



**WAVV 2012 How to
Monitor and Optimize
CICS TS Storage**

Mike Poil

CICS Level 3 Service

IBM Hursley

poilmike@uk.ibm.com

April 2012



Important Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. There is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

When the word "CICS" is used, it refers to CICS TS for VSE/ESA 1.1.1.

Agenda

- Introduction
- z/VSE Partition structure.
- How does GETVIS/FREEVIS work?
- z/OS GETMAIN/FREEMAIN/STORAGE.
- How do I make more storage available below 16MB?
- CICS and GETVIS command output.
- How much Space Getvis do I need in a Dynamic Partition?
- The CICS DSAs.
- How does CICS use DSALIM and EDSALIM storage?
- Who is using the GETVIS storage?
- The main CICS IMVS subpools.
- I am running out of GETVIS, can I make it bigger?
- I may have a GETVIS storage leak, what do I do?
- I am running out of DSALIM or EDSALIM, can I make it bigger?
- I am getting CICS SOS, what documentation do I need?
- What can I do to get better use of DSA and EDSA storage?
- How much storage are my transactions are using?
- Q & A.

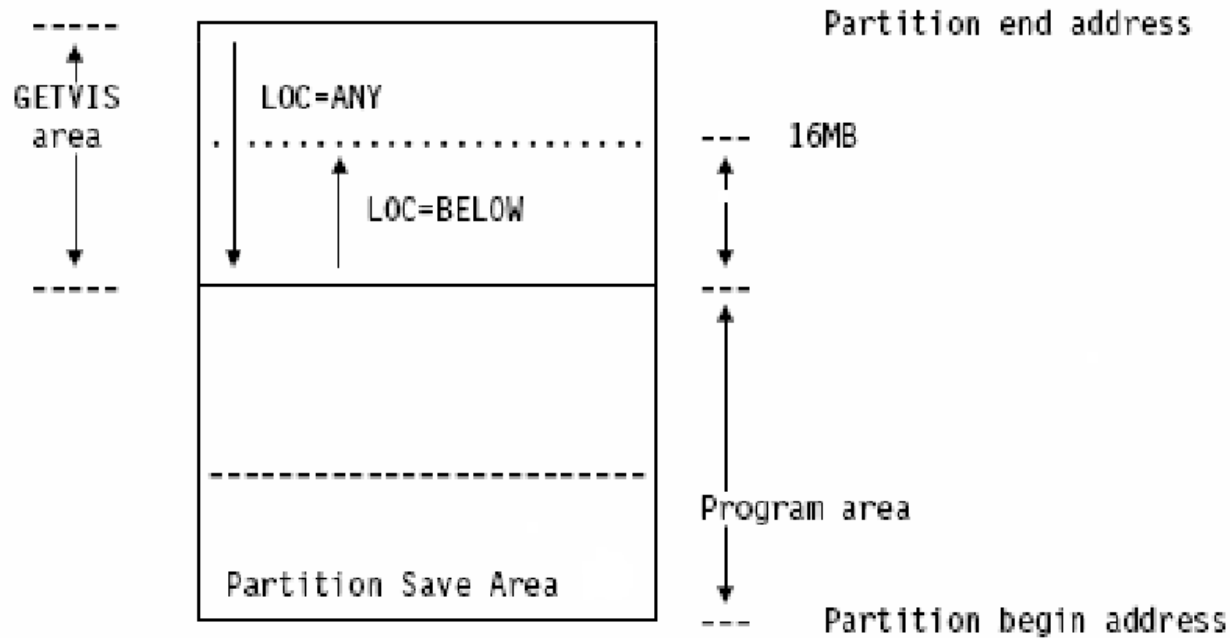
Introduction

- § This is based on experience of looking at CICS storage issues raised in PMRs.
- § If you have questions, feel free to email me.

z/VSE Partition Structure

- § The partition starts with a X'78'-byte main task save area.
- § This is followed by the "Program Area", whose size is determined by the specified or default // EXEC SIZE= value.
- § The remainder is the Partition Getvis Area and its control information.
- § CICS TS uses // EXEC DFHSIP,SIZE=DFHSIP (SIZE=4K).
- § The next slide has a simple picture.

z/VSE Partition Structure



How does GETVIS/FREEVIS work?

