043000 80000000
0000 00D7 009FE828 SSCH 020C7 00 02 0776722C 0269F070 03C2FF81 7E435DE8          11:55:55.272704190 0003
0000 00D7 009FE828 SVCR   0 00000000_30E2E7B0 00000000 0269F070 009D7DE8          11:55:55.272704251 0003
                                47043000 80000000
0000 00D7 009FE828 SVC     1 00000000_30E2E7C2 00000000 00000001 009D7DD0 Wait          11:55:55.272704395 0003
                                47043000 80000000
0000 00D7 009FE828 SVCR   1 00000000_30E2E7C2 809FEB78 00000001 009D7DD0          11:55:55.272704445 0003
                                47043000 80000000
0000 0001 00000000 I/O 020C7 00000000_00000000 00C04007 30F82960 0C000000 00000080 00000000 0001 0001 11:55:55.272841768 0003
                                07060000 00000000          0269F070 00020000 00000000
0000 00D7 07C8AF00 SRB      00000000_0133D368 000000D7 07767200 0776722C      00          00D7 00D7 11:55:55.272842343 0003
                                47040000 80000000 009FE828 80
0000 00D7 07C8AF00 SSRV   2          80FEA67A 009D7DD0 7F000000 00000000 Post          11:55:55.272842816 0003
                                00000000
0000 00D7 009FE828 DSP    00000000_30E2E7C2 00000000 00000001 009D7DD0 00000000 00000000 00D7 00D7 11:55:55.272843141 0003
                                47043000 80000000

```

Before the Problem Occurs – z/OS

- **Ensure the z/OS Master Trace Table is large enough to capture events**
 - This is a table of MVS SYSLOG data
 - The default is 24K
 - The recommendation is 1000K
 - The z/OS command “MT SIZE(1000K)” can be specified in the z/OS SCHEDxx SYS1.PARMLIB member

```
DISPLAY TRACE
```

```
IEE843I 13.17.07 TRACE DISPLAY 479
```

```
SYSTEM STATUS INFORMATION
```

```
ST=(ON,3000K,12000K) AS=ON BR=OFF EX=ON MO=OFF MT=(ON,1000K)
```

Before the Problem Occurs – z/OS

▪ Use the Common Storage Tracker

- This tracks the ownership of areas of common storage (SQA/CSA)
- Support uses this for many reasons
- DIAGxx PARMLIB member
 - VSM TRACK CSA(ON) SQA(ON)
- Negligible performance impact
- Table stored in ESQA

Before the Problem Occurs – z/OS

- **Use the Common Storage Tracker**
 - Data can be displayed in IPCS in several ways

```
IPCS VERBX VSMDATA 'OWNCOMM SUMMARY'  
IPCS VERBX VSMDATA 'OWNCOMM DETAIL ALL SORTBY (ADDRESS) '  
IPCS VERBX VSMDATA 'OWNCOMM DETAIL ALL SORTBY (ASIDADDR) '  
IPCS VERBX VSMDATA 'OWNCOMM DETAIL ALL SORTBY (ASIDLEN) '  
IPCS VERBX VSMDATA 'OWNCOMM DETAIL ALL SORTBY (LENGTH) '  
IPCS VERBX VSMDATA 'OWNCOMM DETAIL ALL SORTBY (TIME) '
```

- IP VERBX VSMDATA 'OWNCOMM DETAIL **OG** SORTBY (xxxxxxx)'
 - Shows orphaned (Owner Gone) common storage
 - z/OS cleans up private virtual storage when a job ends
 - It does not clean up common storage

Before the Problem Occurs – z/OS

- Use the Common Storage Tracker

- IPCS VERBX VSMDATA 'OWNCOMM SUMMARY'

```

***** GRAND TOTALS *****

```

Description	Total Length	SQA	CSA	ESQA	ECSA	CAUB
Total SYSTEM-owned	03F75B08	054748	027250	02224778	01CD59F8	0260BA28
Total for active ASIDS	0B98C198	01EF10	0F4BD8	00C6C410	0AC0C2A0	*****
Total for "Owner Gone"	001A79E8	002BD0	074078	0001D588	00113818	*****
Total for "No Detail"	FFFFEE00	000000	000000	FFFFEE00	00000000	*****
Grand Total	0FAA8488	076228	18FEA0	02EACF10	0C9F54B0	

```

*****

```

Before the Problem Occurs – z/OS

- Use the Common Storage Tracker

- IPCS VERBX VSMDATA 'OWNCOMM SUMMARY'

Total									
ASID	Job Name	Id	St	Length	SQA	CSA	ESQA	ECSA	CAUB
----	-----	-----	--	-----	-----	-----	-----	-----	-----
0000	*SYSTEM*	Ac	03F75B08	054748	027250	02224778	01CD59F8	0260BA28
0001	*MASTER*	Ac	00A64F78	00BC90	02C650	003A2060	0068AC38	0260BB38
0002	PCAUTH	Ac	00000898	000040	000000	00000858	00000000	02ED9010
0003	RASP	Ac	000005C0	000000	000000	000005C0	00000000	02ED9078
0004	TRACE	Ac	000011E8	000000	000000	000011E8	00000000	02ED90E0
0005	DUMPSRV	Ac	0000DB70	000040	000000	0000C8D0	00001260	02ED9148
0006	XCFAS	Ac	00217F70	0000D0	000000	002179D8	000004C8	02ED91B0
0007	GRS	Ac	00005E18	000030	000000	000054E8	00000900	02ED9218
...									

Before the Problem Occurs – z/OS

■ Use the Common Storage Tracker

– IPCS VERBX VSMDATA 'OWNCOMM DETAIL ALL SORTBY(ASIDADDR)'

Date	Time	ASID	Job Name	Id	St	T	Address	Length	Ret	Addr	MM/DD/YYYY	HH:MM:SS
01E0	IMS1CTL	STC28621	Ac	C	00BD8000	00000900	380292C8	03/18/2024	20:39:51			
	Data	----->	D3C3C440	00000000	C0ACB800	00000000	*LCD{.....*				
01E0	IMS1CTL	STC28621	Ac	C	00BD9000	00001000	38029358	03/18/2024	20:39:51			
	Data	----->	00C23000	00001000	00000000	00000000	*.B.....*					
01E0	IMS1CTL	STC28621	Ac	C	00BDC000	00000740	38029358	03/18/2024	20:39:51			
	Data	----->	47F0F020	1BC4C6E2	E2C4D3C3	F04EE2D7	*.00..DFSSDLC0+SP*					
01E0	IMS1CTL	STC28621	Ac	C	00BDC740	00000108	38029358	03/18/2024	20:40:18			
	Data	----->	E2D3D6C7	C9D5E3C1	00000000	00000000	*SLOGINTA.....*					
												...

Before the Problem Occurs – z/OS

▪ **BESTFITCSA**

- Speaking of common storage – use the VSM BESTFITCSA(YES) parameter in the DIAGxx PARMLIB member
 - This avoids fragmentation of CSA/ECSA
 - Unfortunately – for historical reasons – the default is NO
- There are other parameters in the DIAGxx PARMLIB member
 - [Statements and parameters for DIAGxx](#)

Before the Problem Occurs – z/OS

- **DISPLAY DIAG** command

```
IGV007I 11.44.05 DIAG DISPLAY 713
VSM TRACK CSA(ON) SQA(ON)
VSM TRACE GET(OFF) FREE(OFF)
VSM ALLOWUSERKEYCSA(YES)
VSM BESTFITCSA(YES)
VSM USEZOSV1R9RULES(YES)
TRAPS NAME()
CBLOC
    VIRTUAL24()
    VIRTUAL31()
REUSASID(NO)
AUTOIPL SADMP(NONE) z/OS(NONE)
```

Before the Problem Occurs – z/OS

- **Allocate enough space for SVC dumps**
 - COMMNDxx member of PARMLIB
 - CD SET,SDUMP,MAXSPACE=5000M
 - Default size is 500M
 - D DUMP,OPTIONS

```
SDUMP- ADD OPTIONS
(ALLPSA , NUC , SQA , LSQA , RGN , LPA , TRT , CSA , SWA , SUMDUMP ,
Q=NO , GRSQ , XESDATA) , BUFFERS=00000000K ,
MAXSPACE=00005000M , MSGTIME=99999 MINUTES ,
MAXSNDSP=015 SECONDS , AUXMGMT=ON , DEFERTND=NO
```

Before the Problem Occurs – z/OS

▪ SVC Dumps

- Obtaining an SVC dump is very important
 - They can be manual dumps or SLIP dumps
- The commands for these dumps can be prepared ahead of time
 - IEADMCxx PARMLIB member – DUMP commands
 - [IEADMCxx PARMLIB Member](#)
 - IEASLPxx PARMLIB member – SLIP commands
 - [IEASLPxx PARMLIB Member](#)
 - Use wildcards (*) to specify job names
 - All jobs must be active or no dump is produced

Before the Problem Occurs – z/OS

▪ SVC Dumps

- DUMP command – the hard way
 - DUMP COMM=(comments)
 - That will come back with an outstand MVS reply

```
R xx,JOBNAME=(imscr,dli,dbrc,irlm),  
SDATA=(COUPLE,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT,XESDATA),END
```

Before the Problem Occurs – z/OS

▪ SVC Dumps

- DUMP command – the easy way
 - Create an IEADMCxx PARMLIB member

```
JOBNAME=(ims*,other*),  
SDATA=(ALLNUC,COUPLE,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT,XESDATA),END
```

- DUMP COMM=(comments),PARMLIB=xx

Before the Problem Occurs – z/OS

- **Prepare for SLIP**
 - Have skeleton SLIPs prepared in IEASLPxx
 - Edit the command as necessary
 - Issue SET SLIP=(xx) to enable the SLIP

Before the Problem Occurs – z/OS

▪ Prepare for SLIP

– Have skeleton SLIPs prepared in IEASLPxx

- System ABEND

```
SLIP SET ,ID=xxx ,COMP=zzz ,ACTION=SVCD ,J=jobname ,  
      JL=(ctlr gn ,dlisas ,dbrc ,irlm ,etc) ,  
SDATA=(ALLNUC ,COUPLE ,CSA ,GRSQ ,LPA ,LSQA ,PSA ,RGN ,SQA ,SUM ,SWA ,TRT ,XESDATA) ,END
```

- User ABEND

```
SLIP SET ,ID=xxx ,COMP=Uzzzz ,ACTION=SVCD ,J=jobname ,  
      JL=(ctlr gn ,dlisas ,dbrc ,irlm ,etc) ,  
SDATA=(ALLNUC ,COUPLE ,CSA ,GRSQ ,LPA ,LSQA ,PSA ,RGN ,SQA ,SUM ,SWA ,TRT ,XESDATA) ,END
```

- Message ID

