SHARE in Orlando 2015

Handling SOS Conditions

Eugene S. Hudders
C\TREK Corporation
Montverde, FL
ehudders@ctrek.com
eshudders@aol.com
(787) 397-4150

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/

CELEBRATING 60 YEARS OF SHARE Influencing IT Since 1955

DISCLAIMERS/TRADEMARKS

- YMMV
- Remember the Political Factor
- CICS/VS, CICS/MVS, CICS/ESA, CICS TS, COBOL LE, COBOL 2, VSAM, DB2, OS/390, MVS, z/OS and z/VSE are Copyrights and/or Trademarks of the International Business Machines Armonk, NY

Agenda

- Short on Storage – SOS
- Possible Causes of SOS
- Big Storage Picture
- User 64-bit Support (GETMAIN64)
- SMX/SQE
- (E) DSA Structure
- Early Warning System (EWS)
- SOS
- Methodology Using IPCS
- Storage Fragmentation
- 64-bit Support
- Review

Short on Storage – SOS

- Simply, an SOS condition is a situation where someone has asked for virtual storage within CICS and was unable to obtain any
  - A CICS SOS condition is not an MVS GETMAIN failure that can result in a CICS cancellation
- SOS conditions are handled by the Storage Manager (SM) in CICS
  - Tasks are suspended until storage becomes available

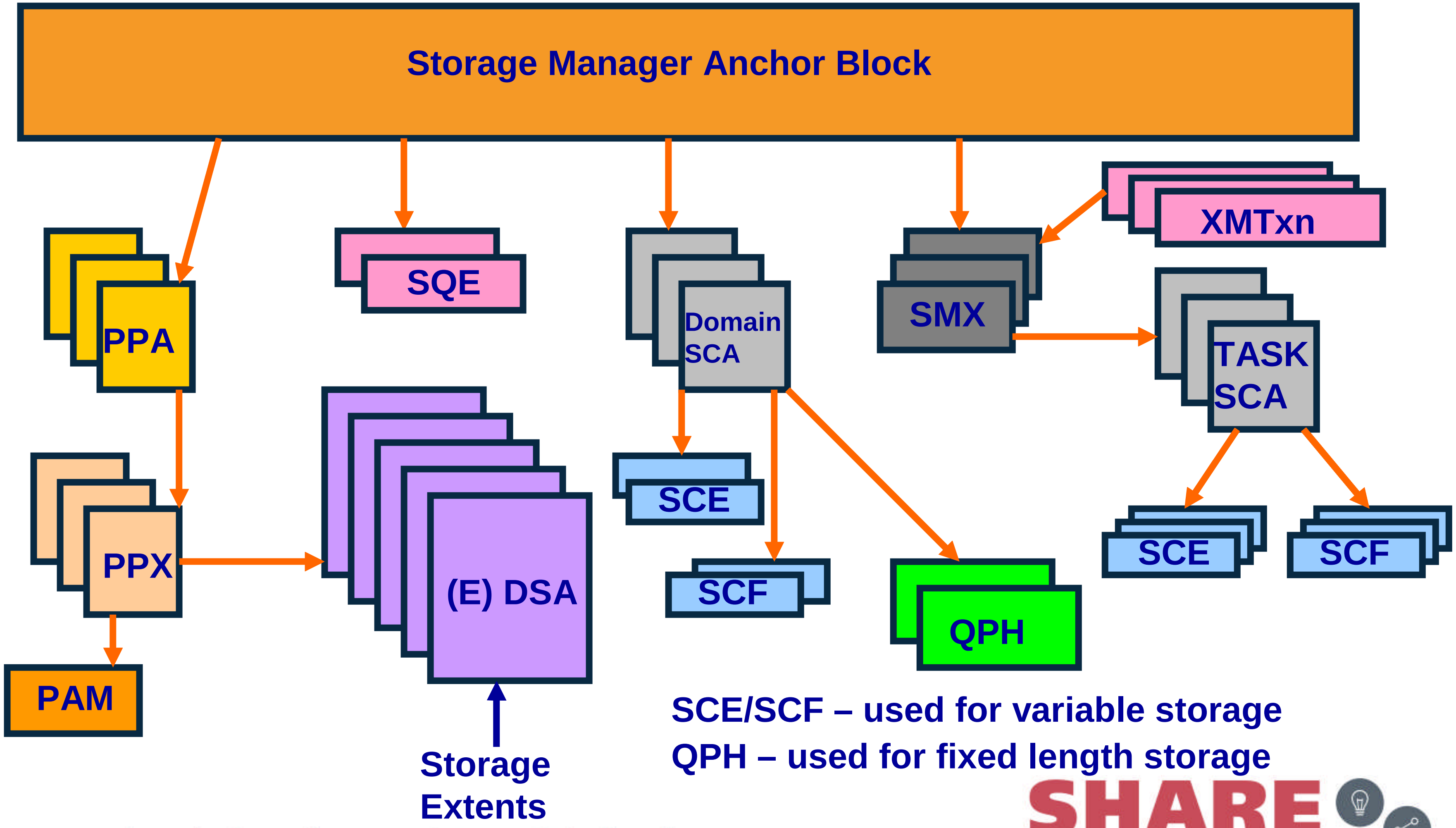
Short on Storage – SOS

- SOS conditions apply to the (E/G) DSA
- You control the amount of virtual storage available to the system via
  - SIT Parameters
    - DSALIM for storage below the line
    - EDSALIM for storage above the line
  - JCL (or other means such as SMF)
    - MEMLIMIT for storage above the Bar
    - REGION for the storage available to CICS below the Bar
- Since you control the amount of storage assigned, then these parameters are considered to be limit conditions

Possible Causes of SOS

- Available storage is too small
  - Low Region size (limits DSALIM/EDSALIM)
  - Low DSALIM/EDSALIM specification
  - Insufficient MEMLIMIT
- MXT is set to a high value
- 24-bit storage programs
- Storage Violations
- Incorrect use of Shared Storage (SDSA/ESDA/GSDSA)
  - Tasks acquiring storage and not releasing the acquired storage GETMAIN SHARED
- Transaction loops requesting storage
- Excessive storage requests
- Unused and over specified TCTUAs and/or TWAs
  - Particularly problematical below the line