§ The simplified GETVIS macro looks like this:

GETVIS ADDRESS= LENGTH= <LOC= > <SPID= > <SVA=YES>

§ ADDRESS is where you want the GETVIS storage start address returned.

§ LENGTH is multiples of 128 bytes for Partition Getvis, and 16 byte multiples for System (and Space) Getvis.

§ LOC is either BELOW the 16MB line or ANY, which is above the 16MB if possible, but below it if not.

§ SPID is an 8-byte area, and allows you to group storage by usage using a 6-byte Subpool ID.

§ Without SPID you use the "default" subpool.

§ SVA=YES says use the System Getvis Area.

How does GETVIS/FREEVIS work?

- § GETVIS manages the storage as a series of named subpools.
- § A subpool owns a series of 4K pages, which may not be contiguous.
- § Each 4K page is viewed as a series of 128 or 16-byte allocation units, and a bitmap manages the individual allocations.
- § LOC=BELOW allocates from low address to high address.
- § LOC=ANY allocates from high address to low address so that it can span the 16MB line if required.
- § The simplified FREEVIS macro looks like this:
FREEVIS ADDRESS= LENGTH= <SVA=YES>
FREEVIS SPID= <SVA=YES> (free the whole subpool)

z/OS GETMAIN/FREEMAIN/STORAGE

- § CICS TS for VSE/ESA was converted from CICS/ESA 4.1, which ran on the OS/390 operating system, now called z/OS.
- § Many z/OS services are emulated, primarily by using z/VSE SVC X'84'/132, although some are invoked by SVC X'83'/131 or by a PC instruction.
- § z/OS GETMAIN/FREEMAIN/STORAGE emulation is performed by using GETVIS/FREEVIS with the **7-byte** subpool IDs "IMVSnnn", where "nnn" is the z/OS subpool number in the range 000 to 255.
- § Some of the subpool numbers used by CICS are documented later in this presentation.
- § MVS Diagnosis: Reference GA22-7588 documents z/OS SVC and subpool numbers.

How do I make more storage available below 16MB?

§ The simple answer may be that you can't, or need z/VSE 4.3/5.1.

§ The 24-bit Shared Area must be reduced in 1MB multiples, reclaim:

1. VTAM SGA24 by using IOBUF31 - do this first.
2. "UNUSED" SVA-24 storage (as shown in MAP).
3. Free SVA-24 Virtual Library.
4. Free System Getvis-24 storage, but leave a buffer.
5. Supervisor control block space, e.g. SYS SDSIZE, SVA SDL - I don't describe this.

§ **Changing only one or two IPL parameters will reclaim the storage.**

§ **Changing the base partition start address will reduce GETVIS-31, so you may need to increase some partition sizes to compensate.**

How do I make more storage available below 16MB?

- § There may be considerations that I do not know about that are related to OEM software products, discuss this with your Vendors.

How do I make more storage available below 16MB?

§ Data required:

1. Output from D NET,VTAMOPTS,OPT=IOBUF31 and D NET,BFRUSE,BUFFER=SHORT
2. SVA statement from the z/VSE IPL procedure
3. MAP output
4. LIBR LD SDL output
5. GETVIS SVA output *after z/VSE has been running for a long time.*

§ The data that follows this slide is from a system where IOBUF31 was already active, this is sample output for VTAMOPTS and BFRUSE:

```
IST1189I IOBUF31 = YES ☞ already active  
IST449I SGALIMIT = NO LIMIT, CURRENT = 988K, MAXIMUM = 5732K  
IST790I MAXIMUM SGA USED = 5732K  
IST449I SGA24 LIMIT = NO LIMIT, CURRENT = 76K, MAXIMUM = 396K  
IST790I MAXIMUM SGA24 USED = 396K ☞ some 24-bit is still used
```

How do I make more storage available below 16MB?