```
SLIP SET ,ID=xxx ,MSGID=zzzzzzzz ,ACTION=SVCD ,J=jobname ,  
      JL=(ctlr gn ,dlisas ,dbrc ,irlm ,etc) ,  
SDATA=(ALLNUC ,COUPLE ,CSA ,GRSQ ,LPA ,LSQA ,PSA ,RGN ,SQA ,SUM ,SWA ,TRT ,XESDATA) ,END
```


Before the Problem Occurs – z/OS

- **Collect the proper SMF records**

- This is the minimum set

- SMF 30 – Common address space work
- SMF7x - RMF
- SMF 79-15 (IRLM long lock records – more later)
- SMF 98
 - High-Frequency Throughput Statistics (HFTS) and IBM® z/OS Workload Interaction Correlator
- SMF 99 at least subtypes 12 & 14
 - System Resource Manager Decisions
- SMF 113 - Hardware capacity, reporting, and statistics

Before the Problem Occurs – z/OS

▪ **Avoid ABENDS40D**

- When virtual storage becomes full z/OS will try to create a dump, e.g. ABENDS878 or ABENDS80A
- z/OS needs some storage to create the dump
 - If the dump is taken data is preserved and IMS cleanup is done
 - If there is not enough storage to create the dump you get ABEND40D
 - Terminate at end of memory
 - Data is not preserved in the dump and IMS cleanup may not occur
- You can reserve storage for z/OS dump processing using exit IEFUSI
 - Ensure your z/OS systems programmer has done this

Before the Problem Occurs – z/OS

▪ Avoid ABENDS40D

- Also have these SLIPs enabled in IEASLPxx to catch information before the dump is attempted
 - 2 SLIPs for IMS control region and 2 SLIPs for DLISAS region

```
SLIP SET, ID=xxx, COMP=80A, ACTION=SVCD, J=jobname,  
      JL=(ctrlrgn, dlisas, dbrc, irlm, etc), ENABLE,  
SDATA=(ALLNUC, COUPLE, CSA, GRSQ, LPA, LSQA, PSA, RGN, SQA, SUM, SWA, TRT, XESDATA), END
```

```
SLIP SET, ID=xxx, COMP=878, ACTION=SVCD, J=jobname,  
      JL=(ctrlrgn, dlisas, dbrc, irlm, etc), ENABLE,  
SDATA=(ALLNUC, COUPLE, CSA, GRSQ, LPA, LSQA, PSA, RGN, SQA, SUM, SWA, TRT, XESDATA), END
```

Before the Problem Occurs – z/OS

- **Avoid ABENDS322 and ABENDS522**
 - ABENDS322 – CPU time exceeded
 - ABENDS522 – Wait time exceeded
 - These abends can cause IMS not to clean up resources and should be avoided
 - JES exit IEFUTL can extend those limits dynamically
 - DFSUTL is an IMS specific IEFUTL sample exit
 - The source is in the delivered SAMPLIB

Before the Problem Occurs – z/OS

- **Avoid ABENDS722**

- Output lines exceeded
- This abend can cause IMS not to clean up resources and should be avoided
- JES exit IEFUSO can extend those limits dynamically
- DFSUSO is an IMS specific IEFUSO sample exit
- The source is in the delivered SAMPLIB

Before the Problem Occurs – IMS

▪ Use FMTO Option D

- Specify IMS Control Region EXEC PARM value FMTO=D
 - Produces an SDUMP for terminating and non-terminating errors
 - Allows the use of SYSMDUMP as a backup in case of an SDUMP failure
 - [Dump Formatting Options](#)

Before the Problem Occurs – IMS

▪ **SYSMDUMP**

- Used as backup if SDUMP files
- Have SYSMDUMP DD cards in the IMS Control Region, DLISAS and DBRC regions
- Must specify DISP=MOD
- Must be scratched and re-allocated after use
- Specify dump options in IEADMR00 PARMLIB member
 - SDATA=(CSA,RGN,SQA,SUM,TRT,ALLNUC,LPA))
- GDG example

```
//SYSMDUMP DD DISP=(MOD,CATLG),DSN=IMSAP05.SYSAIMS.SYSMDUMP(+1),  
//          DCB=(LRECL=4160,RECFM=FB,BLKSIZE=29120),UNIT=3390,  
//          SPACE=(CYL,(750,100),RLSE),DATACLASS=dtcls,VOL=SER=volser
```

Before the Problem Occurs – IMS

▪ **SYSUDUMP**

- Used by IMS dependent regions
- Specify dump options in IEADMP00 PARMLIB member
 - `SDATA=(CB,ERR,SUM) PDATA=(JPA,LPA,PSW,REGS,SA,SPLS)`

Before the Problem Occurs – IMS

▪ Install the IMS Dump Formatter

- The IMS dump formatter has EXTENSIVE options for formatting IMS and related address spaces in SVC dumps
 - [Installing the IMS Dump Formatter](#)
- Play with the dump formatter in a sandbox system to see all the wonderful features

Before the Problem Occurs – IMS

- **Activate basic IMS table traces**
 - These should be on at all times
 - They have low CPU overhead
 - DFSVSMxx PROCLIB member
 - DISP=ON, SCHD=ON, DL/I=ON, LOCK=ON
 - These are internal trace tables
 - They can be made external via command
 - /TRACE SET ON TABLE xxxx OPTION LOG
 - You may be asked to turn on other table traces for special circumstances

Before the Problem Occurs – IMS

- **Allocate External Trace Data Sets**
 - DFSTRA01 and DFSTRA02 DD cards
 - RECFM=VB
 - LRECL=4004
 - BLKSIZE=20024
 - DFSTRA01 and DFSTRA02 dynamic allocation members (DFSMDA)
 - If these data sets are not present IMS will write the external traces to the IMS log (OLDS)
 - This could affect performance

Before the Problem Occurs – IMS

- **Set LOCKTIME and DEADLOK**
 - LOCKTIME is an IMS parameter
 - Specified in the DFSVSMxx PROCLIB member for online
 - Specified in the DFSVSAMP DD for batch
 - DEADLOK is an IRLM parameter
 - LOCKTIME corresponds to IRLM parameter TIMEOUT

Before the Problem Occurs – IMS

▪ Set LOCKTIME and DEADLOK

- [Enabling the IRLM lock timeout function](#)

```
LOCKTIME= (ABEND, STATUS, ABEND, STATUS)
```

- xxxxx = LOCKTIME for Online
- yyyyy = LOCKTIME for Batch or BMP or JBP
- Value is seconds – 1-32,767
- ABEND – ABEND the thread when the value is exceeded
- STATUS – return BD status when the value is exceeded

Before the Problem Occurs – IMS

▪ Set **LOCKTIME** and **DEADLOK**

- IMS will always pass a TIMEOUT value to IRLM
 - If LOCKTIME is not coded IMS will pass 300 seconds
 - If LOCKTIME is coded IMS will pass the smallest of the Online or Batch values
- IRLM calls the IMS lock timeout exit (DFSLLKX0) if a lock waits longer than TIMEOUT
 - IMS will check Online or Batch and how long it has been waiting
 - If appropriate it will instruct IRLM to reject the lock request
 - IRLM will issue message DXR162I
 - IMS will write an SMF 79-15 record

Before the Problem Occurs – IMS

▪ Set LOCKTIME and DEADLOK

- The IRLM TIMEOUT value can be changed via command
 - F IRLM,TIMEOUT=
- The IMS LOCKTIME parameter can be changed with the UPDATE IMS command
 - [UPDATE IMS Command](#)
 - But that only changes what IMS considers a timeout
 - IMS does not notify IRLM of this change
 - You may also need to change the IRLM TIMEOUT parameter

Before the Problem Occurs – IMS

- **Set LOCKTIME and DEADLOK**
 - DEADLOK is never passed by IMS
 - It is established in the IRLM
 - TIMEOUT is checked during DEADLOK cycles
 - DEADLOK is usually much smaller than TIMEOUT or LOCKTIME
 - IRLM checks for both deadlocks and timeout regardless of any IMS parameter

Before the Problem Occurs – IMS

▪ Set **LOCKTIME** and **DEADLOK**

– Recommendations

- For many systems a DEADLOK time of 1 second is good
 - It should never be more than one second for production systems
 - IRLM checks for a deadlock only after 2 cycles
 - DEADLOK=1 can wait a long a 2 seconds
- The IRLM TIMEOUT value should be an integer factor of or equal to the lower of the Online or Batch timeout value

Before the Problem Occurs - IMS

- **There are 7 Common Service Layer (CSL) address spaces**
 - CQS – Common Queue Server
 - DBRC – Database Recovery Control (BPE is optional)
 - IMS Connect (HWS)
 - ODBM – Open Database Manager
 - OM – Operations Manager
 - RM – Resource Manager
 - SCI (Structured Call Interface)

Before the Problem Occurs - IMS

- **There are 7 Common Service Layer (CSL) address spaces**
 - These address spaces are built on a set of common services
 - BPE – Base Primitive Environment
 - BPE provides internal tracing for itself and the CSL address spaces
 - These trace tables can be very useful for analysis of complicated problems
 - The tracing is defined in control cards in the BPECFGxx PROCLIB member

Before the Problem Occurs - IMS

- **There are 7 Common Service Layer (CSL) address spaces**

- These trace control cards should be in the BPECFGxx PROCLIB member for every CSL address space
 - Control cards for the “wrong” address space will be ignored
 - They will not affect performance