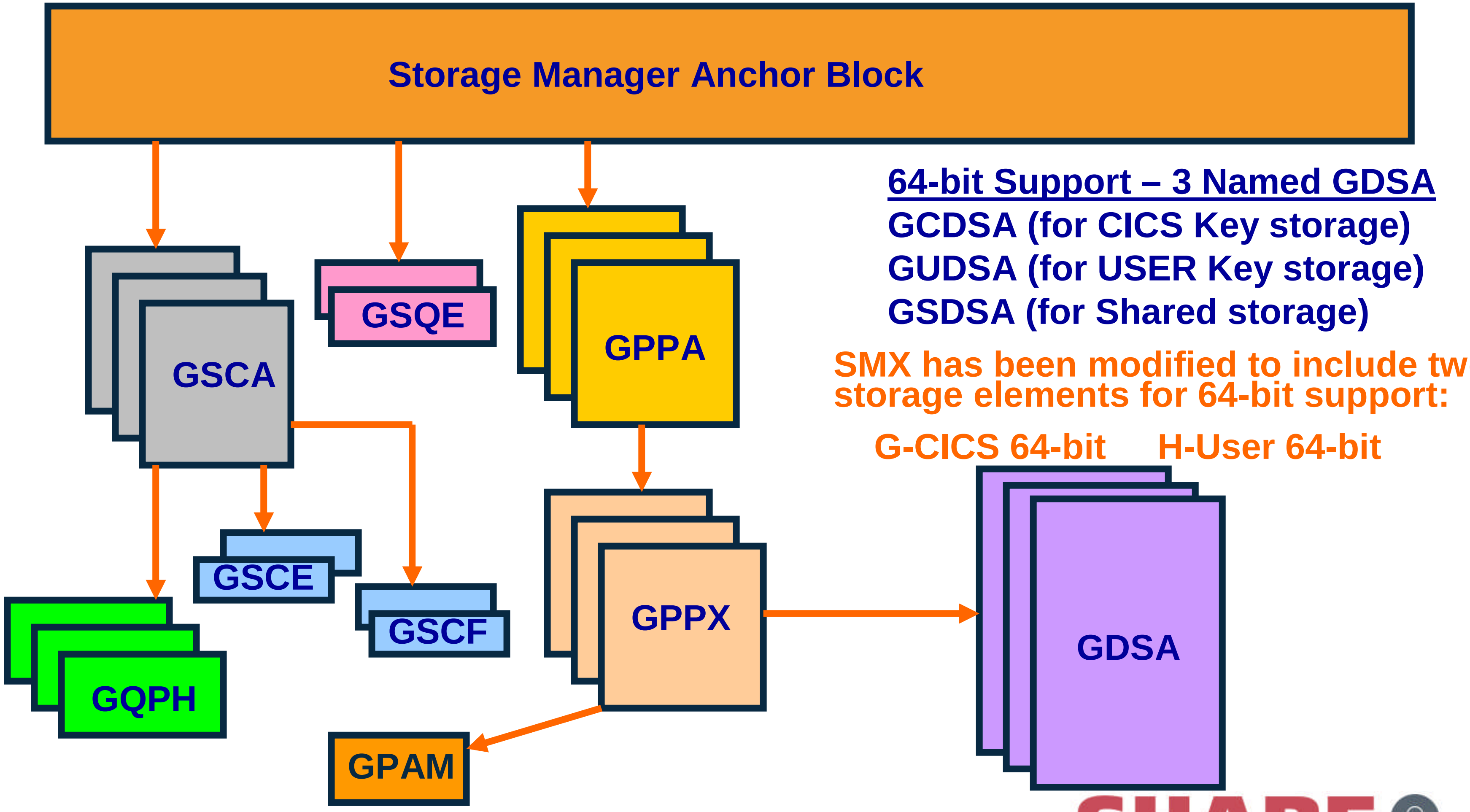
Big Storage Picture



SCE/SCF – used for variable storage

QPH – used for fixed length storage

Big Storage Picture



<u>64-bit Support – 3 Named GDSA</u>

GCDSA (for CICS Key storage)

GUDSA (for USER Key storage)

GSDSA (for Shared storage)

SMX has been modified to include two storage elements for 64-bit support:

G-CICS 64-bit H-User 64-bit

Task 64-Bit Support

- Command available to obtain 64-bit storage
  - Available only for non-LE AMODE(64) assembler application programs
- Format

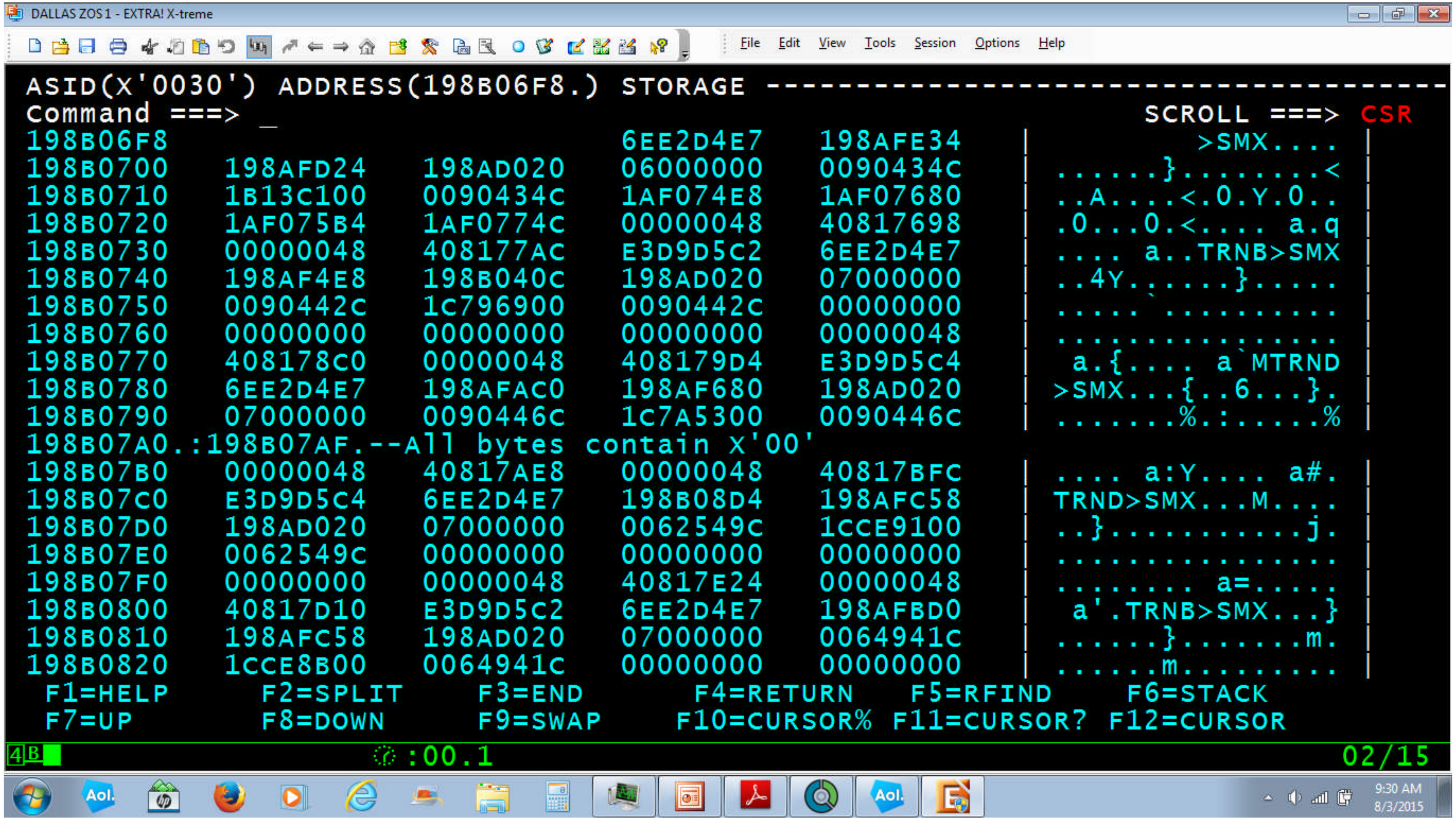
  -
    ```
    EXEC CICS GETMAIN64
              SET (set-ptr-64)
              FLENGTH (data-value)
              LOCATION (cvda)
              SHARED
              NOSUSPEND
              USERDATAKEY|CICSDATAKEY
    END-EXEC
    ```

Transaction Storage Area
SMX

SMX	
+X’004’	↑ Next SMX
+X’008’	↑ Previous SMX
+X’010’	FLAGS
+X’014’	TRAN #
+X’018’	↑ XMTxn
+X’020’	↑ CICS24 (M)
+X’024’	↑ CICS31 (C)
+X’028’	↑ USER24 (B)
+X’02C’	↑ USER31 (U)
+X’030’	↑ CICS64 (G) (8 bytes long)
+X’038’	↑ USER64 (H) (8 bytes long)
+X’040’	Transaction Id

Flags: Used to indicate things like storage freeze, clear on FREEMAIN, etc.

SMX