§ Before:

SVA GETVIS=(2M,6M),PSIZE=(652K,7M)

MAP

| SPACE AREA | V-SIZE | GETVIS | V-ADDR | UNUSED | NAME |
|------------|--------|--------|---------------|-------------|---------------------------|
| S SUP | 760K | | 0 | | \$\$A\$SUPX |
| S SVA-24 | 1916K | 2944K | BE000 | <u>384K</u> | Ç Reclaim 384K-64K = 256K |
| 0 BG V | 1536K | 10752K | <u>600000</u> | 249856K | Ç shared area 6MB |

```

-----
SDL      TOTAL ENTRIES :    908   (100%)
. . .
SVA(24)  TOTAL SPACE   :   1852K  (100%)
         USED SPACE    :   1603K  ( 87%)
         - PFIXED AREA:    166K  (  9%)  START AT: 00273540
         FREE SPACE    :    249K  ( 13%)           Ç Reclaim 192K (64K multiples)
-----

```

```

-----
GETVIS USAGE  SVA-24  SVA-ANY  SVA-24  SVA-ANY
AREA SIZE:    2,912K  9,704K
USED AREA:    1,516K  5,544K MAX. EVER USED:  1,736K  5,968K
FREE AREA:    1,396K  4,160K LARGEST FREE:   1,320K  2,716K
-----

```

Reclaim 2,912K-1,736K-128K = 1,152K-128K = 1024K (64K multiples)

How do I make more storage available below 16MB?

§ If you have more than 1MB, how do you decide what to reclaim?

1. Unused SVA-24 - there is no parameter to change for this, *but leave at least 64K to allow for Supervisor expansion due to simple things like ADDing new devices.*
2. Unused Virtual Library (and do you really need all of the 24-bit phases you have loaded?).
3. Unused SVA Getvis-24, leaving *at least* 128K unused (*or whatever is safe for you*).

§ In this case I chose to ignore (2) as System Getvis had so much free storage:

1. Absorb 256K.
2. Do not change.
3. Reduce by 768K.

§ *When you IPL, check immediately and be prepared to change the values if you made a mistake.*

How do I make more storage available below 16MB?

§ After:

SVA GETVIS=(1280K,6M),PSIZE=(652K,7M)

MAP

| SPACE | AREA | V-SIZE | GETVIS | V-ADDR | UNUSED | NAME |
|-------|--------|--------|--------|---------------|------------|-------------------|
| S | SUP | 760K | | 0 | | \$\$A\$SUPX |
| S | SVA-24 | 1916K | 2188K | BE000 | <u>64K</u> | |
| 0 | BG V | 1536K | 10752K | <u>500000</u> | 249856K | Ç shared area 5MB |

| GETVIS USAGE | SVA-24 | SVA-ANY | SVA-24 | SVA-ANY |
|--------------|---------------|---------|-----------------|---------------|
| AREA SIZE: | <u>2,144K</u> | 9,704K | | |
| USED AREA: | 1,508K | 5,544K | MAX. EVER USED: | <u>1,724K</u> |
| FREE AREA: | 636K | 4,160K | LARGEST FREE: | <u>604K</u> |

CICS and GETVIS command output

- § The way that CICS acquires 24-bit GETVIS storage during initialization means that **all** 24-bit storage is acquired and then some of it is freed.
- § This causes the reported **xx-24 MAX. EVER USED** to be the same as the **AREA SIZE**, and you cannot calculate the true High-Water-Mark.
- § Enter command **GETVIS xx,RESET** after CICS has initialized.

CICS and GETVIS command output for xx-24

```

getvis g1
AR 0015 GETVIS USAGE      G1-24      G1-ANY      G1-24      G1-ANY
AR 0015  AREA SIZE:      11,260K    39,932K
AR 0015  USED AREA:      5,696K    34,364K MAX. EVER USED:  11,260K    39,932K
AR 0015  FREE AREA:      5,564K    5,568K LARGEST FREE:    5,456K    5,456K
AR 0015 DYNAMIC-SPACE GETVIS USAGE
AR 0015  AREA SIZE:      1,024K
AR 0015  USED AREA:      84K          MAX. EVER USED:      96K
AR 0015  FREE AREA:      940K          LARGEST FREE:      940K
AR 0015 1I40I  READY

```

- § GETVIS G1,RESET has probably not been done as MAX. EVER USED = AREA SIZE, so we don't know what the real HWM is.
- § We know we have used at least 5,696K, but more like 5,804K (11,260K-5,456K).
- § We have 5,456K contiguous.