```
TRCLEV=( * ,HIGH ,BPE , PAGES=300)      /* SET DEFAULT FOR BPE      */
TRCLEV=( * ,HIGH ,CQS , PAGES=300)      /* SET DEFAULT FOR CQS      */
TRCLEV=( * ,HIGH ,DBRC , PAGES=300)     /* SET DEFAULT FOR DBRC     */
TRCLEV=( * ,HIGH ,HWS , PAGES=300)     /* SET DEFAULT FOR ICON     */
TRCLEV=( * ,HIGH ,ODBM , PAGES=300)     /* SET DEFAULT FOR ODBM     */
TRCLEV=( * ,HIGH ,OM , PAGES=300)       /* SET DEFAULT FOR OM       */
TRCLEV=( * ,HIGH ,RM , PAGES=300)       /* SET DEFAULT FOR RM       */
TRCLEV=( * ,HIGH ,SCI , PAGES=300)      /* SET DEFAULT FOR SCI      */
```

While the Problem is Happening

While the Problem is Happening

- **Have a documented procedure for gathering documentation**
 - Do not wait for us to tell you what to do
 - Set up automation to execute as many steps as possible

While the Problem is Happening

▪ **IMS DC Monitor**

- The first thing you should when there is a problem is start the IMS DC Monitor
 - /TRA SET ON MON ALL
 - /TRA SET ON MON ALL INTERVAL(xx)
 - Turn monitor off after nn seconds
- 120 seconds would be ideal
- 60 seconds is ok
- I will settle for 30 seconds
- The IMS log is for integrity and recovery – not performance or detailed investigation

While the Problem is Happening

▪ Online Monitors

- Look at the online monitors
 - IMS
 - Db2
 - CICS
 - Other

- **Have your z/OS team look at RMF III (or the equivalent)**
 - **CPU delays**
 - **DASD delays**
 - **Other**

While the Problem is Happening

- **Set a SLIP**
 - Do not wait for us to tell you
 - ABEND SLIP
 - Message SLIP

While the Problem is Happening

- **Take an SVC dump**

- Do not wait for us to tell you
- DUMP COMM
- Include all IMS address spaces
 - Control, DLI, DBRC, IRLM
 - Suspect BMP's, MPR's, IFP's
- Include other address spaces as needed
 - CICS, Db2, MQ, RRS, TCP/IP ...
- They should all be in the same SVC dump

While the Problem is Happening

▪ If there is a problem with the OTMA interface from IMS

Connect or MQ

- Turn on these two traces
 - /TRA SET ON TMEMBER xxxxxxxx
 - xxxxxxxx is the IMS Connect or MQ member name
 - /TRA SET ON TABLE OTMT OPTION LOG VOLUME HIGH
- Take an SVC dump of IMS
 - Be sure to include IMS Connect or MQ in the dump

While the Problem is Happening

- **If there is problem with application interface to IMS databases**
 - Turn on the following trace
 - /TRA SET ON PSB xxxxxxxx COMP
 - (more later)

While the Problem is Happening

- **If there is a problem with application IMS DC calls**
 - GU, CHNG, ISRT, PURG to IOPCB and ALTPCB's
 - Unexpected behavior
 - Unexpected status codes
 - Turn on the following trace
 - /TRA SET ON PROGRAM psbname
 - (more later)

While the Problem is Happening

- **If there is a problem CPU or long Waits**
 - Activate APA
 - Activate STROBE

While the Problem is Happening

▪ **IMS Connect Recorder Trace**

- There are 2 IMS Connect “Recorder Traces”
 - The standard IMS Connect Recorder Trace
 - The “BPE” recorder trace
- The standard IMS Connect Recorder trace is easier to implement and easier to format
- The BPE Recorder Trace has more detail but is harder to format
- Details of these traces are in the Appendix

While the Problem is Happening

▪ **IMS Connect Extensions Traces**

- IMS Connect Extensions has Collection Level and Trace Level
 - Collection Level is 0-4
 - If collection level is not 4 you can dynamically change it using the Set System Definition panel to set it to 4 to capture more data when there is a problem
 - Trace Level is 0-2
 - You can use the TRACE command to dynamically change it to capture more data when there is a problem
 - Trace Level 2 captures application data
- Check BMC Energizer for similar function

While the Problem is Happening

▪ **TCP/IP Packet Trace**

- If you are using IMS Connect and seeing messages such as HWSP1415E and HWSP1485E (more later) you should start a TCP/IP packet trace for the Port that IMS Connect is listening on
 - Have your TCP/IP team ready to do this
 - (more later)

While the Problem is Happening

▪ **LE ABENDs**

- If you are receiving ABENDU40xx ABENDs that is LE (Language Environment) trapping ABENDs such as 0C1 or 0C4 or 0C7
 - These are usually application ABENDs
 - These are difficult to analyze as LE dumps
- You should set these LE options to get a more usable dump
 - TRAP(ON,NOSPIE)
 - TERMTHDACT(UADUMP)
- You could also set a SLIP on 0Cx and get an SVC dump
 - Read Information APAR II11016
 - [HOW TO FIND PSW AND REGS AT TIME OF ERROR WITH LE](#)

While the Problem is Happening

▪ Stopping IMS Regions

– IMS Control Region

- Always take an SVC dump BEFORE stopping/cancelling IMS
- Do not use F IMSxxx,DUMP
 - It cleans up some IMS areas before ABENDU0020

– IMS Dependent Regions

- /STOP REGION xxx
- /STOP REGION xxx ABDUMP
- **/STOP REGION XXX CANCEL**
 - **This can bring down the IMS Control Region**

While the Problem is Happening

▪ Stopping IMS Regions

- As a last resort
 - Take an SVC dump
 - Stop IMS
 - C IMSxxx,dump
 - Use Omegamon Kill or the equivalent
 - Prepare to IPL

While the Problem is Happening

- **The last things to do is ready the documentation**
 - /TRA SET OFF
 - /SWI OLDS
 - Copy monitor data sets
 - Copy standard IMS Connect Recorder Trace
 - **Save RMF III data**

After the Problem - **YOU**

After the Problem - **YOU**

- **I will be using IMS Performance Analyzer (IMSPA) and IMS Problem Investigator (IMSPI) for some examples as I am familiar with these products**
 - IMSPI can also be run in batch to produce reports of formatted log records
- **I am sure BMC AMI Log Analyzer for IMS has similar reports**

After the Problem - YOU

- **AGF – Always Google First**
 - Google knows about all IMS manuals
 - Search on an ABEND code
 - e.g. Search IMS 0402

IBM

<https://www.ibm.com/docs/en/ims/15.2.0?topic=0500-0402>

IMS 15.2 - IMS abend codes – 0402

(links to)

<https://www.ibm.com/docs/en/ims/15.2.0?topic=0500-0402>

After the Problem - YOU

- **AGF – Always Google First**
 - Google knows about all IMS manuals
 - Search on a message number
 - e.g. Search DFS554A

IBM

[https://www.ibm.com › docs › ims › topic=dfs600i-dfs5...](https://www.ibm.com/docs/en/ims/15.4.0?topic=dfs600i-dfs554a)

IMS 15.4 - IMS messages - DFS554A

(links to)

<https://www.ibm.com/docs/en/ims/15.4.0?topic=dfs600i-dfs554a>

After the Problem - YOU

- **AGF – Always Google First**
 - Google knows about all IMS closed APARs
 - Search on symptom and/or module name or APAR number
 - e.g. Search ABEND0C4 DFSASK00

IBM
[https://www.ibm.com › support › pages › apar](https://www.ibm.com/support/pages/apar)
DFSASK00 ABEND S0C4 WHEN STOPPING BMP WITH ...
When customer issuing command /STO REG XX ABDUMP to stop a JVM enabled BMP region, the BMP region abend s0c4 on DFSASK00 ...

(links to)

<https://www.ibm.com/support/pages/apar/PI40118>

PI40098: DFSASK00 ABEND S0C4 WHEN STOPPING BMP WITH /STO REG XX ABDUMP.

After the Problem - **YOU**

- **RTM – Read the Manual!!!!!**
 - IMS has a dedicated team to maintain IMS documentation
 - They are the best in the business
 - If you look up ABEND codes and error messages you can resolve many problems quickly
 - [IBM IMS Documentation](#)

After the Problem - YOU

▪ Analyzer DC Monitor

- IMS provides program DFSUTR20 to process IMS DC monitor data
 - It does not produce useful reports
- Use IMS Performance Analyzer or BMC AMI Log Analyzer for IMS

```
//IMSPA EXEC PGM=IPIMAIN,REGION=0K,PARM='V151,NOSTAE'  
//STEPLIB DD DISP=SHR,  
// DSN=highnode.IPI440.SIPILINK  
//MONITOR DD DISP=SHR,DCB=BUFNO=50,  
// DSN=highnode.data.IMS1.MONITOR  
//SYSPRINT DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),  
// DSN=highnode.date.IMS1.MON.SYSPRINT  
//SYSUDUMP DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),  
// DSN=highnode.date.IMS1.MON.SYSUDUMP  
//IPIDIAGS DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),  
// DSN=highnode.date.IMS1.MON.IPIDIAGS  
//MONREPTS DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),  
// DSN=highnode.date.IMS1.MON.REPORTS
```

After the Problem - YOU

■ Analyzer DC Monitor

- Use IMS Performance Analyzer or BMC AMI Log Analyzer for IMS

```
//IPICMD DD *
*
  IMSPAMON ESAFO (CALL) ,ALTSCHED
  IMSPAMON DDNAME (MONREPTS) ,
           STATSDDNAME (MONREPTS) ,
           DETAILDDNAME (MONREPTS)
*
  IMSPAMON SCHEDTRAN
  IMSPAMON PROGSUM
  IMSPAMON TSIWAIT (SUMMARY ,DETAIL ,DDNAME (MONREPTS) )
  IMSPAMON PSBREPORTS (BYPSTB ,FUNC-SEGNAME)
  IMSPAMON STATIS (DDNAME (MONREPTS) )
  IMSPAMON ESAF (DDNAME (MONREPTS) )
  IMSPAMON FPRSCONT (DDNAME (MONREPTS) )
  IMSPAMON SYNCCOUT (DDNAME (MONREPTS) )
  IMSPAMON FPBUFFER (DDNAME (MONREPTS) )
  IMSPAMON FPOTHRD (DDNAME (MONREPTS) )
  IMSPAMON FPVSOSUM (DDNAME (MONREPTS) )
*
  IMSPAMON EXECUTE
/*
```

After the Problem - YOU

- **Analyze the PROGRAM trace**
 - Use DFSERA10 to format the x'6701' program trace records
 - The report is not formatted well

```
//SYSIN DD *  
CONTROL CNTL STOPAFT=EOF  
OPTION PRINT O=5 ,T=X ,V=6701 ,L=2 ,COND=M ,E=DFSERA30  
OPTION PRINT O=9 ,T=C ,V=LA3A ,L=4 ,COND=E ,E=DFSERA30  
OPTION PRINT O=5 ,T=X ,V=6701 ,L=2 ,COND=M ,E=DFSERA30  
OPTION PRINT O=9 ,T=C ,V=LA3B ,L=4 ,COND=E ,E=DFSERA30  
/*
```


After the Problem - YOU

▪ Analyze the PSB Trace

- Use DFSERA10 with an exit of DFSERA50 to format the PSB trace
 - Output will be in the form of DFSDDLTO control cards
 - They can also be used to replay a scenario

```
//ERA50 EXEC PGM=DFSERA10,REGION=4000
//STEPLIB DD DISP=SHR,DSN=highnode.ims15.SDFSRESL
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DCB=BUFNO=50,
// DSN=highnode.imslog
//TRCPUNCH DD DISP=(,CATLG),UNIT=SYSDA,
// SPACE=(CYL,(500,200),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=24000),
// DSN=highnode.psbtrace.dlt0
//SYSIN DD *
CONTROL CNTL STOPAFT=EOF
OPTION PRINT O=5,T=X,V=5F,L=1,COND=E,E=DFSERA50
/*
```


After the Problem - YOU

■ Analyze the PSB Trace

- Use DFSERA10 with an exit of DFSERA50 to format the PSB trace
 - Output will be in the form of DFSDDLTO control cards

```

S 1 1 1 1 1      000051                                00400002
L      REPL                                           00400004
L      DATA D.|.....?...±ABCDEFGHIJKLMN           0  0N..... 0WXYX00400006
           1      .....*/i.m.æ.....&.....           00400008
E 01   DBD1SEG1 0009D.|.....?                       00400010
L      ISRT  DBD1SEG1                                00400012
L      DATA  ....12345.....abcdefg..±..           00400014
E 02   DBD1SEG2 00045.|.....W                       00400016
S 1 1 1 1 1      000016                                00200018
L      GU     DBD2SEG1*----- (SSAFIELD= AA.....x)  00200020
E      DATA  D.|.....?...±FGHIJKLMNOPQRETU         0  0N..... 0XYZX00200022
           2      .....abcdef....B.....           00200024
E 01   DBD2SEG1 00045.|.....V                       00200026

```

After the Problem - YOU

- **Analyze DC and DB calls and I/O's**
 - Use the IMSPA Program Trace against the IMS DC Monitor
 - Trace of every DL/I call
 - Time
 - Status
 - I/O's
 - Select by PSB and/or transaction code and/or region

