WAVV 2012 How to Monitor and Optimize CICS TS Storage

Mike Poil

CICS Level 3 Service

IBM Hursley

poilmike@uk.ibm.com

April 2014 update

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.

The ICCF CICS partition is a special case since ICCF has to allocate storage for its own purposes such as the 24-bit Interactive Partitions, and this is not described.

Vendor Software may also change the way that storage is allocated and managed.

Agenda

- Introduction and useful fixes
- 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?
- What about System Getvis-24?
- 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?
- Storage Leak or Creep?
- I may have a GETVIS storage leak, what do I do?
- I suspect a CICS DSA 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?

Introduction

- Apart from the 4K DFHSIP phase, every other bit of storage is allocated by z/VSE GETVIS/FREEVIS/CDLOAD services, and CICS makes great use of z/OS GETMAIN/FREEMAIN/STORAGE services as emulated by GETVIS/FREEVIS.

- GETVIS storage is suballocated as a number of named subpools, and each subpool owns a number of 4K pages.

- The SIT DSALIM and EDSALIM values are fully allocated in one IMVS GETVIS subpool, and storage is moved in 256K/1MB extents to a number of other IMVS subpools by CICS as it required for the 8 CICS DSAs.

- Each DSA comprises a series of named CICS subpools, and a subpool will expand or contract based on demand for element storage within that DSA.

- Some CICS subpools are for fixed-length "Quickcells", and unless you use CETR SM=1-3 or STNTRSM=(1,3) trace levels you will not see the CICS GETMAIN/FREEMAIN activity for them in CICS trace.