CICS and GETVIS command output for xx-ANY

```

getvis g1
AR 0015 GETVIS USAGE      G1-24      G1-ANY      G1-24      G1-ANY
AR 0015  AREA SIZE:      11,260K    39,932K
AR 0015  USED AREA:       5,696K    34,364K MAX. EVER USED:  11,260K    39,932K
AR 0015  FREE AREA:       5,564K    5,568K LARGEST FREE:   5,456K    5,456K
AR 0015 DYNAMIC-SPACE GETVIS USAGE
AR 0015  AREA SIZE:       1,024K
AR 0015  USED AREA:        84K      MAX. EVER USED:    96K
AR 0015  FREE AREA:       940K      LARGEST FREE:     940K
AR 0015 1I40I  READY

```

- § **ANY includes both 31-bit and 24-bit storage.**
- § **We know we have used at most 39,932K.**
- § **That means we have used all of it at one point in time!**
- § **We have 5,456K contiguous, which is the same as the G1-24, so it must be 24-bit storage that is left.**

CICS and GETVIS command output

```
getvis g1
AR 0015 GETVIS USAGE      G1-24      G1-ANY      G1-24      G1-ANY
AR 0015 AREA SIZE:      10,236K    39,932K
AR 0015 USED AREA:      5,784K    31,540K MAX. EVER USED:    5,808K    31,584K
AR 0015 FREE AREA:      4,452K    8,392K LARGEST FREE:      4,440K    8,360K
AR 0015 DYNAMIC-SPACE GETVIS USAGE
AR 0015 AREA SIZE:      1,024K
AR 0015 USED AREA:      96K          MAX. EVER USED:      96K
AR 0015 FREE AREA:      928K          LARGEST FREE:      928K
AR 0015 1I40I  READY
```

§ Any comments on this data?

How much Space Getvis do I need?

§ Before CICS shutdown issue GETVIS xx:

```

getvis g1
AR 0015 GETVIS USAGE      G1-24      G1-ANY      G1-24      G1-ANY
AR 0015  AREA SIZE:      11,260K    39,932K
AR 0015  USED AREA:       5,696K    34,364K MAX. EVER USED:  11,260K    39,932K
AR 0015  FREE AREA:       5,564K    5,568K LARGEST FREE:      5,456K    5,456K
AR 0015 DYNAMIC-SPACE GETVIS USAGE
AR 0015  AREA SIZE:       1,024K
AR 0015  USED AREA:        84K      MAX. EVER USED:      96K
AR 0015  FREE AREA:       940K      LARGEST FREE:       940K
AR 0015 1I40I  READY

```

§ 1,024K is too big, I would set a minimum of 96K + 140K for a dump and round up for safety to 256K.

§ Don't just check it once, you need to measure *any* High-Water-Mark over a period of time to make sure that you have found the real one.

The CICS DSAs

§ (E)CDSA

- § CICS-key storage for control blocks.
- § CICS-key task storage.

§ (E)RDSA

- § Reentrant CICS nucleus and user programs (very good for cache performance).

§ (E)SDSA

- § CICS GETMAIN SHARED storage.
- § Non-reentrant CICS nucleus and user programs.

§ (E)UDSA

- § User-key task storage.

How does CICS use DSALIM and EDSALIM storage?

- § Each DSA grows over time in units of *extents* based on demand, and normally settles at a high-water-mark, although freeing a whole extent is supposed to be possible.
- § DSALIM is 256K per extent, and EDSALIM is 1MB per extent.
- § SIT overrides can set a fixed size for any of the 8 DSAs.
- § DFH0STAT, DFHSTUP and DFHPD410 DATA SM=1 give detailed usage.
- § Beware of OEM monitors that tell you that the "(E)DSA is nn% full", when what they mean is "nn% of the current DSA size, which is only mm% of the total (E)DSALIM limit".
- § **Always make sure that you understand what monitor output is telling you!**

How does CICS use DSALIM and EDSALIM storage?