```
ASID(X'0030') ADDRESS(198B06F8.) STORAGE ----------------------------------
Command ===> _                                                  SCROLL ===> CSR
198B06F8                      6EE2D4E7  198AFE34  |         >SMX.... |
198B0700  198AFD24  198AD020  06000000  0090434C  | ......}........< |
198B0710  1B13C100  0090434C  1AF074E8  1AF07680  | ..A....<.0.Y.0.. |
198B0720  1AF075B4  1AF0774C  00000048  40817698  | .0...0.<.... a.q |
198B0730  00000048  408177AC  E3D9D5C2  6EE2D4E7  | .... a..TRNB>SMX |
198B0740  198AF4E8  198B040C  198AD020  07000000  | ..4Y......}..... |
198B0750  0090442C  1C796900  0090442C  00000000  | .....`.......... |
198B0760  00000000  00000000  00000000  00000048  | ................ |
198B0770  408178C0  00000048  408179D4  E3D9D5C4  |  a.{.... a`MTRND |
198B0780  6EE2D4E7  198AFAC0  198AF680  198AD020  | >SMX...{..6...}. |
198B0790  07000000  0090446C  1C7A5300  0090446C  | .......%.:.....% |
198B07A0.:198B07AF.--All bytes contain X'00'
198B07B0  00000048  40817AE8  00000048  40817BFC  | .... a:Y.... a#. |
198B07C0  E3D9D5C4  6EE2D4E7  198B08D4  198AFC58  | TRND>SMX...M.... |
198B07D0  198AD020  07000000  0062549C  1CCE9100  | ..}...........j. |
198B07E0  0062549C  00000000  00000000  00000000  | ................ |
198B07F0  00000000  00000048  40817E24  00000048  | ........ a=..... |
198B0800  40817D10  E3D9D5C2  6EE2D4E7  198AFBD0  |  a'.TRNB>SMX...} |
198B0810  198AFC58  198AD020  07000000  0064941C  | ......}.......m. |
198B0820  1CCE8B00  0064941C  00000000  00000000  | ......m......... |
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=STACK
 F7=UP        F8=DOWN      F9=SWAP     F10=CURSOR%  F11=CURSOR?  F12=CURSOR
4B                          :00.1                                        02/15
```

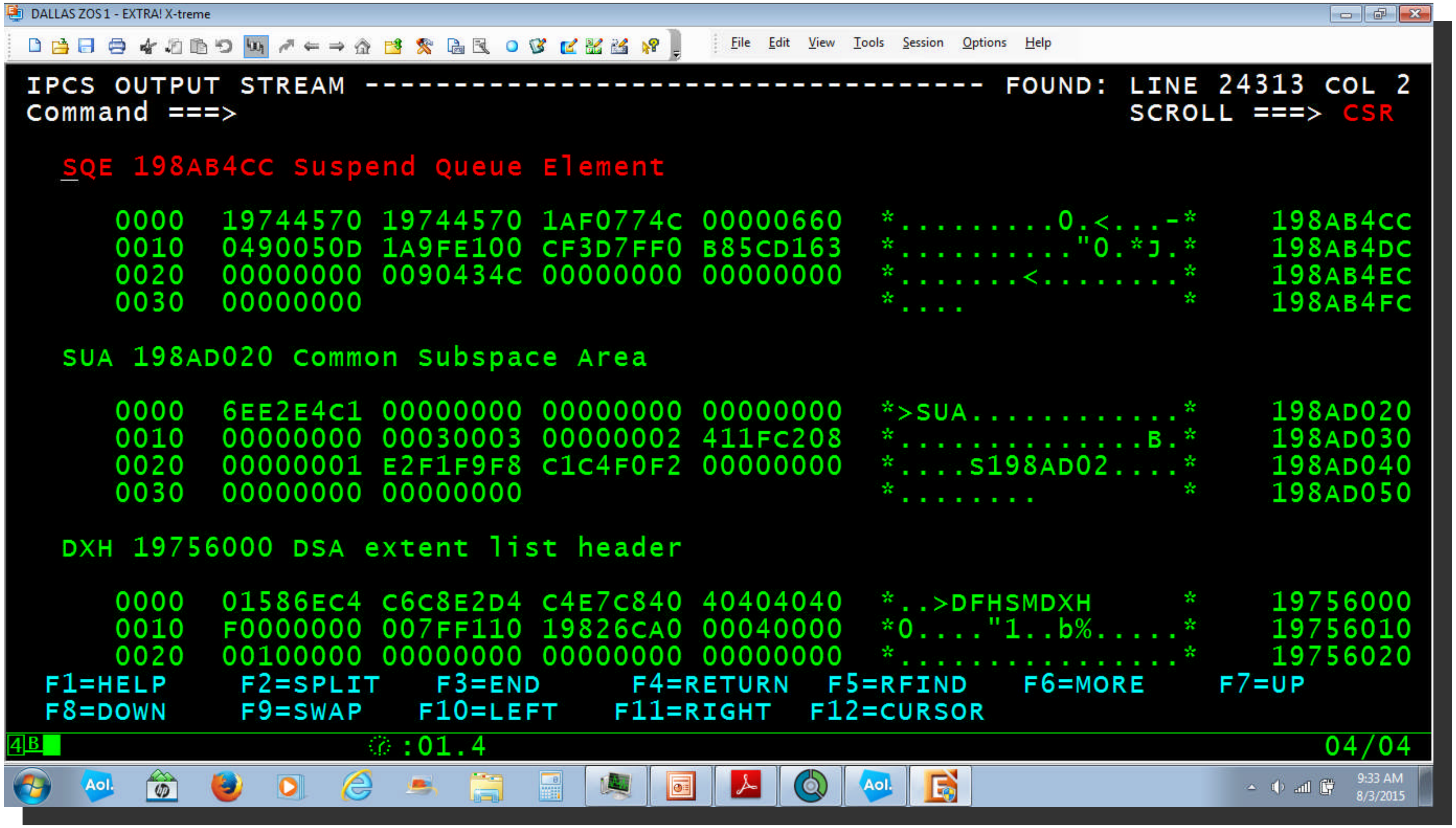
Task 64-Bit Support

- Changes in the Task storage information in the SMX control block
  - M→ CICS Key Below the Line (24-bit)
  - B → USER Key Below the Line (24-bit)
  - C → CICS Key Above the Line (31-bit)
  - U → USER Key Above the line (31-bit)
  - G → CICS Key Above the Bar (64-bit)
  - H → USER Key Above the Bar (64-bit)

Suspend Queue Element (G) SQE

SQE		GSQE
+X’000’	↑ Next SQE	+X’000’
+X’004’	↑ Previous SQE	+X’008’
+X’008’	↑ SCA	+X’010’
+X’00C’	# of Bytes Requested	+X’018’
+X’014’	↑ KE_TASK	+X’02C’
+X’018’	Time Suspended (TOD)	+X’030’
+X’024’	Transaction Number	+X’038’

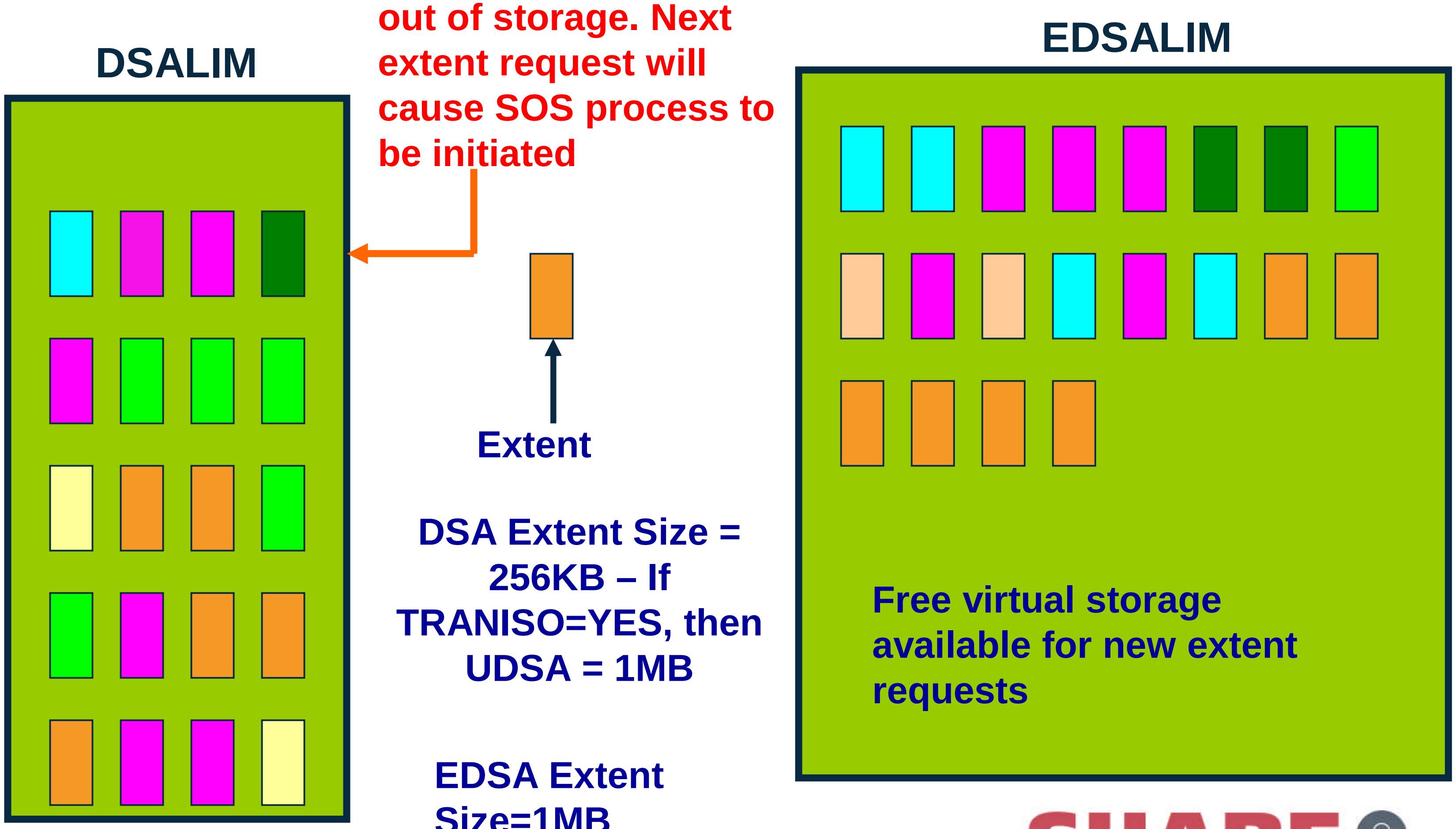
SQE



DALLAS ZOS 1 - EXTRA! X-treme
File Edit View Tools Session Options Help