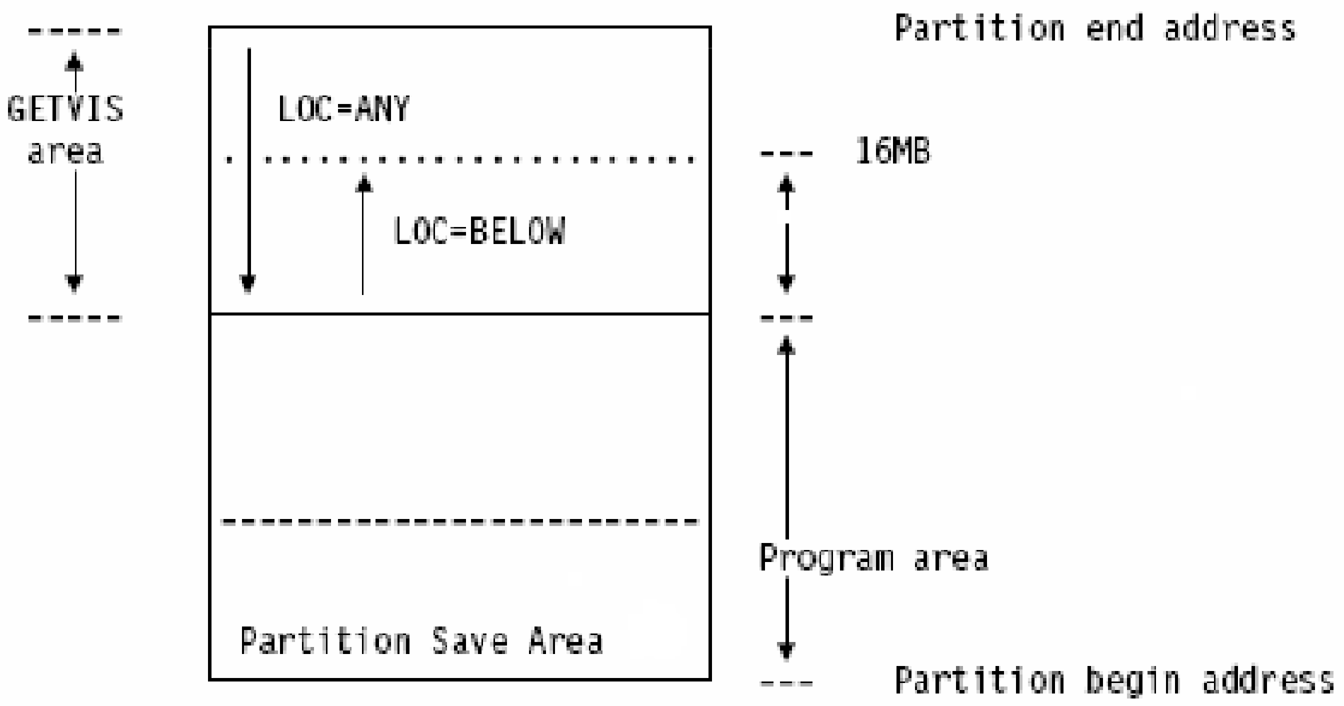
Useful Fixes

- A GETMAIN leak in VSAM can occur due to failed secondary extent allocations; for z/VSE 4.3 install the PTF for APAR DY47426, for 5.1 it is DY47427.
- VSAM EXCPAD problems can cause task delays and result in more concurrent tasks than is normal, this increases the CICS (E)DSA storage used and there is the potential for SOS; for z/VSE 4.3 install the PTFs for Hiper APARs DY47403 and DY47445, for 5.1 the Hiper APARs are DY47407 and DY47447.
- Install the PTF for PM75006 to help the CICS Explorer use less ECDSA storage.
- A GETVIS leak can occur after a CA split when using KSDS under CICS, for 5.1 install the PTF for Hiper APAR DY47452, which also fixes missing updates.
- DB2 Client APAR PM41022 resolves a storage leak in CICS subpool PGLLE.
- There is scope for changing CICS modules DFHETRX and DFHSUWT to use 31-bit CICS Kernel Stack Storage instead of 24-bit.

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 by z/VSE and CICS uses z/VSE SVC X'84'/132 or a PC instruction (e.g. STORAGE OBTAIN/RELEASE), but products such as VSAM use a subset via SVC X'83'/131, which includes GETMAIN/FREEMAIN.**
- **SVC 84 requires // EXEC ,OS390, but SVC 83 does not.**
- **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, 000 being the default.**
- **Some of the subpool numbers used by CICS are documented later.**
- **MVS Diagnosis: Reference GA22-7588 documents the z/OS SVCs and the subpools.**

How do I make more storage available below 16MB?

- **The simple answer may be that you can't, or need z/VSE 4.3 or 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. IODEV=1024, SYS SDSIZE, SVA SDL - I don't cover this in the presentation.

- **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 Vendor 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     384K      ←  Reclaim 384K-64K = 256K
0   BG V        1536K   10752K   600000  249856K      ←  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%)              ←  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     64K
0   BG V        1536K   10752K   500000  249856K       ←  shared area 5MB

------------------------------------------------------------------------------------
GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24    SVA-ANY
AREA SIZE:    2,144K     9,704K
USED AREA:    1,508K     5,544K MAX. EVER USED:    1,724K     5,968K
FREE AREA:      636K     4,160K LARGEST FREE:        604K     2,716K
```

CICS and GETVIS command output

- Please note that GETVIS command output shows usage/free in units of 4K pages, the actual amount of *allocated* storage will be less than is shown.
- 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 shortly 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-24

```
getvis g1,reset
AR 0015 GETVIS USAGE    G1-24     G1-ANY                     G1-24     G1-ANY
AR 0015  AREA SIZE:   11,260K    39,932K
AR 0015  USED AREA:    5,676K    33,624K MAX. EVER USED:    5,688K     33,668K
AR 0015  FREE AREA:    5,584K     6,308K LARGEST FREE:       5,572K     6,264K
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 performed on a different execution of CICS in G1.
- DSALIM=5M.
- The USED AREA 5,676K is not normally a lot more than DSALIM, unless you have ICCF and a number of 24-bit ICCF interactive partitions.

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, which 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.**
- **Note: DFH0STAT shows 24-bit and 31-bit storage, not 24-bit and both.**

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.

What about System Getvis-24?

- CICS is just like any program and z/VSE will allocate System Getvis-24 for certain data areas and control blocks to support its operation.

- Outside of MRO, which uses the DEFAULT subpool, the CICS code has very few requests for SVA storage.

- If you have an issue with System Getvis usage, you need to take it up with z/VSE Support.