§ CEMT I DSA shows the current usage:

I DSA

STATUS: RESULTS - OVERTYPE TO MODIFY

Sosstatus(Notsos)

| | | |
|-------------------------|------------------|----------------------------|
| Dsalimit(05242880) | DSALIM 5M | (you can change the value) |
| Cdsasize(00524288) | CDSA 0.5M in use | |
| Rdsasize(00524288) | RDSA 0.5M | |
| Sdsasize(00262144) | SDSA 0.25M | |
| Udsasize(00262144) | UDSA 0.25M | In use total 1.5M of 5M |
| Edsalimit(0450887680) | EDSALIM 430M | (you can change the value) |
| Ecdsasize(0375390208) | ECDSA 358M | |
| Erdsasize(0005242880) | ERDSA 5M | |
| Esdsasize(0001048576) | ESDSA 1M | |
| Eudsasize(0001048576) | EUDSA 1M | In use total 365M of 430M |

§ DSALIM and EDSALIM define the maximum available usage.

§ The entire DSALIM and EDSALIM values are allocated from GETVIS even if all of it is not used.

Who is using the Getvis Storage?

§ **GETVIS xx,ALL** - this is z/VSE 4.2 with **CICS** in red and others in black:

```

SUMMARY REPORT
SUBPOOL      REQUEST  <---G1-24-AREA---   ---G1-ANY-AREA-->
IMVS252      4,360K      15,728K  unallocated (E)DSALIM
IMVS130      768K        5,120K   allocated (E)DSALIM
Default      488K        3,264K   the world and his dog
IMVS000      152K        244K    default subpool 0
CELH24       40K         0K
IMVS132      36K         176K    KE domain stack
CELHAN       32K         20K
IMVS229      8K          32K
IMVS251      4K          44K
IMVS230      4K          0K
DFHEVP       4K          48K    emulation
IMVS229      0K          4K
USHEAP       0K          64K
USTKAN       0K          16K
SUBPOOL TOTALS      5,896K      24,760K
  
```

§ **SUBPOOL TOTALS** are less than are shown in the "GETVIS xx" output as they do not include the amount used for the GETVIS Control Information.

§ The next slide is z/VSE 5.1, you will see something very similar for z/VSE 4.3.

Who is using the Getvis Storage?

| SUBPOOL | REQUEST | <---F8-24-AREA--- | ---F8-ANY-AREA--> | |
|----------------|---------|-------------------|-------------------|-------------------------------------|
| IMVS129 | | 3,584K | 66,560K | unallocated (E)DSALIM |
| Default | | 3,144K | 3,212K | |
| IMVS000 | | 2,688K | 464K | default subpool 0 |
| IMVS252 | | 548K | 5,496K | (E)RDSA |
| IMVS130 | | 512K | 2,048K | allocated (E)DSALIM |
| IMVS130 | | 512K | 366,592K | allocated (E)DSALIM |
| CELH24 | | 80K | 0K | |
| IJBVSM | | 44K | 32K | VSAM default subpool |
| IMVS132 | | 40K | 212K | |
| CELHAN | | 28K | 36K | |
| IJBAU | | 24K | 460K | VSAM alternate index control blocks |
| IPNRSO | | 16K | 12K | |
| IMVS229 | | 8K | 32K | |
| IMVS254 | SVA | 4K | 0K | |
| IMVS253 | SVA | 4K | 0K | |
| IMVS230 | | 4K | 0K | |
| IMVS253 | SVA | 4K | 0K | |
| IMVS255 | SVA | 4K | 4K | |
| IJBCTG | | 4K | 8K | VSAM catalog management |
| DFHEVP | | 4K | 48K | |
| IMVS251 | | 0K | 44K | |
| IMVS229 | | 0K | 4K | |
| IJBPLH | | 0K | 4K | VSAM PLH etc. |
| USHEAP | | 0K | 128K | |
| USTKAN | | 0K | 32K | |
| IJBLSR | | 0K | 144K | VSAM LSR buffers etc. |
| IJBBUF | | 0K | 684K | VSAM NSR buffers |
| SUBPOOL TOTALS | | 11,240K | 446,260K | |