```
IMSPAMON PRINTAT (EOF)
IMSPAMON TRACE (PSBNAME (PSB06500) ,
                LONG ,
                LIMIT (999) )
IMSPAMON EXECUTE
```

After the Problem - YOU

■ Analyze DC and DB calls and I/O's

– IMSPA Program Trace

Report from 18Apr2008 14.11.26.89		IMS 9.1.0		IMS Performance Analyzer 4.1		Report to 18Apr2008 14.11.26.89						
PROGRAM TRACE												
18Apr2008 14.31.46.10		PSB=PSB06500 Tran=PSB06500		Scheduled in Region 49		Page 1						
Relative Time	Pgm Time	Call	ST PCB	Feedbck	Mod IWT	DDname	IWT Elapsed	Breakdown of Call Time			Call Elap	
Secs.Mil.Mic	Sc.Mil.Mic	No.	PCBname	Func Cd	Segname Lvl	ULE (# IWAITs)	Sc.Mil.Mic	Pct CPY	Pct DLA	Pct IWT	Sc.Mil.Mic	
-	9.623					IWAIT to Load PSB06500	9.513					
-- 0 --		*** Schedule start ***										
+	226.212	0.030	1	I/O PCB	GU	QC		18.92%	81.08%	.00%	0.037	
+	528.970	302.721	2	DBD00004	GU	ROOT0004 01	1	0.951	19.84%	80.16%	64.17%	1.482
+	529.129					DBH OSAM DBD00004	0.951					
+	614.691	84.239	3	DBD00003	GU	GE 00	7	26.831	.09%	99.91%	98.73%	27.177
+	614.827					VSM VSAM DD300001	0.589					
+	615.441					VSM VSAM DD300001	0.568					
+	616.039					VSM VSAM DD300001	15.586					
+	631.685					VSM VSAM DD300002	0.555					
+	632.265					VSM VSAM DD300002	0.544					
+	632.833					VSM VSAM DD300002	0.589					
+	633.452					DBH OSAM DD300003	8.400					
+	641.897	0.029	4	DBD00003	GU	GE 00		19.44%	80.56%	.00%	0.072	
+	641.980	0.011	5	DBD00003	GU	GE 00		23.68%	76.32%	.00%	0.038	
+	642.028	0.010	6	DBD00003	GU	GE 00		20.93%	79.07%	.00%	0.043	
+	642.091	0.020	7	DBD00003	GU	GE 00		20.59%	79.41%	.00%	0.068	
+	642.174	0.015	8	DBD00003	GU	GE 00		23.08%	76.92%	.00%	0.039	
+	642.223	0.010	9	DBD00003	GU	GE 00		23.68%	76.32%	.00%	0.038	
+	642.281	0.020	10	DBD00003	GU	GE 00		21.95%	78.05%	.00%	0.041	
+	642.334	0.012	11	DBD00003	GU	GE 00		21.05%	78.95%	.00%	0.038	

After the Problem - YOU

- Analyze DC and DB calls and I/O's
 - IMSPA Program Trace - continued

PROGRAM TRACE													
18Apr2008 14.31.46.10		PSB=PSB06500 Tran=TRAN6500		Scheduled in Region 49				Page 2					
Relative Time	Pgm Time	Call	ST	PCB	Feedbk	Mod	IWT	DDname	IWT Elapsed	Breakdown of Call Time			Call Elap
Secs.Mil.Mic	Sc.Mil.Mic	No.	PCBname	Func	Cd	Segname	Lvl	ULE (# IWAITs)	Sc.Mil.Mic	Pct CPY	Pct DLA	Pct IWT	Sc.Mil.Mic
+	656.681	0.021	30	DBD00003	GU	ROOT0003	01	3	1.858	1.09%	98.91%	88.43%	2.101
+	656.840							VSM VSAM DD3IX001	0.599				
+	657.468							VSM VSAM DD3IX001	0.578				
+	658.072							VSM VSAM DD3IX001	0.681				
+	658.813	0.031	31	DBD00003	GN	GE ROOT0003	01			50.00%	50.00%	.00%	0.024
+	658.853	0.016	32	DBD00003	GN	SEG30005	03	1	0.675	1.62%	98.38%	90.97%	0.742
+	658.870							DBH OSAM DD300006	0.675				
+	659.623	0.028	33	DBD00003	GNP	SEG30006	04	1	12.702	.11%	99.89%	99.52%	12.763
+	659.635							DBH OSAM DD300007	12.702				
+	672.442	0.056	34	DBD00003	GN	SEG30005	03			26.23%	73.77%	.00%	0.061
+	672.525	0.022	35	DBD00003	GNP	SEG30006	04			21.95%	78.05%	.00%	0.041
+	672.588	0.022	36	DBD00003	GN	GE SEG30003	02			75.00%	25.00%	.00%	0.012
+	672.610	0.010	37	DBD00003	GN	GE ROOT0003	01			72.73%	27.27%	.00%	0.011
+	672.641	0.020	38	DBD00003	GN	GE ROOT0003	01			81.82%	18.18%	.00%	0.011
+	672.665	0.013	39	DBD00003	GU	SEG30003	02			39.13%	60.87%	.00%	0.023
+	672.700	0.012	40	DBD00005	GU	ROOT005	01	2	1.296	.57%	99.43%	61.68%	2.101
+	673.450							VSM VSAM DD5IX001	0.614				
+	674.091							DBH OSAM DD500001	0.682				
+	674.813	0.012	41	DBD00005	GN	SEG501	02			42.31%	57.69%	.00%	0.026
+	674.847	0.008	42	DBD00005	GN	SEG501	02			47.06%	52.94%	.00%	0.017
+	674.869	0.005	43	DBD00005	GN	SEG501	02			44.44%	55.56%	.00%	0.018
+	674.892	0.005	44	DBD00005	GN	SEG501	02			47.06%	52.94%	.00%	0.017
+	674.914	0.005	45	DBD00005	GN	SEG501	02			47.06%	52.94%	.00%	0.017
+	674.936	0.005	46	DBD00005	GN	SEG501	02			47.06%	52.94%	.00%	0.017
+	674.958	0.005	47	DBD00005	GN	SEG501	02			47.06%	52.94%	.00%	0.017

After the Problem - YOU

▪ Analyze Deadlocks

- Use DFSERA10

```
//SYSIN      DD  *  
CONTROL CNTL STOPAFT=EOF  
OPTION      PRINT  O=5 , L=2 , T=X , V=67FF , C=M , E=DFSERA30  
OPTION      PRINT  O=33 , L=8 , T=C , V=DEADLOCK , E=DFSERA30 , C=E  
/*
```

- Use IMSPA

```
//IPICMD      DD  *  
IMSPALOG     DEADLOCK (SUMMARY , LIST)  
/*
```

- [How to Read a Deadlock Report](#)
- [IMS Locking by Rich Lewis](#)

After the Problem - YOU

- **Analyze CPU and Waits**
 - Run APA reports
 - Run STROBE reports
 - Use interactive panels
 - If you see something that implicates IMS
 - Save the sample data set
 - Send APA extract
 - Send STROBE reports

After the Problem - **YOU**

- **Analyze OTMA Synchronous Callout (ICAL)**
 - IMS writes x'6701' log records for every ICAL
 - YRTR – Resume TPIPE
 - YCRT – Cancel Resume TPIPE (usually after ICON timeout)
 - YOUT – Message sent
 - YACK – ACK received from Client
 - YNAK – NAK received from Client
 - YRSP – Response received from Client
 - YAKO – ACK sent to Client
 - YNKO – NAK sent to Client

After the Problem - YOU

▪ Analyze OTMA Synchronous Callout (ICAL)

- Use IMSPI to examine the records
 - Filter on x'6701' log record type

Field Name +	Oper	Value +
CTDUPID	EQ	YRTR
CTDUPID	EQ	YCRT
CTDUPID	EQ	YOUT
CTDUPID	EQ	YACK
CTDUPID	EQ	YNAK
CTDUPID	EQ	YRSP
CTDUPID	EQ	YAKO
CTDUPID	EQ	YNKO

- MACRO DFSYMSG documents the OTMA headers
- [Format of Synchronous Callout Log Records](#)

After the Problem - YOU

- **Analyze TCP/IP Packet Trace**
 - Format the trace using IPCS
 - First – allocate an IPCS Dump Directory (IPCSDDIR)

```
//IPCSDMP EXEC PGM=IKJEFT01,REGION=8M
//SYSTSPRT DD SYSOUT=*
//INDEX DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSTSIN DD *
DELETE (IPCS.DD)
DELETE (IPCSDDIR.DD)

DEFINE CLUSTER(NAME(IPCSDDIR.DD) REUSE) -
  INDEX(NAME(IPCSDDIR.DDI) CYLINDERS(10 10) ) -
  DATA(NAME(IPCSDDIR.DDD) CYLINDERS (100 100) KEYS(128 0) -
  RECORDSIZE(384 3072) )
/*
```

After the Problem - YOU

- **Analyze TCP/IP Packet Trace**
 - Format the trace using IPCS

```
//          EXEC PGM=IKJEFT01,DYNAMNBR=100,REGION=3500K
//IPCSDDIR DD  DISP=SHR,DSN=userid.IPCSDDIR.DD
//SYSPROC  DD  DSN=SYS1.SBLSCLI0,DISP=SHR
//SYSTSPRT DD  DISP=(,CATLG),UNIT=SYSDA,
//              SPACE=(CYL,(300,300),RLSE),
//              DCB=(RECFM=VBA,LRECL=137,BLKSIZE=27998),
//              DSN=highnode.packet.print
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
PROFILE MSGID
IPCS NOPARM
SETDEF DSN('highnode.packet.trace') -
  LIST NOCONFIRM
CTRACE COMP(SYSTCPDA) -
  LOCAL FULL
END
/*
```

After the Problem - YOU

▪ Analyze TCP/IP Packet Trace

- These are the important fields

IpHeader: Version	: 4	Header Length:	20
Tos	: 00	QOS:	Routine Normal
Service			
Packet Length	: 52	ID Number:	07A1
Fragment	: DontFragment	Offset:	0
TTL	: 59	Protocol:	TCP
Ch			
Source	: 11.222.33.44		
Destination	: 22.33.444.55		
TCP			
Source Port	: 12345 ()	Destination Port:	5000
()			
Sequence Number	: 3097544804	Ack Number:	4201578462
Header Length	: 32	Flags:	Ack

After the Problem - **YOU**

▪ **Analyze TCP/IP Packet Trace**

- These are the important flags
 - ACK – this is a TCP/IP internal ACK – not an application ACK
 - SYN – start a socket connection
 - PSH – send data
 - FIN – end a socket connection
 - RST – reset a socket connection (BAD)
- Flags can be combined
 - ACK SYN
 - ACK'ing the SYN
 - ACK PSH
 - ACK a previous transmission and send data
- If using AT-TLS data will be encrypted
 - But it is the flow that is important
 - You can get the data elsewhere

After the Problem - YOU

▪ **IMS Connect HWSP1415E and HWSP1485E messages**