```
 IPCS OUTPUT STREAM ----------------------------------- FOUND: LINE 24313 COL 2
 Command ===>                                                  SCROLL ===> CSR

   SQE 198AB4CC Suspend Queue Element

      0000  19744570 19744570 1AF0774C 00000660  *.........0.<...-*    198AB4CC
      0010  0490050D 1A9FE100 CF3D7FF0 B85CD163  *.........."0.*J.*    198AB4DC
      0020  00000000 0090434C 00000000 00000000  *.......<........*    198AB4EC
      0030  00000000                             *....            *    198AB4FC

   SUA 198AD020 Common Subspace Area

      0000  6EE2E4C1 00000000 00000000 00000000  *>SUA............*    198AD020
      0010  00000000 00030003 00000002 411FC208  *..............B.*    198AD030
      0020  00000001 E2F1F9F8 C1C4F0F2 00000000  *....S198AD02....*    198AD040
      0030  00000000 00000000                    *........        *    198AD050

   DXH 19756000 DSA extent list header

      0000  01586EC4 C6C8E2D4 C4E7C840 40404040  *..>DFHSMDXH     *    19756000
      0010  F0000000 007FF110 19826CA0 00040000  *0...."1..b%.....*    19756010
      0020  00100000 00000000 00000000 00000000  *................*    19756020
  F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=MORE     F7=UP
  F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=CURSOR
4B                              :01.4                                        04/04
```

(E) DSA Storage Structure



DSALIM

DSALIM is completely out of storage. Next extent request will cause SOS process to be initiated

Extent

DSA Extent Size = 256KB – If TRANISO=YES, then UDSA = 1MB

EDSA Extent Size=1MB

EDSALIM

Free virtual storage available for new extent requests

Early Warning System
EWS

- You should monitor the Loader/Program Domain for program fetches/loads/removed
- Three statistics fields are important to observe
  - Times program is used
    - Indicates program activity
  - Times program is fetched
    - Could be as a result of a new copy
  - Times program has been removed
    - Indication of low virtual storage availability

SOS

- SOS is usually a reflection of insufficient virtual storage in the (E) DSAs or GDSA
  - DFHSM0133 – SOS above the line
  - DFHSM0131 – SOS below the line
  - DFHSM0606 – the amount of MVS above the bar storage available to CICS is critically low
- SOS conditions can cause thrashing conditions by entering and exiting the SOS routines
  - DFHSM0134 – system is no longer under stress above
  - DFHSM0132 – system is no longer under stress below
  - DFHSM0607 – the amount of MVS above the bar storage available to CICS is no longer critically low
- Space for the (E) DSA is controlled by the SIT parameters
  - DSALIM – for virtual storage below the line
  - EDSALIM – for virtual storage above the line

SOS

- Space for the GDSA is controlled by the JCL parameter MEMLIMIT
  - There are other ways to specify MEMLIMIT
  - Fixing SOS above the Bar may require a recycling of CICS
- Many SOS conditions still occur below the line
- Unless you have a “rogue” transaction, there is no justification for SOS above the line
- Main throttles used to control SOS conditions are:
  - MXT
    - ➢ Possible solution is lowering MXT
  - TCLASS
    - ➢ Possible solution is to limit transactions that require a lot of storage

SOS

- General solutions to SOS conditions are:
  - Increase the (E) DSALIM (if possible)
    - ➢ Can be done via CEMT, if sufficient Region storage is available
      - ➢ Make sure you have a large REGION size (e.g., 0M)
    - ➢ Can probably be done above the line (EDSALIM)
    - ➢ May not be possible below the line (DSALIM)
      - ❖ “VS is free but sometimes you can’t even buy any!”
  - Increase the MEMLIMIT
    - ➢ Requires a CICS recycling
    - ➢ May not be an issue today but may be in the future as more use of above the bar storage is made available
    - ➢ Java usage
- Cancel transactions using a performance monitor or CEKL
  - May solve the problem in the short run but the SOS condition may reoccur

SOS

```
I DSA
 STATUS:  RESULTS - OVERTYPE TO MODIFY
    Sosabovebar(Notsos)          Memlimit(Nolimit)
    Sosaboveline(Notsos)         Gcdsasize(1G)
    Sosbelowline(Notsos)         Gsdsasize(0)
                                 Gudsasize(0)

   Dsalimit( 06291456 )
    Cdsasize(00524288)
    Rdsasize(00262144)
    Sdsasize(00262144)
    Udsasize(01048576)


   Edsalimit( 0629145600 )
    Ecdsasize(0152043520)
    Erdsasize(0035651584)
    Esdsasize(0001048576)
    Etdsasize(0001048576)
    Eudsasize(0024117248)

                                       SYSID=CT52 APPLID=CICSTS52
  RESPONSE: NORMAL                          TIME: 16.37.25  DATE: 07/09/15
PF 1 HELP      3 END      5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