The main CICS IMVS Subpools

- § Some subpools may occur twice because they use different keys; USER key is always Storage Protection Key 9, and CICS key is the normal Partition Key.
- § Subpool 000 is for general CICS use; the 31-bit CICS Trace Table is allocated in this subpool, and can be a large amount of storage.
- § Subpool 129 is unallocated DSA extents if SIT STGPROT=YES.
- § Subpool 130 is allocated DSA storage, but (E)RDSA with SIT RENTPGM=PROTECT may use subpool 252.
- § Subpool 132 is CICS Nucleus Stack storage; this contains a save area and variables for each CICS module as it is executed for a CICS task.
- § Subpool 252 is unallocated DSA extents if SIT STGPROT=NO, and (E)RDSA with SIT RENTPGM=PROTECT.
- § **DSALIM and EDSALIM are mapped to multiple subpools, the total will stay the same, but the amount in each subpool can vary over time.**

I am running out of GETVIS, what do I do?

§ **GETVIS-31 is easy, either increase the partition size or reduce EDSALIM as in step (3) etc. substituting 1,024K.**

§ **For GETVIS-24:**

1. Reduce the Shared 24-bit area if you have not done that.
2. Reduce DSALIM? (Some customers do actually over-allocate DSALIM!)
3. Monitor DSALIM over time; e.g. use DFH0STAT with my Rexx code to produce a cumulative CSV file.
4. Decrease in 256K multiples, but leave at least 256K free to avoid a possible SOS.
5. You can experiment by reducing DSALIM in CEMT I DSA, but keep a CEMT task active in case it goes horribly wrong - the storage is released back to z/VSE.

I may have a GETVIS Storage Leak, what do I do?

- § Start by using **GETVIS xx,ALL** at regular intervals to see which subpools grow, but remember to add allocated and unallocated (E)DSALIM as one for comparison purposes.
- § **GETVIS xx,DETAIL** shows the actual storage address ranges as well.
- § The **z/VSE SHOW** command may identify the data at the addresses - **SHOW xx,address.hex_length**.
- § If you can identify the subpool or the data, contact the product supplier.
- § If it is the GETVIS default subpool, it could be any one of a million culprits, so use an SDAID GETVIS/FREEVIS trace - it is highly unlikely to be CICS.
- § SDAID is not helpful if it is a z/OS GETMAIN leak, you have to trace all SVC X'84's, i.e. all z/OS SVCs, and there are also PC instructions.
- § The good news is that I have never seen one (and don't ever want to either!).

I am running out of (E)DSALIM, what do I do?

§ EDSALIM is normally easy, the simple way is to make the partition bigger in units of 1MB and add the difference to SIT EDSALIM, or follow the process below with units of 1,024K to get it *now*.

§ For DSALIM:

1. Monitor GETVIS xx to see how much LARGEST FREE is available, round down to 256K multiples and don't use all of it.
2. Increase SIT DSALIM or increase it in CEMT I DSA now.

§ An example:

```
getvis v1
AR 0015 GETVIS USAGE      V1-24      V1-ANY      V1-24      V1-ANY
AR 0015 AREA SIZE:      11,260K    60,412K
AR 0015 USED AREA:      5,668K    30,992K MAX. EVER USED:    5,668K    40,320K
AR 0015 FREE AREA:      5,592K    29,420K LARGEST FREE:      5,592K    20,092K
```

What documentation do I need for SOS problems?

- § A CICS system dump (an SDUMP), *plus* SYSLOG *plus* SYSLST.
- § Please send all 3 items for a PMR.
- § CICS allows you to get an SDUMP on *any* DFH message, the message for SOS is either DFH**SM0131** below 16MB, or DFH**SM0133** above 16MB:

CEMT S SYD(SM013n) ADD SYS MAX(1)
- § This adds the entry to the in-storage Dump Table, add both if you want to.
- § CICS Service will look at the storage in detail, which will include how CICS is using its own internal subpools.
- § Use a PLTPI program to enter SPI command SET SYSDUMPCODE to make sure that you have standard CICS system dump handling every time you start it, as there are no configuration parameters to do this.