```
HWSP1415E TCP/IP SOCKET FUNCTION CALL FAILED; F=READ , R=-1, E=1127, M=SDRC, ID=DELDUMMY,  
IPv4=11.222.3.44
```

- IMS Connect is just the messenger
 - It did a TCP/IP Read or Write and got back a -1 return code from TCP/IP
 - This almost always means the IMS Connect Client ended the socket connection prematurely
- This should be investigated by your TCP/IP systems programmer and the application team

After the Problem - YOU

▪ IMS Connect HWSP1415E and HWSP1485E messages

```
HWSP1415E TCP/IP SOCKET FUNCTION CALL FAILED; F=READ , R=-1, E=1127, M=SDRC, ID=DELDUMMY,  
IPv4=11.222.3.44
```

- IPv4 (or IPV6) is the IP address of the Client
- E=xxxx is the TCP/IP error code
 - The most common ones are
 - 1121 (x'0461') ECONNRESET Connection reset by peer.
 - 1127 (x'0467') ETIMEDOUT Connection timed out.
 - You can Google others

After the Problem - YOU

▪ **IMS Connect HWSP1415E and HWSP1485E messages**

```
HWSP1415E TCP/IP SOCKET FUNCTION CALL FAILED; F=READ , R=-1, E=1127, M=SDRC, ID=DELDUMMY,  
IPv4=11.222.3.44
```

- If the ID is DELDUMMY this means the Client opened the socket but never sent a message
 - IMS accepted the socket and did a read for the initial message and got the -1 return code
- If the ID is HWSxxxxx it is the IMS TM Resource Adapter
- If the ID is GMPxxxxx it is the IMS TM Service Provider for z/OS Connect

After the Problem - YOU

▪ Analyze Virtual Storage

- This is important for out of storage conditions or if you need to increase pools and buffers
 - General Storage Statistics report in the IMSPA IRUR Report
 - Real Storage
 - Common Virtual Storage
 - IMS Control Region Virtual Storage
 - DLI Region Virtual Storage
 - Use the EDA option in the IMS Dump Formatter

After the Problem - YOU

▪ Analyze Virtual Storage

- Use the EDA option in the IMS Dump Formatter
 - Option 5 – SYS
 - STATS
- Processor Data

```
Processor Data:

Machine type:          3931 z16
Machine S/N:          1AA988

Total CPUs online:    5
- Normal CPs:         4
- zIIPs:              1
- zAAPs:              0

Real Storage (GB):    295.1
```

After the Problem - YOU

▪ Analyze Virtual Storage

- Use the EDA option in the IMS Dump Formatter
 - Common Storage

Common Storage Statistics:			
Storage Type	Max (Limit)	Current Alloc	% Alloc
-----	-----	-----	-----
CSA	2,772,992	1,227,208	44.3%
SQA	987,136	301,072	30.5%
ESQA	124,481,536	39,424,264	31.7%
ECSA	596,643,840	310,314,464	52.0%
Shared 64	534,773,760 M	1,135,107 M	0.2%
Common 64	67,584 M	15,693 M	23.2%

After the Problem - YOU

▪ Analyze Virtual Storage

- Use the EDA option in the IMS Dump Formatter
 - Control Region Storage

CTL Region Private Storage:

Storage Type	Max (Limit)	Current Alloc	% Alloc
User 24 Pvt	5,308,416	2,506,752	47.2%
Auth 24 Pvt	10,461,184	757,760	7.2%
Total Space	10,461,184	3,264,512	31.2%
User 31 Pvt	943,718,400	84,307,968	8.9%
Auth 31 Pvt	1,315,962,880	21,159,936	1.6%
Total Space	1,315,962,880	105,467,904	8.0%
User 24 Rgn Top	5,308,416	2,527,232	47.6%
User 24 Rgn Eff	2,527,232	2,506,752	99.2%
User 31 Rgn Top	943,718,400	84,467,712	9.0%
User 31 Rgn Eff	84,467,712	84,307,968	99.8%

CTL Region 64-bit Storage:

Storage Type	Max (Limit)	Current Alloc	% Alloc
64-bit Pvt	16,384 M	363 M	2.2%
64-bit Shr	534,773,760 M	0 M	0.0%
64-bit Com	67,584 M	2,080 M	3.1%

After the Problem - YOU

▪ Analyze Virtual Storage

- Use the EDA option in the IMS Dump Formatter
 - DLI Region Storage

DLI Region Private Storage:

Storage Type	Max (Limit)	Current Alloc	% Alloc
User 24 Pvt	9,412,608	589,824	6.3%
Auth 24 Pvt	9,412,608	405,504	4.3%
Total Space	9,412,608	995,328	10.6%
User 31 Pvt	67,108,864	18,747,392	27.9%
Auth 31 Pvt	1,414,529,024	13,606,912	1.0%
Total Space	1,414,529,024	32,354,304	2.3%
User 24 Rgn Top	9,412,608	770,048	8.2%
User 24 Rgn Eff	770,048	589,824	76.6%
User 31 Rgn Top	67,108,864	18,751,488	27.9%
User 31 Rgn Eff	18,751,488	18,747,392	100.0%

DLI Region 64-bit Storage:

Storage Type	Max (Limit)	Current Alloc	% Alloc
64-bit Pvt	16,384 M	345 M	2.1%
64-bit Shr	534,773,760 M	0 M	0.0%
64-bit Com	67,584 M	0 M	0.0%

After the Problem - YOU

▪ Analyze Virtual Storage

- Use the EDA option in the IMS Dump Formatter to see all common storage allocations
 - E(DA) – 5(SYS) – CDECOMM
 - IMS creates CDEs and SDEs for all allocated Common storage
 - SDE's track GETSTOR storage
 - SP (subpool) 227 and 228 – SQA
 - SP (subpool) 231 and 241 – CSA
 - Browse through the list
 - Interesting reading!!!

After the Problem - YOU

▪ Analyze Virtual Storage

- Use the EDA option in the IMS Dump Formatter to see all common storage allocations

Name	Address	EPA	Length	SP	Use	AS	Text
DFSLUEE0	31A12548	31A12548	88	231	1	CTL	.00..DFSLUEE012/
DFSLUDIO	2F1E7E20	2F1E7E20	8672	231	1	CTL	x4...DFSLUDIO-20
DFSLIEE0	312990F0	312990F0	1752	231	1	CTL	x4...DFSLIEE0-20
DFSATB00	2F1EBD38	2F1EBD38	4808	231	1	CTL	x4...DFSATB00-20
LWTO	31A096E0	31A096E0	800	231	1	CTL
LSCD....	31A09000	31A09000	432	231	1	CTL	LSCDLSCDE...TRIS
DFSBUCI0	00B76568	00B76568	16	231	1	CTL	..Y.....
DFSFXC00	2F264070	2F264070	3920	231	1	CTLz10
RSDSCB	00B76460	00B76460	264	231	1	CTL	DCB 1 FOR RDS:
DFSBLK0B	00B5E068	00B5E890	12184	231	1	CTL	SSCD..6.....
DFSSQHDR	00B76098	00B76098	968	231	1	CTL	.H..QHDR.....
DFSAOSFM	2F1EE000	2F1EE000	5888	231	1	CTL	x4...DFSAOSFM+15
DFSAOSF0	00B70000	00B70000	15312	241	1	CTL	...0...DFSAOSF0+
DFSVES00	319DC000	319DC000	560	231	1	CTL	.00..DFSVES00+S2
...							
Storage Summary:							
		24-Private		31-Private		24-Common	31-Common
CTL CDEs:		0		0		191,984	95,836,440
DLI CDEs:		0		0		22,184	161,063,464
SDEs:		0		0		296	4,692,144

After the Problem - **YOU**

▪ **Analyze Overall Performance**

- You can produce many reports with one execution of IMSPA
 - Output should go to data sets
 - Not SYSOUT
 - The Internal Resource Usage Report (IRUR) has the most data
 - All the reports should be browsed
 - Detailed analysis of the reports is a topic for another day

After the Problem - YOU

- **Analyze Overall Performance**
 - Start of IMSPA execution JCL

```
//IMSPA EXEC PGM=IPIMAIN,REGION=0K,PARM='V151,NOSTAE'  
//STEPLIB DD DISP=SHR,  
// DSN=highnode.IPI440.SIPILINK  
//LIMS1001 DD DISP=SHR,DCB=BUFNO=50,  
// DSN=highnode.date.IMS1.IMSLOG01  
//LIMS1002 DD DISP=SHR,DCB=BUFNO=50,  
// DSN=highnode.date.IMS1.IMSLOG02  
//IPISMQW1 DD UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE)  
//IPISMQW2 DD UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE)  
//SYSPRINT DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
// DSN=highnode.date.IMS1.LOG.SYSP  
//SYSUDUMP DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
// DSN=highnode.date.IMS1.LOG.DUMP  
//LOGINFO DD DISP=(,CATLG),UNIT=SYSDA,  
// SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
// DSN=highnode.date.IMS1.LOG.LOGINFO
```


After the Problem - YOU

■ Analyze Overall Performance

– Continuation of IMSPA Execution JCL

```
//ANALDD DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.ANAL  
//AVALDD DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.AVAL  
//CHECKPT DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.CKPT  
//CPURDD DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.CPUR  
//DBUADD DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.DBU  
//DEADLOCK DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.DEAD  
//FPDBURPT DD DISP=(,CATLG),UNIT=SYSDA,  
//      SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//      DSN=highnode.date.IMS1.LOG.URPT
```

After the Problem - YOU

▪ Analyze Overall Performance

– Continuation of IMSPA Execution JCL

```
//FPTTXRPT DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.XRPT  
//FPRUCRPT DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.CRPT  
//FPVSORPT DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.ORPT  
//GAPS      DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.GAPS  
//IRURDD    DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.IRUR  
//MGRXDD    DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.MGRX  
//MSGQDD    DD DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.MSGQ
```

After the Problem - YOU

- **Analyze Overall Performance**

- End of IMSPA Execution JCL

```
//SBDD      DD  DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.SBDD  
//SUMM0001 DD  DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.SUMM  
//TRANRESU DD  DISP=(,CATLG),UNIT=SYSDA,  
//          SPACE=(CYL,(100,100),RLSE),DCB=BLKSIZE=0,  
//          DSN=highnode.date.IMS1.LOG.RESU
```

After the Problem - YOU

■ Analyze Overall Performance