SOS

```
•    I SYS
•     STATUS:  RESULTS - OVERTYPE TO MODIFY
•       Aging( 32768 )                  Oslevel(011300)
•       Akp( 00200 )                   Progautoctlg( Ctlgnone )
•       Cicstslevel(050200)            Progautoexit( DFHPGADX )
•       Cmdprotect(Cmdprot)             Progautoinst( Autoactive )
•       Db2conn()                      Reentprotect(Reentprot)
•       Debugtool( Nodebug )            Release(0690)
•       Dfltuser(CICSUS52)             Runaway( 0000500 )
•       Dsalimit( 06291456 )           Scandelay( 0000 )
•       Dsrtprogram( NONE     )         Sdtran(CESD)
•       Dtrprogram( DFHDYP   )          Sosabovebar(Notsos)
•       Dumping( Sysdump )              Sosaboveline(Notsos)
•       Edsalimit( 0629145600 )         Sosbelowline(Notsos)
•       Forceqr( Noforce )             Storeprotect(Active)
•       Logdefer( 00005 )              Time( 0010000 )
•       Maxtasks( 0100 )               Tranisolate(Active)
•       Memlimit(Nolimit)
•       Mqconn(MQDWCONN)
•       Mrobatch( 001 )
•
•                                            SYSID=CT52 APPLID=CICSTS52
•      RESPONSE: NORMAL                          TIME: 16.38.46  DATE: 07/09/15
•    PF 1 HELP      3 END      5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

SOS

- CICS has a built in mechanism to recover from SOS conditions
  - CICS monitors storage usage and occasionally compresses programs to recover space
  - CICS will delay new tasks when the UDSA/EUDSA is running low on virtual storage
  - CICS will perform program compression when there is no more storage available from the DSALIM/EDSALIM
    - ➢ Note: there are no programs to compress in the UDSA/EUDSA
  - CICS will steal empty extents from other (E) DSAs to satisfy storage requests from other (E) DSAs
  - When system is under stress, it will cancel tasks that have SPURGE and DTIMOUT set
  - If all else fails, set SOS and wait for storage availability
  - No new tasks are accepted

SOS

- Analyze the situation and/or produce a dump
  - You may not have time to analyze an SOS condition using a performance monitor especially if it is a production system
  - CICS does not produce a dump automatically
    - ➢ Manual dump required using CEMT
    - ➢ Set up an automatic dump by adding the SM0131, SM0133, SM0606 code using CEMT S SYD(xxxxxx)
- SOS conditions above the BAR are not easy to resolve as there are no programs, therefore, program compression is not an option
  - SOS for the storage above the bar is based on a 90% value, that is, when the total storage allocated reaches 90%, CICS considers that it is SOS above the bar. At this point, CICS will not honor new requests and will use the 10% for existing transactions
  - Fortunately, there is little above the bar usage today outside of IBM related control blocks and Channels/Containers
    - Java
  - Cancel tasks that are using GETMAIN64 storage
  - If unresolved, a recycling of CICS may be required and an increase in the MEMLIMIT may be required

Setting Up for SM Dump

```
I SYD
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Syd(EYU0XZPT) Sys Loc     Max( 005 ) Cur(0000)
  Syd(EYU0XZSD) Sys Loc     Max( 005 ) Cur(0000)
  Syd(FC0304  ) Sys Loc     Max( 001 ) Cur(0003)
  Syd(SM0131  ) Sys Loc     Max( 005 ) Cur(0000)
  Syd(SM0133  ) Sys Loc     Max( 005 ) Cur(0001)
  Syd(SM0606  ) Sys Loc     Max( 005 ) Cur(0000)













                                         SYSID=CT52 APPLID=CICSTS52
  RESPONSE: NORMAL                             TIME: 12.01.25  DATE: 07/13/15
PF 1 HELP       3 END       5 VAR        7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

SOS

- Other solutions to resolve SOS conditions
  - ➢ Lower MXT
  - ➢ Use TCLASS to control storage “hogs”
  - ➢ Split CICS (more MRO)
  - ➢ Move BMS maps above the line to relieve SOS below the line
  - ➢ Use DTIMOUT and SPURGE
  - ➢ Convert 24-bit programs to 31-bit
  - ➢ Correct storage violation situations
  - ➢ Control excessive storage requests
  - ➢ Eliminate unneeded resource definitions
  - ➢ Tune system by reducing the task residency time – Reduce Physical I/O and CPU utilization
    - ▪ LSR tuning
    - ▪ DB2 thread reuse/Threadsafe
    - ▪ NSR buffering
    - ▪ DFHTEMP buffering
  - ➢ Ensure need for TWA
  - ➢ Ensure proper sizing for TCTUA in large terminal regions
  - ➢ Properly size # of strings for NSR files
  - ➢ Increase Region size if not at maximum or IEFUSI intercepts
  - ➢ Increase MEMLIMIT

SOS Methodology

- Get a dump to analyze with IPCS
  - Use the KE, SM, DS and TR displays
- Review tasks that were in storage
  - Look at suspended tasks to see suspend reason
  - Review storage allocated by task
    - ➢ Tasks waiting for storage may not necessarily be the cause of the SOS condition
    - ➢ Review requests for large pieces of storage

SOS Methodology

- Some times the SOS condition could have been caused by a Domain storage area (SCA)
  - This type of bug is normally IBM’s and in general should be documented
  - Each (E) DSA has its own Domain control blocks
  - To determine if the SOS was caused by a Domain storage request, you will need to review the Domain SCA information
    - ➢ Look for a Domain storage that has a lot allocated (Pagestg)

Domain Storage

Look at the last column for very large numbers

```
IPCS OUTPUT STREAM ---------------------------------------- Line 625 Cols 4 81
 Command ===>                                        SCROLL ===> CSR
  M: Domain subpool summary (ECDSA)

 Name       Id Chn  Initf Bndry Fxlen Q-c   Gets  Frees  Elems  Elemstg Pagestg

 >LGJMC   0125        4K     4   124  Y       0      0      3     372      4K
 AITM_TAB 013F        4K     8   584  Y       0      0     20   11680     16K
 AP_TCA31 0098      128K   256  1792  Y      11      9     16   28672    128K
 AP_TXDEX 009A        4K     8    72  Y       0      0    204   14688     16K
 APAID31  00B9        4K     8   152  Y       4      4      0       0      4K
 APBMS    00B4  Y           16                0      0      0       0      0K
 APCOMM31 00B5              16                4      4      0       0      0K
 APDWE    00B7        4K     8    32  Y       5      5      0       0      4K
 APICE31  00B8        4K     8   216  Y     636    636      5    1080      4K
 APURD    00B6              16                0      0      0       0      0K
 ASYNCBUF 00CE               4  4096          0      0      1    4096      4K
 ASYNCBUF 00D0               4 65536          0      0      1   65536     64K
 BAGENRAL 008D              16                0      0      8    1536      4K
 BAOFBUSG 0096               8    24  Y       0      0      0       0      0K
 BAOFT_ST 008F               8   136  Y       0      0      0       0      0K
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
. . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Console Log Message

```
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG    303.102 S0W1 S0W1 01/17/2011 0W        4,177   COLUMNS 52- 131
 COMMAND INPUT ===>                                     SCROLL ===> CSR
0090  $HASP250 KVPPPAM PURGED -- (JOB KEY WAS C732435C)
0290  $CJ(423),P
0090  $HASP890 JOB(KVLINK42) 005
0090  $HASP890 JOB(KVLINK42)  STATUS=(AWAITING PURGE),CLASS=A,
0090  $HASP890               PRIORITY=1,SYSAFF=(ANY),HOLD=(NONE),
0090  $HASP890               PURGE=YES,CANCEL=YES
0090  $HASP250 KVPPPX PURGED -- (JOB KEY WAS C73243D1)
0090  $HASP250 KVLINK42 PURGED -- (JOB KEY WAS C73243FB)
0090  $HASP395 SYSVEA1  ENDED
0090  $HASP250 SYSVEA1 PURGED -- (JOB KEY WAS C7322E4B)
0290  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0043.
0290  IEA631I  OPERATOR SYSVEA0  NOW INACTIVE, SYSTEM=S0W1    , LU=ACSW0204
0090  $HASP395 SYSVEA0  ENDED
0090  $HASP250 SYSVEA0 PURGED -- (JOB KEY WAS C7322E02)
0290  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0025.
0090  +DFHSM0133  CICSTS41 CICS is under stress (short on storage above 16MB).
******************************** BOTTOM OF DATA ********************************
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