How do I get better use of DSALIM and EDSALIM?

§ The killer is normally DSALIM.

§ This is not an exhaustive list:

- § Look for PTFs that reduce 24-bit usage for all products that run under CICS, that includes OEM products.
- § Sometimes a product's behaviour changes between releases, always check Virtual Storage usage as part of you migration test plan, don't wait until it blows up in production.
- § Fully migrate to 31-bit programs.
- § Check Assembler PPT DATALOCATION.
- § Check PCT TASKDATALOC.
- § Do you really need LE ALL31(OFF)?
- § Check that LE HEAP() uses nK - 16 bytes to avoid using another page of UDSA.

How do I get better use of DSALIM and EDSALIM?

- § Consider selectively using AUXTRACE on transactions - CETR Special Trace.
- § Sample ABBREV AUXTRACE with AP=1,EI=1 and SM=1 show the returned addresses.
- § Application-initiated requests are bracketed by EIP entries.

```

00044 1 SM 0301 SMGF ENTRY GETMAIN 48,YES,00,TASK
00044 1 SM 0302 SMGF EXIT GETMAIN/OK 02F00448
00044 1 SM 0301 SMGF ENTRY GETMAIN 28,YES,RUWAPool,TASK31
00044 1 SM 0302 SMGF EXIT GETMAIN/OK 02F00448
00044 1 SM 0301 SMGF ENTRY GETMAIN 38EC,YES,00,LE_TWA,TASK31
00044 1 SM 0302 SMGF EXIT GETMAIN/OK 02F00488
00044 1 SM 0301 SMGF ENTRY GETMAIN 2AB8,YES,LE_RUWA,TASK31
00044 1 SM 0302 SMGF EXIT GETMAIN/OK 02F03D88
00044 1 SM 0301 SMGF ENTRY GETMAIN 7EA0,YES,LE_RUWA,TASK24
00044 1 SM 0302 SMGF EXIT GETMAIN/OK 006C0008
00044 1 AP 00E1 EIP ENTRY GETMAIN
00044 1 SM 0C01 SMMG ENTRY GETMAIN 338,YES,00,USER24,EXEC
00044 1 SM 0C02 SMMG EXIT GETMAIN/OK 006C7EB8
00044 1 AP 00E1 EIP EXIT GETMAIN OK

```


How much storage are my transactions using?

- § Use an OEM CICS performance monitor or CICS Monitor data.
- § Task DSA storage is allocated for EUDSA in 64K multiples, otherwise 4K.
- § You can also see it in a dump formatted with SM=1:

==SM: Transaction block summary

| SMX Address | Tran # | Tran Token | Data Key | Data Loc | Clear Stg | Freeze Stg | Remote Tran | C24 Address | SCA | U24 Address | SCA | C31 Address | SCA | U31 Address | SC |
|----------------|---------|---------------|-------------|-------------|--------------|---------------|----------------|----------------|-----|----------------|-----|----------------|-----|----------------|----|
| 0BE2B32C | 0014209 | 07D07D00 | CICS | Any | No | No | No | 0BE2F890 | | 0BE2F7DC | | 0BE2F728 | | 0BE2F674 | |

==SM: Task subpool summary

. . .

| SMX | Addr | Name | Id | Loc | Acc | Gets | Frees | Elms | Elemstg | Pagestg | |
|----------|----------|------|----|-----|-----|------|-------|------|---------|---------|-------|
| 0BE2B32C | M0014209 | 01 | B | C | 6 | 3 | 3 | 4208 | 8K | | CDSA |
| | C0014209 | 03 | A | C | 3 | 0 | 3 | 6448 | 12K | | ECDSA |
| | B0014209 | 02 | B | U | 0 | 0 | 0 | 0 | 0K | | UDSA |
| | U0014209 | 04 | A | U | 1 | 0 | 1 | 4016 | 64K | | EUDSA |

Any Questions ?