The CICS DSAs

- **(E)CDSA**
  - CICS-key storage for control blocks and CICS-key program task storage.
  - CICS-key non-reentrant programs.
- **(E)RDSA**
  - Reentrant CICS nucleus and user programs (very good for cache performance).
- **(E)SDSA**
  - CICS GETMAIN SHARED storage.
  - Non-reentrant User-key programs.
- **(E)UDSA**
  - User-key task storage.

How does CICS use DSALIM and EDSALIM storage?

- SIT DSALIM and EDSALIM allocate that amount of storage for CICS to use for the DSAs from Partition Getvis during CICS initialisation, and remains allocated even if all of it is not used unless CEMT is used to reduce xDSALIM.
- Each DSA grows over time in units of *extents* based on demand, and normally settles at a high-water-mark; an empty extent is not normally freed, it can only be moved to another DSA to avoid an SOS.
- DSALIM is 256K per extent, and EDSALIM is 1MB per extent.
- DFH0STAT, DFHSTUP and DFHPD410 DATA SM=1 give detailed usage.
- Beware of Vendor 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 show the maximum available usage.**

Who is using the Getvis Storage?

- GETVIS xx,ALL - this is z/VSE 4.2 with CICS in red and others in black; IMVS000 is used by CICS and other products:

```
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 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   KE stack
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 CICS subpools may occur more than once in the display, this is normally due to different storage keys being used.
- Subpool 000 is for general use by CICS and other products; the CICS Trace Table uses this and can be big, and the CICS Explorer may also use a lot.
- Subpool 129 is unallocated DSA extents if SIT STGPROT=YES.
- Subpool 130 is allocated DSA storage, but (E)RDSA with SIT RENTPGM=PROTECT uses 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; subpool 252 can be used for other things.
- 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!)

   - Monitor DSALIM over time; e.g. use DFH0STAT with my Rexx code to produce a cumulative CSV file.

   - Decrease in 256K multiples, but leave at least 256K free to avoid a possible SOS.

   - 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.

Storage Leak or Creep?

- **Both of these have similar symptoms, and can result in all available storage being used given the right prevailing conditions and time.**
- **A "leak" is a bug that results from a control block or data area being allocated but not (always) freed, and is what I would assume until proven otherwise.**
- **"Creep":**
  - Is from fragmentation within the storage used due to a pattern of requests to allocate and free storage that does not fit well with the design of the storage management code.
  - The storage high-water-mark for "used" continually grows, and so does the amount of "free" storage, although not necessarily at the same rate.
  - Can be very difficult to spot as the software may either not provide this level of information or may require manual analysis of printed dump data and control blocks.
  - Cannot be tuned out unless you have appropriate external control over the storage management software or the size/pattern of the requests.

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

- GETVIS xx,RESET than GETVIS xx,DETAIL at regular intervals to see which subpools grow, but remember to sum allocated and unallocated (E)DSALIM as one for comparison purposes (GETVIS xx,ALL shows the subpool sizes).
- The z/VSE SHOW command may identify the data at the addresses - SHOW xx,address.hex_length_max_1000.
- 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, but it is unlikely to be CICS.
- SDAID is not helpful for a z/OS GETMAIN leak in an IMVS subpool, you need to TRACE SVC=(83,84) but only look at the formatted MVS-SVC=4, 10 and 120 entries, and there are also PC instructions (INSTR=B218); the analysis is a task for CICS Support to do once you have run SDAID and used DOSVSDMP option 4 to format the trace.

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

- **We need a dump of the partition at STOPSD time to trace the SDAID addresses back to the code being used - you *must* use DUMP xx,0-7FFFFFFF,cuu if you don't use a CEMT P SNAP.**

- **Here is an example of the required SDAID job to check CICS running in C2:**

```
// EXEC SDAID
OUTDEV  T=580
TRACE INSTR=B218  AREA=C2 ADDR=0:7FFFFFFF OUTPUT=(GREG)
TRACE SVC=(83,84) AREA=C2 ADDR=0:7FFFFFFF OUTPUT=(GREG)
/*
```

- **The register contents on entry to the SVCs 4,10 and 120 (all you get with SDAID) are described in z/OS MVS Diagnosis Reference GA22-7588 under SVC 120 (0A78).**

- **The PC call with R14=0000030B/00000311 for STORAGE OBTAIN/RELEASE has the same register usage; there is less overhead if this trace is not used.**

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

- Here is SDAID output that showed a GETMAIN leak at address 10944822 in phase IKQNEX without a FREEMAIN (APAR DY47426 or DY47427):