System is hung and the system under stress message is on the console log

↓

CEMT DSA Display

```
•       I DSA
•        STATUS:  RESULTS - OVERTYPE TO MODIFY
•           Sosabovebar(Notsos)
•           Sosaboveline(Sos)
•           Sosbelowline(Notsos)
•
•          Dsalimit( 06291456 )
•           Cdsasize(00786432)
•           Rdsasize(00262144)
•           Sdsasize(00262144)
•           Udsasize(00262144)
•
•          Edsalimit( 0096468992 )
•           Ecdsasize(0053477376)
•           Erdsasize(0027262976)
•           Esdsasize(0001048576)
•           Eudsasize(0001048576)
•
•          Memlimit(Nolimit)
•
•
•                                            SYSID=CT41 APPLID=CICSTS41
•         RESPONSE: NORMAL                           TIME: 18.54.10  DATE: 01/17/11
•       PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Important information is as follows:

1) **SOS above**

2) **The total EDSALIM is 96 MB**

3) **The amount of free storage available is 96 MB – 83 MB = 13 MB**

53.5 + 27.3 + 1.0 + 1.0 = 82.8 MB or 83 MB

Naturally, you had better taken this display before the SOS or have a CEMT ready. However, you can get this information from the dump

Dump Request

```
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
   Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG    303.102 S0W1 S0W1 01/17/2011 0W        4,193   COLUMNS 52- 131
 COMMAND INPUT ===>                                    SCROLL ===> CSR
0090  +DFHDU0201  CICSTS41 ABOUT TO TAKE SDUMP. DUMPCODE: MT0001  , DUMPID:
      1/0004
0090  IEA045I AN SVC DUMP HAS STARTED AT TIME=18.55.18 DATE=01/17/2011 017
0090  FOR ASID (002D)
0090  QUIESCE = YES
0090  IEA794I SVC DUMP HAS CAPTURED: 018
0090  DUMPID=006 REQUESTED BY JOB (CICSTS41)
0090  DUMP TITLE=CICS DUMP: SYSTEM=CICSTS41 CODE=MT0001   ID=1/0004
0090  +DFHDU0202  CICSTS41 SDUMPX COMPLETE. SDUMPX RETURN CODE X'00'
0290  IEF196I IGD100I 0D8A ALLOCATED TO DDNAME SYS00022 DATACLAS (        )
0290  IEF196I IEF285I   MVS1.SVCDUMP.S0W1.D110118.T005518.S00006     CATALOGED
0290  IEF196I IEF285I   VOL SER NOS= DUMP02.
0090  IEA611I COMPLETE DUMP ON MVS1.SVCDUMP.S0W1.D110118.T005518.S00006 023
0090  DUMPID=006 REQUESTED BY JOB (CICSTS41)
0090  FOR ASID (002D)
0090  INCIDENT TOKEN: SVSCPLEX S0W1     01/18/2011 00:55:18
********************************* BOTTOM OF DATA ********************************
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

CEMT P DUMP was issued

IPCS Option 0

```
•   ------------------------- IPCS Default Values ------------------- BOTH updated
•    Command ===>
•
•     You may change any of the defaults listed below.  The defaults shown before
•     any changes are LOCAL.  Change scope to GLOBAL to display global defaults.
•
•     Scope   ==> =6      (LOCAL, GLOBAL, or BOTH)
•
•     If you change the Source default, IPCS will display the current default
•     Address Space for the new source and will ignore any data entered in
•     the Address Space field.
•
•     Source  ==> DSNAME('MVS1.SVCDUMP.S0W1.D110118.T005518.S00006')
•     Address Space   ==>
•     Message Routing ==> NOPRINT TERMINAL
•     Message Control ==> CONFIRM VERIFY FLAG(WARNING)
•     Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL
•
•   Press ENTER to update defaults.
•
•   Use the END command to exit without an update.
•    F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
•    F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
•   . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Set up for dump

IPCS Option 6

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

===> verbx dfhpd660,'sm=3'


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


 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
. . . . . . . . . . . . . . . . . . . . . . . . . . .
```

SM Summary

```
IPCS OUTPUT STREAM ---------------------------------------- Line 48 Cols 1 78
 Command ===>                                          SCROLL ===> CSR
  ===SM: STORAGE MANAGER DOMAIN - SUMMARY

    SM Domain status:                INITIALISED
    Storage recovery:                YES
    Storage protection requested:      YES
    Storage protection active:         YES
    Reentrant program option:          PROTECT
    Transaction isolation requested:   NO
    Transaction isolation active:      NO

    Current DSA limit:               6144K
    Current DSA total:               1536K
    Currently SOS below 16M:                NO

    Current EDSA limit:                92M
    Current EDSA total:                79M

Currently SOS above 16M:             YES

Current GDSA limit:        17592186040320M
Current GDSA total:                  3M
Currently SOS above 2G:                  NO

MEMLIMIT:                          NOLIMIT
MEMLIMIT Source:                   REGION=0
F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
  F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
```

(E) DSA That is SOS

```
IPCS OUTPUT STREAM ---------------------------------------- Line 204 Cols 1 78
 Command ===>                                          SCROLL ===> CSR
  ==SM: ECDSA Summary

    Size:                         52224K
    Cushion size:                   128K
    Current free space:            3164K  ( 6%)
  * Lwm free space:                2476K  ( 4%)
  * Hwm free space:               18552K  (35%)
    Largest free area:             1024K
  * Times nostg returned:              0
  * Times request suspended:           1
    Current suspended:                 1
  * Hwm suspended:                     1
  * Times cushion released:            0
    Currently SOS:                   YES
  * Times went SOS:                    1
  * Time at SOS:            00:00:00.000
  * Storage violations:                0
    Access:                         CICS
  * Extents added:                     1
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
. . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Suspend Queue Element Information

```
IPCS OUTPUT STREAM --------------------------------------- Line 1060 Cols 1 78
 Command ===>                                              SCROLL ===> CSR



 ==SM: Suspend queue summary  ←

    KE Task  Tran #  Susptok  Subpool  DSA      Request

    1A7FF700 0000176 0806000D C0000176 ECDSA     16353024




 ==S2: Suspend queue summary  ←

    KE Task  Tran #  Susptok  Subpool  GDSA      Request




 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
 . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Below the BAR

Above the BAR

Important information from the SQE

1) **KE_Task Address (X'1A7FF700')**

2) **2) Amount of storage requested (16 MB)**

3) **(E) DSA affected (ECDSA)**

4) **Transaction Number (176)**

No additional tasks suspended because only one task caused the SOS

Task Summary

```
•   IPCS OUTPUT STREAM ---------------------------------------- Line 471 Cols 1 78
•    Command ===>                                        SCROLL ===> CSR
•       Current number of tasks:      20
•
•
•      SMX Addr Name        Id Loc Acc   Gets  Frees  Elems  Elemstg Pagestg Tran
•      32726064 M0000004 0001  B   C       0      0      0        0      0K CSOL
•               C0000004 0003  A   C       0      0      2     2000      4K
•               B0000004 0002  B   U       0      0      0        0      0K
•               U0000004 0004  A   U       0      0      0        0      0K
•      327260A8 M0000005 0001  B   C       0      0      1     1584      4K CSSY
•               C0000005 0003  A   C       0      0      0        0      0K
•               B0000005 0002  B   U       0      0      0        0      0K
•               U0000005 0004  A   U       0      0      0        0      0K
•      327260EC M0000006 0001  B   C       0      0      1     1584      4K CSSY
•               C0000006 0003  A   C       0      0      0        0      0K
•               B0000006 0002  B   U       0      0      0        0      0K
•               U0000006 0004  A   U       0      0      0        0      0K
•      32726174 M0000008 0001  B   C       0      0      0        0      0K CEPM
•               C0000008 0003  A   C       0      0      2     2352      4K
•               B0000008 0002  B   U       0      0      0        0      0K
•    F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
•    F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
•   . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Task Summary (Cont.)