```
//IPICMD DD *
*
IMSPALOG  OUTPUTDD (ANALDD)
*
IMSPALOG  ANALYSIS (TIME , INCL (APPC , BMP , OTMA) )
IMSPALOG  ANALYSIS (TRANCODE , INCL (APPC , BMP , OTMA) )
IMSPALOG  AVAIL (PROGRAM , DATABASE)
IMSPALOG  CHECKPOINT (SUMMARY)
IMSPALOG  CPUR (TC , P)
IMSPALOG  DBUPDATE
IMSPALOG  DEADLOCK (LIST , SUMMARY)
IMSPALOG  FPDBUPD
IMSPALOG  FPIRUC (RESUSAGE , BUFFER)
IMSPALOG  FPTRNEX (CALLS , BUFFER , VSO , SYNCFAIL , OTHERFP)
IMSPALOG  FPVSO
IMSPALOG  GAP (THRESHOLD (0.100000) )
IMSPALOG  IRUR
IMSPALOG  MGREX (TCREPORT , PGMABEND , BACKOUTFAIL , SECVIO , IOERROR , SNAPTRC)
IMSPALOG  MSGQ (DDNAME (MSGQDD) )
IMSPALOG  SB (INCL (APPC , BMP , MSC , MSGSW) )
IMSPALOG  TRANRESU (SUMMARY , CALLSUM)
*
IMSPALOG  EXECUTE
/*
```

After the Problem – Helping IBM

After the Problem - IBM

- **Make the explanation as detailed as possible**
 - Especially the initial entry
 - We can start faster
 - We can route to the proper team
 - Include recent changes in the environment
 - List exact times
 - List resources
 - Transaction codes
 - PSB names
 - Show messages
 - Symptom dump if available

After the Problem - IBM

- **Send ALL documentation TERSED**
 - Including txt files (JOBLOG, SYSLOG, etc)

- **Send files without our having to ask for them**
 - Dumps
 - MVS SYSLOG
 - SYS1.LOGREC
 - IMS logs (SLDS)
 - Must contain at least 2 IMS checkpoints
 - JOBLOGs
 - Trace data sets (not reports)

After the Problem - IBM

- **Send files with descriptive file names**
 - IMSLOG1, IMSLOG2
 - IMSLOG.T0102, IMSLOG.T0105
 - We spend a lot of time looking at every log and getting them in order

End – Questions?

Appendix

Appendix

▪ Standard IMS Connect Recorder Trace

- Activated by
 - R xx,RECORDER OPEN
 - F icon, UPDATE MEMBER TYPE(IMSCON) START(TRACE)
 - UPDATE IMSCON TYPE(CONFIG) START(RECORDER)
- Events are written to a sequential data set
 - Allocate with IEFBR14
- Format and print with IDCAMS

```
//S1 EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
PRINT IDS(USER1.RECORDER.TRACE)
/*
```

- [Standard Recorder Trace Mapping](#)

Appendix

▪ Standard IMS Connect Recorder Trace

– The first 96 bytes is the prefix

```

000000 00000000 C9C3D6D5 D9C30052 00000877 15413472 0108224F 00000000 00000000 *...ICONRC.....|.....*
000020 C8E6E2E3 C5E2E3F1 C2D3F991 09C5646E C2D3F991 09E57CEE 00000000 00000000 *HWSTEST1BL9..E.>BL9..V@.....*
000040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 5CC9D7C2 *.....*IPB*

```

– The eyecatcher is ICONxx where xx=

- RC – message received from ICON Client
- SN – message sent to ICON Client
- TO – timeout sent to ICON Client
- ER – error in processing
- AR – input message from Client processed by XML adapter
- AE – input message from Client rejected by XML adapter
- AX – output message to Client processed by XML adapter

Appendix

▪ **Standard IMS Connect Recorder Trace**

–Additional trace types

- ME – MSC error in processing
- MR – MSC receive
- MS – MSC send
- OE – ODBM error in processing
- OR – ODBM receive
- OX – ODBM send
- RE – OTMA remote ALTPCB error
- RR – OTMA remote ALTPCB receive
- RS – OTMA remote ALTPCB send

Appendix

▪ Standard IMS Connect Recorder Trace

- The first 96 bytes is the prefix

```

000000 00000000 C9C3D6D5 D9C30052 00000877 15413472 0108224F 00000000 00000000 *...ICONRC.....|.....*
000020 C8E6E2E3 C5E2E3F1 C2D3F991 09C5646E C2D3F991 09E57CEE 00000000 00000000 *HWSTEST1BL9..E.>BL9..V@.....*
000040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 5CC9D7C2 *.....*IPB*

```

- The **time** of the record is at offset x'10
 - X'HHMMSSTT' local time
- The **date** of the record is at offset x'14'
 - x'CCYYDDDF'
 - CC – Century – x'01' = 21st century
 - YY – Year
 - DDD – Julian day
 - F – Packed sign
- The **Clientid** is at offset x'20'

Appendix

▪ Standard IMS Connect Recorder Trace

- The first 96 bytes is the prefix
- The ***IPB** at offset x'003C' indicates the Input Buffer follows

```

000000 00000000 C9C3D6D5 D9C30052 00000877 15413472 0108224F 00000000 00000000 *....ICONRC.....|.....*
000020 C8E6E2E3 C5E2E3F1 C2D3F991 09C5646E C2D3F991 09E57CEE 00000000 00000000 *HWSTEST1BL9..E.>BL9..V@.....*
000040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 5CC9D7C2 *.....*IPB*

```

- The ***OPB** at offset x'02FC' indicates that the Output Buffer follows

```

0002E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 5CD6D7C2 *.....*OPB*

```

Appendix

Standard IMS Connect Recorder Trace

– This is the input buffer for a message from the IMS Connect Client

```

000060  000000B2 00500000 5CE2C1D4 D7D3E55C 00000000 00001000 C8E6E2E3 C5E2E3F1 *.....&...*SAMPLE*.....HWSTEST1*
000080  00200040 E3D9C1D5 F0F0F0F1 C9C4E2C4 C5E2E3F1 40404040 40404040 E4E2C3D9 *... TRAN0001IMSDEST1      USER*
0000A0  F0F0F0F1 40404040 40404040 5C5C5C5C 5C5C5C5C 005A0000 E3D9C1D5 F0F0F0F1 *0001      *****.!..TRAN0001*
0000C0  40839389 A2A34040 40404040 40404040 40404040 40404040 40404040 40404040 *.....*
0000E0  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *.....*
000100  40404040 40404040 40404040 40400004 00000000 00000000 00000000 00000000 *.....*
000120  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
000140  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
000160  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
000180  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0001A0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0001C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0001E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
...

```

– The format is **LLLLLLZZirmLLZZtrancode** **datasuffix**

– The ‘*****’ at offset x’00AC’ is IMS Connect overlaying the input password

- This occurs even if the prefix is not an IRM

– The IRM DSECT is in MACRO HWSIMSCB

– If this was input from ITRA the OTMA prefix would have been pre-built

Appendix

Standard IMS Connect Recorder Trace

– This is the output buffer to OTMA in the ITOCRC record

```

000300 01400000 00000000 00000000 0000A0F0 00000000 00000708 00000000 00010000 * .....0.....*
000320 00480020 00400000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000340 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00004040 * .....*
000360 40404040 40400000 006AC614 0902E4E2 C5D9F0F0 F0F10903 40404040 40404040 * .....F...USER0001...*
000380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0003A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0003C0 00000000 00000000 00000000 00000000 00000100 0000C9C4 E2C4C5E2 E3F10000 * .....IMSDEST1...*
0003E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000400 00000000 00005C5C 5C5C5C5C 5C5C1000 00000000 00004040 40404040 40400000 * .....*****.....*
000420 00000200 00000000 00000000 00004040 40404040 40400000 00000000 00000000 * .....*
000440 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000460 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000480 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0004A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0004C0 00000000 00000000 00000000 00000000 0000005A 0000E3D9 C1D5F0F0 F0F140C4 * .....!...TRAN0001 D*
0004E0 C1E2C140 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *ATA*
000500 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * .....*
000520 40404040 40404040 40404040 00000000 00000000 00000000 00000000 00000000 * .....*
000540 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000560 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000580 00000000 00000000 00000000 00000000 00000000 00000000 00000000 5CC5D5C4 * .....*END*

```

– The format is **OTMALLZZ**trancode data

– The ICON **OTMA User Data** is mapped in macro HWSOMPFX

- Record is written before key fields are been valued by IMS Connect
- You will have to look in the ITOCSN record or OTMA TPIPE trace to see them

Appendix

■ Standard IMS Connect Recorder Trace

– This is the input buffer from OTMA in the ITOCSN message

```

000060 01800000 0000F9F9 F9F94040 4040A0F0 00000001 00000000 00000000 00010000 * .....9999 .0.....*
000080 00481020 00404040 40404040 40400000 00000000 00000000 00000000 00000000 * .....*
0000A0 00000000 0000C2D3 F99109E5 1DEE0000 00000000 00000000 00000000 00004040 * .....BL9..V.....*
0000C0 40404040 40400000 006AC614 0902E4E2 C5D9F0F0 F0F10903 40404040 40404040 * .....F...USER0001..*
0000E0 51005001 80555555 55555555 55555555 55555555 55555555 55555555 55555555 * ..&.....*
000100 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 * .....*
000120 555594A4 A0A2ADA1 A7155555 55555555 55550100 0000C9D4 E2C4C5E2 E3F1C8E6 * .....IMSDEST1HW*
000140 E2E3C5E2 E3F1F9F0 F0F14040 4040C2D3 F99109C1 652E0000 00000000 00000000 *STEST19001 BL9..A.....*
000160 00002754 CD480000 00000000 00001000 20000000 00004040 40404040 40400000 * .....*
000180 00000200 00000000 00000000 00004040 40404040 40400000 00000000 00000000 * .....*
0001A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0001C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0001E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000200 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
000220 00000000 00000000 00000000 00000000 00000054 0300C940 D3D6E5C5 40C9D4E2 * .....I LOVE IMS*
000240 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * .....*
000260 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * .....*
000280 40404040 40400000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0002A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*
0002C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....*

```

– The format is **OTMALLZZ**reply

– The IMS Connect **OTMA User Data** is now complete

Appendix

▪ Standard IMS Connect Recorder Trace

– This is the output buffer to the IMS Connect Client in the ITOCSN message

```

000300 00540300 C940D3D6 E5C540C9 D4E24040 40404040 40404040 40404040 40404040 *...I LOVE IMS *
000320 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * *
000340 40404040 40404040 40404040 40404040 40404040 000C0000 5CC3E2D4 D6D2E85C * *...*CSMOKY**
000360 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
000380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
...
000580 00000000 00000000 00000000 00000000 00000000 00000000 5CC5D5C4 *.....*END*

```

– The format is LLZZreplyCSMOKY

- This is what is sent from the IMS Connect sample user message exit HWSSMPL0
 - There is no LLLL
- The output could be anything the client understands
- It would be nice to pass back the input prefix for correlation
- CSMOKY is documented in MACRO HWSIMSCB

Appendix

Standard IMS Connect Recorder Trace

– This is the prefix and input buffer from the ICON Client for a Resume TPIPE

```

0000  00000000 C9E3D6C3 D9C30052 00000877  14485764 0104212F 00000000 00000000  *...ITOCRC.....*
00020 C3D3C9C5 D5E3F0F1 BB980D5C B650B241  BB980D61 7CCA4E80 00000000 00000000  *CLIENT01...*.&.../@.+.....*
00040 00000000 00000000 00000000 00000000  00000000 00000000 00000000 5CC9D7C2  *.....*IPB*
00060 0000006C 00500000 5CE2C1D4 D7D3C55C  00000000 42FF0000 C3D3C9C5 D5E3F0F1  *...%.&...*SAMPLE*.....CLIENT01*
00080 004001D9 40404040 40404040 C4E2F140  40404040 C3D3C9C5 D5E3F0F1 40404040  *. .R          DS1      CLIENT01  *
000A0 40404040 40404040 40404040 40404040  40404040 00140000 40404040 40404040  *                ....          *

```

– This is the output buffer to OTMA for the Resume TPIPE

```

002E0 00000000 00000000 00000000 00000000  00000000 00000000 00000000 5CD6D7C2  *.....*OPB*
00300 01100000 28000000 00000000 0000A0E0  00000000 00000000 00000000 00010000  *.....\.....*
00320 000C0240 C3D3C9C5 D5E3F0F1 006AC614  09024040 40404040 40400903 40404040  *... CLIENT01..F... .. *
00340 40404040 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *                .....*
00360 00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *.....*
00380 00000000 00000000 00000000 00000000  00000000 00000100 0000C4E2 F1404040  *.....DS1 *

```

- The x'01' at offset x'0301' indicates a command
- The x'28' at offset x'0304' is the Resume TPIPE on HOLDQ command
- The TPIPE name is in the State data – offset x'0324'

Appendix

▪ Standard IMS Connect Recorder Trace

– This is the prefix and start of the input buffer for a Duplicate Client

0000	00000000	C9E3D6C3	E2D50052	00000877	14485757	0104212F	00000000	00000000	* ITOCSN *
00020	C4C5D3C4	E4D4D4E8	BB980D5C	A3D50A62	00000000	00000000	BB980D5C	A3D52C82	*DELDUMMY . . . * . N *
00040	00000000	00000000	BB980D5C	A4031B02	00010000	00000000	00000000	5CC9D7C2	* * *IPB*
00060	01204000	28010000	00000000	0000A0E0	00000000	00000000	00000000	00010000	* \ *

– This is the output buffer for the Duplicate Client

- As sent by the IMS Connect sample user message exits
- The **Return Code** and **Reason Code** are at offset x'0C' in the output message
- The REQSTS is documented in MACRO HWSIMSCB

002E0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	5CD6D7C2	* *OPB*
00300	00140000	5CD9C5D8	E2E3E25C	00000008	00000038	00000000	00000000	00000000	* *REQSTS* *

Appendix

▪ BPE Recorder Trace

- Runs a BPE External Trace
- Out put goes to GDG data sets
 - Standard Record trace stopped when the data set was full
- Requires additional setup
- Uses new commands
- Uses new print facilities

Appendix

- **BPE Recorder Trace**
 - Define the GDG

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  DEFINE GENERATIONDATAGROUP -
    (NAME (IMSTESTL.RCTR.GDG01) -
     NOEMPTY -
     SCRATCH -
     LIMIT(255))
//
```

Appendix

▪ BPE Recorder Trace

- Define the External Trace using the EXTTRACE statement in the BPECFGxx member
 - [BPECFG= EXTTRACE Syntax](#)
- Examples:

```
EXTTRACE (GDGDEF ( DSN (IMSTEST .RCTR .GDG01)  
                UNIT (SYSDA) VOLSER (000000)  
                SPACE (100) SPACEUNIT (CYL)  
                BLKSIZE (32760) ) COMP (HWS) )
```

```
EXTTRACE (GDGDEF ( DSN (IMSTEST .RCTR .GDG01)  
                STORCLAS (STANDARD)  
                SPACE (100) SPACEUNIT (CYL)  
                BLKSIZE (32760) ) COMP (HWS) )
```


Appendix

▪ BPE Recorder Trace

- Define the RCTR Trace using the TRCLVL statement in the BPECFGxx member

```
TRCLEV=(RCTR, NONE, HWS, EXTERNAL=YES)
```

- If you specify NONE the trace will not be automatically started when IMS Connect starts
- If specify LOW or MEDIUM or HIGH the trace will automatically start every time IMS Connect is started
 - You probably do not want to do this
- If you do not specify EXTERNAL=YES the trace will only be written to in-core trace tables

Appendix

▪ BPE Recorder Trace

- Start the BPE Recorder Trace using the MVS modify (F) command

```
F icon,UPDATE TRACETABLE NAME(RCTR) OWNER(HWS) LEVEL(HIGH) EXTERNAL(YES)
```

- Stop the BPE Recorder Trace using the MVS modify (F) command

```
F icon,UPDATE TRACETABLE NAME(RCTR) OWNER(HWS) LEVEL(NONE) EXTERNAL(YES)
```

- Display the status of the trace using the MVS modify (F) command

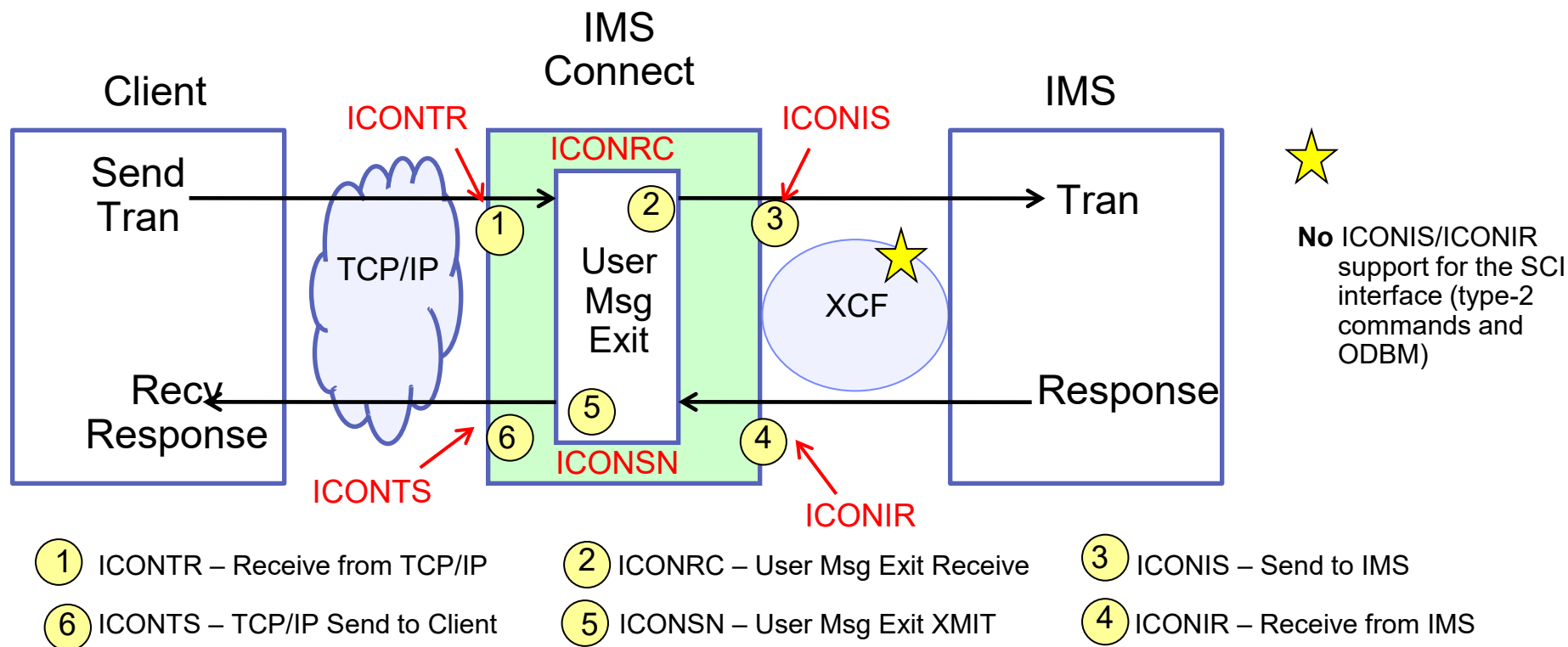
```
F HWS1,DISPLAY TRACETABLE NAME(RCTR)
```

```
BPE0030I TABLE OWNER LEVEL #PAGES EXT #ENTRIES #CYCLES
BPE0000I RCTR HWS MEDIUM 300 YES 4 0
BPE0032I DISPLAY TRACETABLE COMMAND COMPLETED
```

Appendix

▪ BPE Recorder Trace

– Trace points



Appendix

▪ BPE Recorder Trace

- Print the RCTR Recorder Trace using IPCS

```
//STEP01 EXEC PGM=IKJEFT01,REGION=0M,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=IMSVS.ADFSRES
//SYSTSPRT DD SYSOUT=*
//IPCSPRNT DD SYSOUT=*
//IPCSPARM DD DISP=SHR,DSN=SYS1.PARMLIB
//SYSTSIN DD *
  ALLOC F(IPCSDDIR) DA('IMSTESTL.DDIR') SHR REUSE
  ALLOC F(INFILE) DA('IMSTESTL.RCTR.GDG01.G0001V00') SHR REUSE
  IPCS NOPARM
  SETDEF DSN('IMSTESTL.RCTR.GDG01.G0001V00')
  SETDEF NOPROBLEM PRINT NOTERMINAL
  VERBX BPETRFM0 'TRACE(TYPE(ALL))'
  END
//
```

Appendix

▪ BPE Recorder Trace

- Print the RCTR Recorder Trace using IPCS
 - Each physical record in the RCTR Recorder data set has a header (ETHD) and multiple trace records
 - Each trace record has a header followed by the record
 - The record itself is the same 1440 byte record as for the non-BPE recorder trace
 - Except for the IR, IS, TR, and TS records which are variable length
 - The print utility prints all of the headers and the data

Appendix

▪ BPE Recorder Trace

– Print the RCTR Recorder Trace using IPCS

- Physical record header (ETHD)

```

-----
--- HWS  RCTR Trace Table ---
-----
ETHD: 00000000
+0000 LL..... 7FA0      ZZ..... 0000      TYPE.... 01      SUBTYPE.. 02      VERSION.. 0001      RESERVED. 00000000
+000C NAME..... RCTR      LENGTH... 00000080  UDATALEN. 00000000  TDATALEN. 00007F10  NUMPGS... 0014      ENTLEN... 0010
+0020 ENTSKPD.. 00000050  BVERS.... 010800  RESERVED. 00      UTYPE.... HWS      UVERS.... 0C0100  RESERVED. 00
+0030 USYSNAME.      UTRMOD... 00000000  00000000      FLG1..... 0A      FLG2..... 00      LEVEL.... 04
+0043 IDX..... 13      CYCLECT.. 00000000  OFFSET... 00000000  BYTELOST. 000113D0  LDTO..... FFFFA21F  68400000
+0058 LSO..... 00000000  00000000      STCK..... C9F3719D  2B37DFA8      RESERVED. 00000000  NEXT..... 00000000
+0070 FIRST.... 44C2A390  TOKVAL... 00000000  ID..... ETHD END
Flag analysis for ethd_flg1:
  ethd_fl_var      (08x) - Trace type creates variable length entries
  ethd_fl_det      (02x) - Trace type employs Direct Ext Trace Facility
Trace level for this table is: HIGH
ETSF: 00007F90
+0000 STCK..... C9F3719D  2B53E680      SEQNUM... 00000000  00041EB2
Variable trace table entries follow with oldest entry printed first