```
---------------------------------------------------------------------------------------------------------------
                                          SNIP
---------------------------------------------------------------------------------------------------------------
IPCS OUTPUT STREAM ---------------------------------------- Line 539 Cols 1 78
 Command ===>                                          SCROLL ===> CSR
   327264E8 M0000161 0001  B   C      0     0     0       0     0K TREC
           C0000161 0003  A   C      3     0     3 16369680  15992K
           B0000161 0002  B   U      0     0     0       0     0K
           U0000161 0004  A   U      0     0     0       0     0K
   327265F8 M0000166 0001  B   C      1     0     1    1584     4K ...-
           C0000166 0003  A   C      3     0     3   23632    28K
           B0000166 0002  B   U      0     0     0       0     0K
           U0000166 0004  A   U      0     0     0       0     0K
   32726350 M0000171 0001  B   C      1     0     1    1584     4K CEMT
           C0000171 0003  A   C      3     0     3   23632    28K
           B0000171 0002  B   U      0     0     0       0     0K
           U0000171 0004  A   U      0     0     0       0     0K
   32726818 M0000176 0001  B   C      0     0     0       0     0K TREC
           C0000176 0003  A   C      2     0     2   16656    20K
           B0000176 0002  B   U      0     0     0       0     0K
           U0000176 0004  A   U      0     0     0       0     0K


  NOTE: figures for GETS and FREES reset at 00:00:00 GMT (the last statistics i
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
. . . . . . . . . . . . . . . . . . . . . . . . . . .
```

KE_TASK Information

```
IPCS OUTPUT STREAM ---------------------------------------- Line 177 Cols 1 78
 Command ===>                                          SCROLL ===> CSR
 008C   1A7FE100 Not Running  0005F700 00166  CEMT    327BDB00 192D5FF8
 008D   1A7FE700 ***Running** 0005F080 00171  CEMT    327BDC80 192D5FF8
 008E   1A7FF100 Unused
 008F   1A7FF700 Not Running  197E0800 00176  TREC    327BD500 192D5FF8
 0090   1A81C100 Not Running  197DD100 00074  TSVR    327BD380 192D5FF8
 0091   1A81C700 Unused
 0092   1A82B100 Unused
 0093   1A82B700 Not Running  197E1100 00161  TREC    327BD200 192D5FF8
 0094   1A83C100 Not Running  197E1800 00151  TREC    327BD080 192D5FF8
 0096   1ACEC700 Not Running  197DD800 00070  CSNE    327B3080 192D5FF8
 0097   1A8FA700 KTCB L8001   00000000                19703C00 327D3FF8
 009C   1A934100 Not Running  197D5100 00023  CFQS    196D0C80 192D5FF8
 00A5   1BC38100 Not Running  197E2800 00056  CISR    3277EE00 192D5FF8
 00A6   1BC38700 Not Running  197E4800 00057  CISE    32776080 192D5FF8
 00A7   1BC48100 Not Running  197E3100 00058  CISM    32776200 192D5FF8

 KE_NUM @STACK   LEN  TYPE ADDRESS  LINK REG OFFSET ERR NAME

 0001   1928F020 0670 Bot  9915E388 991602BC 001F34     DFHKETCB
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND  F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
. . . . . . . . . . . . . . . . . . . . . . . . . . .
```

CEMT running because dump requested – however, this is not the cause of the SOS

SHARE in Orlando 2015

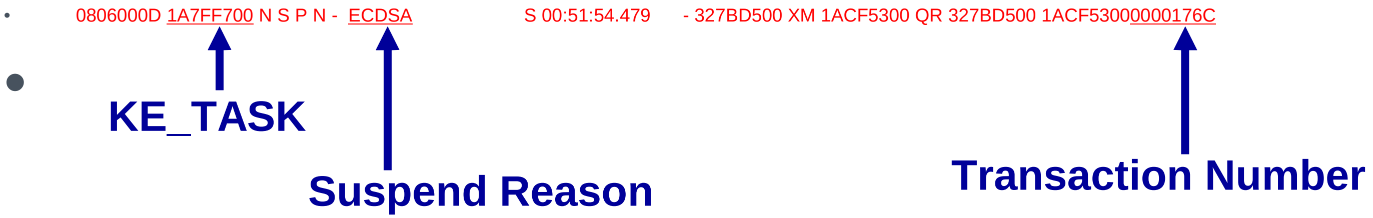
KE_TASK STACK

```
•    KE_NUM @STACK   LEN  TYPE ADDRESS  LINK REG OFFSET ERR NAME
•
•    008F   1A815020 0170 Bot  99101E00 99102216 000416     DFHKETA
•    008F   1A815190 0380 Dom  9911BD50 9911BF68 000218     DFHDSKE
•    008F   1A815510 0880 Dom  991445F8 99145938 001340     DFHXMTA
•    008F   1A815D90 06B0 Dom  99A0C6D0 99A0D80C 00113C     DFHPGPG
•                    Int   +0002DC 99A0C862 000192     INITIAL_LINK
•    008F   1A816440 0D40 Dom  99D00000 99D02E72 002E72     DFHAPLI1
•                    Int   +00330C 99D00B6C 000B6C     LE370_INTERFACE
•                    Int   +002E16 99D03E62 003E62     GETMAIN
•    008F   1A817180 02B0 Dom  9912EF40 991316B4 002774     DFHSMGF
•                    Int   +00270A 9912F1C0 000280     SUSPEND_REQUEST
•    008F   1A817430 0230 Sub  991FD710 991FE4B4 000DA4     DFHSMSQ
•                    Int   +000BAA 991FD8E6 0001D6     SUSPEND_REQUEST
•    008F   1A817660 0370 Dom  9910F428 99111932 00250A     DFHDSSR
•                    Int   +001506 9910FD96 00096E     POP_TASK
```

Note GETMAIN and the following actions taken

1) **GETMAIN issued (DFHSMGF)**
2) **Suspend requested (no storage)**
3) **SQE created (DFHSMSQ)**
4) **Dispatcher suspends (DFHDSSR)**

DS Information