```

- These fields are not documented

Appendix

▪ BPE Recorder Trace

- Print the RCTR Recorder Trace using IPCS
 - Trace Record Header – one per trace record in the physical record

```
RCTR trace table entry:
      Code:      00                      Record #: 9921460
      Subcode:   8B                      Continue: 0
      TimeStamp: 213 142525.784445
TTVE: 00000000
      Variable entry prefix:
      LL..... 05C0                      REC#..... 009763B4
      ZZ..... 05C0                      CONT..... 00000000
      VLEN..... 000005A0
TTE: 00000010
      Variable entry fixed section:
      CODE..... 00
      SCDE..... 8B
      B1B2..... 0000
      WD01..... 000F0050
      STCK..... C9F3719D 2B37DFA8
      Data +00: |.....&I3.....y|
Data: 00000020 Length: 1440
```

- These fields are not documented

Appendix

▪ BPE Recorder Trace

– Print the RCTR Recorder Trace using IPCS

Variable entry variable section:									
Offset	0	4	8	C	0	4	8	C	EBCDIC Data
+000000	00000000	C9C3D6D5	E2D50052	00000877	14252578	0112213F	00000000	00000000	...ICONSN.....
+000020	D6E3D4F0	F0F0F6F6	C9F3719D	2B36F6B6	00000000	00000000	C9F3719D	2B36F6B6	OTM00066I3....6.....I3.....
+000040	00000000	00000000	00000000	00000000	00010000	00000000	00000000	5CC9D7C2*IPB
+000060	01800000	0000F9F9	F9F14040	4040A0F0	00BF5A48	00000000	00000000	000100009991 .0..!.....
+000080	00481020	00004040	40404040	40400000	00000000	00000000	00000000	00000000
+0000A0	00000000	0000C9F3	719D2A55	72B20000	00000000	00000000	00000000	00004040I3.....
+0000C0	40404040	40400000	006AC614	0902D6E3	D4F0F0F0	F6F60903	C8E6E2F1	D4C5D440F...OTM00066..HWS1MEM
+0000E0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +000100 TO +00011F SAME AS ABOVE									
+000120	00000000	00000000	00000000	00000000	00000100	0000C9D4	E9F14040	4040D6E3IMZ1 OT
+000140	D4F0F0F0	F6F6F9F9	F9F14040	4040C9F3	6DB1809D	8AB40000	00000000	00000000	M000669991 I3_.....
+000160	0000453D	A3480000	00000000	00001010	20000000	00004040	40404040	40400000	...t.....
+000180	00000203	00000000	00000000	00004040	40404040	40400000	00000000	00000000
+0001A0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0001C0 TO +00021F SAME AS ABOVE									
+000220	00000000	00000000	00000000	00000000	00000053	0000C3D9	C5C4C9E3	40D3C9D4CREDIT LIM
+000240	C9E340D6	D2404040	40404040	40404040	40404040	40404040	40404040	40404040	IT OK
+000260	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040
+000280	40404040	40000000	00000000	00000000	00000000	00000000	00000000	00000000
+0002A0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0002C0 TO +0002DF SAME AS ABOVE									
+0002E0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	5CD6D7C2*OPB
+000300	00000063	00530000	C3D9C5C4	C9E340D3	C9D4C9E3	40D6D240	40404040	40404040CREDIT LIMIT OK
+000320	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040
+000340	40404040	40404040	40404040	40404040	40404040	40404000	0C10025C	C3E2D4D6*CSMO
+000360	D2E85C00	00000000	00000000	00000000	00000000	00000000	00000000	00000000	KY*.....
+000380	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0003A0 TO +00057F SAME AS ABOVE									
+000580	00000000	00000000	00000000	00000000	00000000	00000000	00000000	5CC5D5C4*END

Appendix

■ BPE Recorder Trace

- The ICONTR (TCP/IP Receive) record is variable length and has only an input buffer
- The prefix is only 64 bytes long
- It shows the message exactly as it was received from TCP/IP

Variable entry variable section:									
Offset	0	4	8	C	0	4	8	C	EBCDIC Data
+000000	000000E8	C9C3D6D5	E3D94040	00000000	00000000	00000000	00000000	00000000	...YICONTR
+000020	00000000	00000000	C9F3719D	2B3C72B1	00000000	00000000	00000000	5CC9D7C2I3.....*IPB
+000040	000000A8	00500000	5CE2C1D4	D7D3F15C	00000000	00001000	D6E3D4F0	F0F0F6F4	...y.&..*SAMPL1*.....OTM00064
+000060	00200040	C3D3C3D2	40404040	C9D4E9F1	40404040	40404040	40404040	D6E3D4F0	... CLCK IMZ1 OTM0
+000080	F0F0F6F4	C8E6E2F1	D4C5D440	D6E3D4F0	F0F0F6F4	00500000	C3D3C3D2	40404040	0064HWS1MEM OTM00064.&..CLCK
+0000A0	40E3F0F0	F0F0F0F5	F1F0F0F0	F4F1F6F9	F2404040	40404040	40404040	F1F8F6F4	T000005100041692 1864
+0000C0	F6F5F340	40404040	40404040	40404040	40404040	40404040	40404040	40404040	653
+0000E0	40404040	00040000						

Appendix

▪ BPE Recorder Trace

- The ICONIS (IMS Send) record is variable length and has only an output buffer
- The prefix is only 64 bytes long
- It shows the message exactly as it was sent to IMS

Variable entry variable section:									
Offset	0	4	8	C	0	4	8	C	EBCDIC Data
+000000	000002DA	C9C3D6D5	C9E24040	00000000	00000000	00000000	00000000	00000000	...ICONIS
+000020	F9F9F9F1	40404040	C9F3719D	2B3D3732	00000000	00000000	00000000	5CD6D7C2	9991 I3.....*OPB
+000040	01400000	0000F9F9	F9F14040	4040A0F0	00BF5A7B	00000000	00000000	000100009991 .0..!#.....
+000060	00480020	00000000	00000000	00000000	00000000	00000000	00000000	00000000
+000080	00000000	0000C9F3	719D2B3C	ACB10000	00000000	00000000	00000000	00004040I3.....
+0000A0	40404040	40400000	006AC614	0902D6E3	D4F0F0F0	F6F40903	C8E6E2F1	D4C5D440F...OTM00064..HWS1MEM
+0000C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0000E0 TO +0000FF SAME AS ABOVE									
+000100	00000000	00000000	00000000	00000000	00000100	0000C9D4	E9F14040	4040D6E3IMZ1 OT
+000120	D4F0F0F0	F6F4F9F9	F9F14040	4040C9F3	6DFDCBF2	94B60000	00000000	00000000	M000649991 I3...2m.....
+000140	0000455F	56400000	00000000	00001010	20000000	00004040	40404040	40400000	...7.
+000160	00000203	00000000	00000000	00004040	40404040	40400000	00000000	00000000
+000180	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0001A0 TO +0001FF SAME AS ABOVE									
+000200	00000000	00000000	00000000	00000000	00000050	0000C3D3	C3D24040	404040E3&..CLCK T
+000220	F0F0F0F0	F0F5F1F0	F0F0F4F1	F6F9F240	40404040	40404040	4040F1F8	F6F4F6F5	000005100041692 186465
+000240	F3404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	3
+000260	40400000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
+000280	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0002A0 TO +0002BF SAME AS ABOVE									
+0002C0	00000000	00000000	00000000	00000000	00000000	00000000	0000	

Appendix

▪ BPE Recorder Trace

- The ICONIR (IMS Receive) record is variable length and has only an input buffer
- The prefix is only 64 bytes long
- It shows the message exactly as it was received from IMS

Variable entry variable section:									
Offset	0	4	8	C	0	4	8	C	EBCDIC Data
+000000	000002FC	C9C3D6D5	C9D94040	00000000	00000000	00000000	00000000	00000000	...ICONIR.....
+000020	00000000	00000000	C9F3719D	2B55CA31	00000000	00000000	00000000	5CC9D7C2I3.....*IPB
+000040	01800000	0000F9F9	F9F14040	4040A0F0	00BF5A79	00000000	00000000	000100009991 .0..!.....
+000060	00481020	00004040	40404040	40400000	00000000	00000000	00000000	00000000
+000080	00000000	0000C9F3	719D2B3E	5DB10000	00000000	00000000	00000000	00004040I3.....)
+0000A0	40404040	40400000	006AC614	0902D6E3	D4F0F0F0	F3F60903	C8E6E2F1	D4C5D440F...OTM00036..HWS1MEM
+0000C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0000E0 TO +0000FF SAME AS ABOVE									
+000100	00000000	00000000	00000000	00000000	00000100	0000C9D4	E9F14040	4040D6E3IMZ1 OT
+000120	D4F0F0F0	F3F6F9F9	F9F14040	4040C9F3	6DB1808C	D6340000	00000000	00000000	M000369991 I3...O.....
+000140	0000453A	26400000	00000000	00001010	20000000	00004040	40404040	40400000
+000160	00000203	00000000	00000000	00004040	40404040	40400000	00000000	00000000
+000180	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0001A0 TO +0001FF SAME AS ABOVE									
+000200	00000000	00000000	00000000	00000000	00000053	0000C3D9	C5C4C9E3	40D3C9D4CREDIT LIM
+000220	C9E340D6	D2404040	40404040	40404040	40404040	40404040	40404040	40404040	IT OK
+000240	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040
+000260	40404040	40000000	00000000	00000000	00000000	00000000	00000000	00000000
+000280	00000000	00000000	00000000	00000000	00000000	00000000	00000100	0000C9D4IM
+0002A0	E9F14040	4040D6E3	D4F0F0F0	F3F1F9F9	F9F14040	4040C9F3	6DD7A67D	B5B10000	Z1 OTM000319991 I3_Pw'....
+0002C0	00000000	00000000	00004557	E0500000	00000000	00001010	20000000	00004040&.....
+0002E0	40404040	40400000	00000203	00000000	00000000	00004040	40404040	

Appendix

■ BPE Recorder Trace

- The ICONTS (TCP/IP Send) record is variable length and has only an output buffer
- The prefix is only 64 bytes long
- It shows the message exactly as it was sent to TCP/IP

Variable entry variable section:									
Offset	0	4	8	C	0	4	8	C	EBCDIC Data
+000000	00000265	C9C3D6D5	E3E24040	00000000	00000000	00000000	00000000	00000000ICONTS
+000020	D6E3D4F0	F0F0F0F1	C9F3719D	2B56B130	00000000	00000000	00000000	5CD6D7C2	OTM00001I3.....*OPB
+000040	00000063	00530000	C3D9C5C4	C9E340D3	C9D4C9E3	40D6D240	40404040	40404040CREDIT LIMIT OK
+000060	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	
+000080	40404040	40404040	40404040	40404040	40404040	40404000	0C10025C	C3E2D4D6*CSMO
+0000A0	D2E85C00	00000000	00000000	00000000	00000000	00000000	00000000	00000000	KY*.....
+0000C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
LINES +0000E0 TO +00025F SAME AS ABOVE									
+000260	00000000	00						

Appendix End