```
•       0806000D 1A7FF700 N S P N -  ECDSA                S 00:51:54.479      - 327BD500 XM 1ACF5300 QR 327BD500 1ACF53000000176C
```

The Dispatcher provides a quick picture of all tasks suspended because of an SOS condition. There could be multiple (E) DSAs and/or tasks

Storage Fragmentation

- Up to now we have discussed SOS situations caused by lack of virtual storage
- There is another cause of SOS where you may have sufficient available storage but the available storage is not contiguous to accommodate the piece being requested
- In these cases you need to look at the Page Allocation Map (PAM) for each extent in the (E) DSA reflecting the SOS condition
- Need to do an SM display and then search for PAM.xxxxx where the “xxxxx” represents the (E) DSA that is SOS

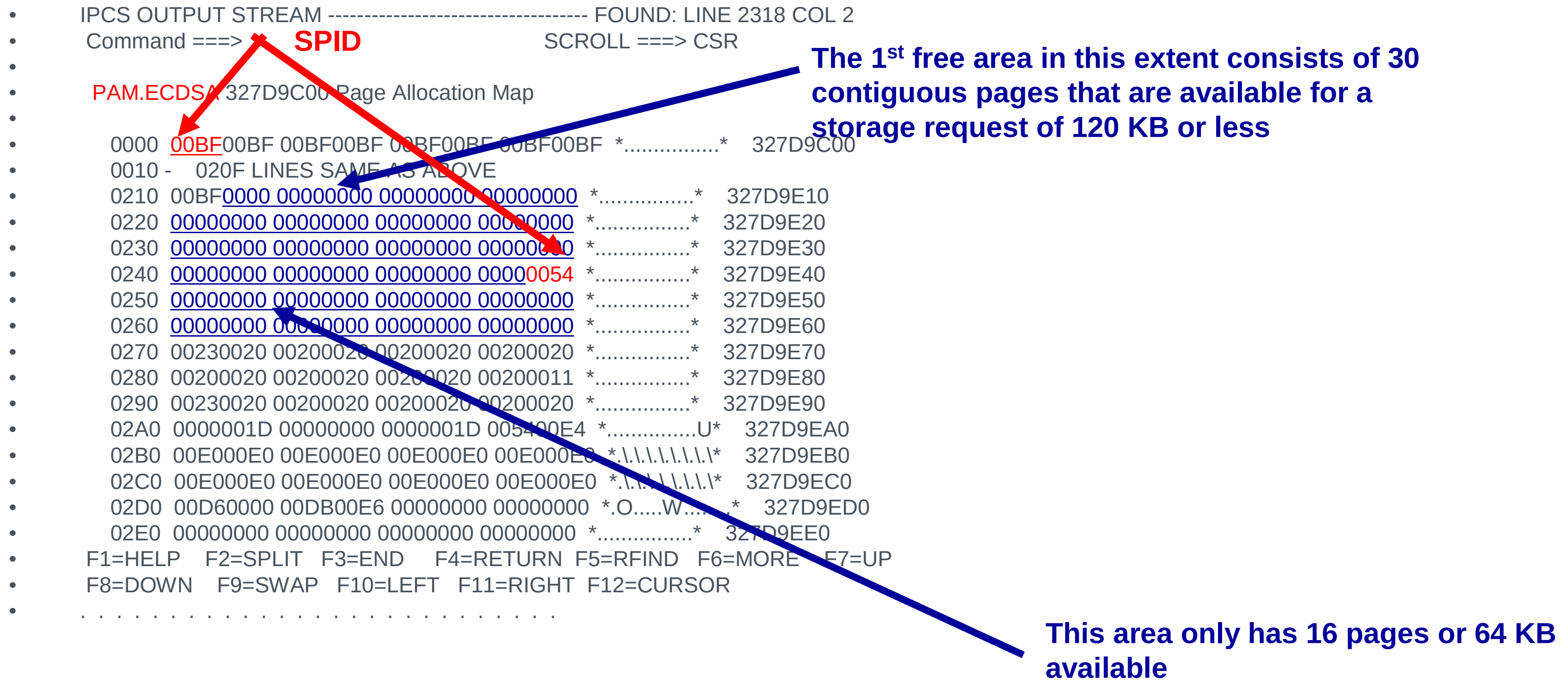
Page Allocation Map – PAM

- The PAM is a control block used to inventory the extent storage allocations
  - There is one PAM per extent
  - Each entry in the PAM is
    - 2 bytes long Below/Above the Line
    - 4 bytes long Above the Bar
  - Each PAM entry represents
    - One 4 KB page in the extent (Below/Above the Line)
    - One MB in the extent (Above the Bar)
  - (E) DSA Information
    - Storage allocations are done in multiples of 4 KB
    - So, if an extent has a size of 1 MB, then there would be 256 entries in the PAM (one per 4 KB – 1024/4) and the PAM would be 512 bytes long (256*2)
    - If a particular 4 KB entry has been assigned to a particular type of storage (SCA), then the corresponding PAM entry would contain the Subpool Identification (SPID) of the SCA
      - ➢ The SPID is a hexadecimal value assigned when the Subpool is created
      - ➢ It is a sequential counter
      - ➢ Each SCA is assigned a unique number

Page Allocation Map – PAM

  - If a particular page is unused, then the corresponding PAM entry would contain a X’0000’ indicating that this page is available
- Therefore, if a request for storage is made and there isn’t an area within the PAM that contains a contiguous number of pages to accommodate the request and there is no more (E) DSALIM available to allocate a new extent or a free extent that can be stolen, then an SOS condition is raised

Storage Fragmentation



```
IPCS OUTPUT STREAM ------------------------------------ FOUND: LINE 2318 COL 2
 Command ===>                                       SCROLL ===> CSR

  PAM.ECDSA 327D9C00 Page Allocation Map

    0000  00BF00BF 00BF00BF 00BF00BF 00BF00BF  *................*    327D9C00
    0010 -    020F LINES SAME AS ABOVE
    0210  00BF0000 00000000 00000000 00000000  *................*    327D9E10
    0220  00000000 00000000 00000000 00000000  *................*    327D9E20
    0230  00000000 00000000 00000000 00000000  *................*    327D9E30
    0240  00000000 00000000 00000000 00000054  *................*    327D9E40
    0250  00000000 00000000 00000000 00000000  *................*    327D9E50
    0260  00000000 00000000 00000000 00000000  *................*    327D9E60
    0270  00230020 00200020 00200020 00200020  *................*    327D9E70
    0280  00200020 00200020 00200020 00200011  *................*    327D9E80
    0290  00230020 00200020 00200020 00200020  *................*    327D9E90
    02A0  0000001D 00000000 0000001D 005400E4  *...............U*    327D9EA0
    02B0  00E000E0 00E000E0 00E000E0 00E000E0  *.\.\.\.\.\.\.\.\*    327D9EB0
    02C0  00E000E0 00E000E0 00E000E0 00E000E0  *.\.\.\.\.\.\.\.\*    327D9EC0
    02D0  00D60000 00DB00E6 00000000 00000000  *.O.....W........*    327D9ED0
    02E0  00000000 00000000 00000000 00000000  *................*    327D9EE0
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
. . . . . . . . . . . . . . . . . . . . . . . . . . .
```

The 1st free area in this extent consists of 30 contiguous pages that are available for a storage request of 120 KB or less

This area only has 16 pages or 64 KB available

Subpool Id

```
•    Name      Id Chn  Initf Bndry Fxlen Q-c   Gets  Frees  Elems  Elemstg Pagestg

•    PI_GENRL 0054              16              23    23    87    57744    72K
•    SMSHRC31 00BF  Y            16               0     0     5  1210464   1184K
                 ↑
                SPID
```

64-Bit Support

- CICSTS32 brought in 64-bit support for Channels and Containers
- Each new release has added additional support above the Bar, for example
  - Mark-Up Language
  - Event Processing
  - Loader
  - AP Task Storage
  - Monitor
  - TS MAIN
  - Shared Storage
  - Console Queue
  - Trace Table
  - JVM
- Size of storage allocated above the bar is controlled by MEMLIMIT parameter
  - Minimum MEMLIMIT is the size of the EDSALIM
  - Recommended MEMLIMIT size has varied by release currently 6 GB
  - Cannot be altered during CICS execution
- No program execution above the bar
- Allocations are in 1 MB increments
- Support for task storage available via GETMAIN64 Command in assembler

64-Bit Support

- Three GDSA defined but only one used consistently
  - GCDSA – Grande CICS Dynamic Storage Area
  - GUDSA – Grande User Dynamic Storage Area
  - GSDSA – Grande Shared Dynamic Storage Area
- Support above the bar is limited to CICS control blocks and areas and container data storage until users incorporate assembler code to allocate and process data Above the Bar
  - CICS handles the movement of container data to above the bar and from above the bar to below the bar for program use
  - User would have to handle the movement of data to and from Above the Bar for program use

64-Bit Support

- SOS conditions above the bar are handled completely different than the same type of conditions below the bar
  - There are no programs to compress
  - The shared areas acquired must be released explicitly by the owner
- In order to alert for possible SOS conditions, CICS uses two separate objectives
  - SOS is signaled when 90% of the MEMLIMIT has been allocated
  - Storage cushion is 5%, that is when 95% of the MEMLIMIT has been allocated
    - Used to service work already in the system
- Going SOS above the bar will probably require a re-cycling of CICS
- At this point, there is limited use of storage above the bar, so consequently the possibility of going SOS above the bar is unlikely
  - Java – should over allocate MEMLIMIT to provide an additional buffer
- However, future use of storage above the bar, especially when user access to 64-bit storage is included in programs, will require special planning

Review

- Use IPCS to determine what transactions were in storage when the SOS occurred
- Determine which transaction or group of transactions has a significant amount of storage
- Look at the SQE to see what transactions were suspended and how much storage they were requesting
  - Identify high storage requests and see if this is the cause of the SOS
- In some cases SOS can be caused by tasks being held up in storage for another reason (e.g., short on strings on a file). This condition coupled with a high MXT may lead to an SOS condition although no task may be requesting large amounts of storage
- If the (E) DSALIM is set too low, increase it substantially
  - The objective for the (E) DSALIM size should be that the peak storage used be somewhere around 60 to 80% of the assigned value
  - There is nothing wrong with over allocation of the (E) DSALIM because CICS will allocate what it needs from that value
    - Exception may be if more MVS storage is needed for buffers
    - You should always leave around 300 KB of MVS storage below the line for SDWA processing
- The peak virtual storage used within the (E) DSAs tends to creep up across time and should be reviewed regularly to avoid surprise
  - Remember your EWS

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/

Thank You!

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/