```
SVC     C2 C2   C00   SVC=83   ADDR=10944822  R00=000000F0 R01=00000000 R15=00000002
                  MVS-SVC=78   (SIMULATED SVC)

GR 0-7    000000F0 00000000 109461E8 909447C0 06692100 030AA670 00000000 1096D890

   8-F    00000000 0B400000 030AA3B8 030AA018 109457BF 034B0FD0 000118F0 00000002
```

- R0 is the number of bytes to GETMAIN/FREEMAIN, R1 is the address for FREEMAIN, and the R15 bytes are:

```
0 Options
1 Key for special subpools only
2 Subpool number
3 Option byte:
0... .... Reserved - Ignored, should be zero.
.1.. .... Storage can be backed anywhere.
..00 .... Storage should have residency of caller.
..01 .... Storage address must be 24 bits.
..11 .... Storage address valid to full 31 bits.
.... 1... Request is variable.
.... .1.. Storage should be on page boundary.
.... ..1. Request is unconditional.
.... ...1 Request is a FREEMAIN (0=GETMAIN - odd value=FREEMAIN, even=GETMAIN).
```

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

- **Well-behaved Vendor code that does a GETMAIN and a FREEMAIN:**

```
SVC    C2 C2  C00  SVC=83  ADDR=002F075C  R00=000004BC R01=00000000 R15=00000000
                  MVS-SVC=78  (SIMULATED SVC)

GR 0-7   000004BC 00000000 00000000 00328CB8 13D90A10 002F06B8 03094E00 00332940

   8-F   00331008 030A3608 13D9042C 13EC13A2 13EB9478 00328CB8 000118F0 00000000

SVC    C2 C2  C00  SVC=83  ADDR=002F0852  R00=000004BC R01=008A8900 R15=00000003
                  MVS-SVC=78  (SIMULATED SVC)                FREEMAIN address

GR 0-7   000004BC 008A8900 00328CB8 93EC14CA 13D90A10 002F06B8 03094E00 00332940
  8-F   00331008 030A3608 13D9042C 13EC13A2 13EB9478 00328CB8 000118F0 00000003
```

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

- If you want to look at the z/VSE GETVIS and FREEVIS activity, this is what you should use and you will need a dump of the partition as well.
- Using SDAID to trace SVCs 3D (GETVIS) and 3E (FREEVIS) does not show you the address of the storage obtained by GETVIS and misses the use of the internal SGETVIS and SFREEVIS requests used by the Supervisor.
- When you use AREA=xx, you ***must*** add the ADDR operand or SDAID will ***not*** trace SVA code activity; the ADDR option is not documented as being available when you look at the Diagnosis Tools manual.
- Sample output is shown on the next slide.

```
// EXEC SDAID
OUTDEV  T=580
TRACE GETVIS=PARTITION AREA=BG ADDR=0:7FFFFFFF
TRACE GETVIS=SPACE     AREA=BG ADDR=0:7FFFFFFF
TRACE GETVIS=SVA       AREA=BG ADDR=0:7FFFFFFF
/*
```

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

- *MYPROG is the name of the phase running in BG*.
- ADDR=061DB98A is in the 31-bit SVA, and I know that the phase is called GETVIS1 as I wrote it.
- You will need a LIBR LD SDL to find the names of SVA-resident phases.

```
GETVIS 0  BG  C01  TYPE=SPC  RC=00  OPT=1B4C  ST_ADDR=0036A000 L=0000004E  SUBPOOL_NAME (EBCDIC) = IJBFBG
                     GETVIS R15 REQUEST OPTIONS ADDRESS OF STORAGE AND LENGTH
              ADDR=00013500  PROGR=""""""""         ---SGETVIS REQUEST---            (HEX)    = C9D1C2C6C2C7
              <ADDR=ADDRESS OF THE REQUESTING INSTRUCTION>
GETVIS 0  BG  C01  TYPE=SVA  RC=00  OPT=1B4C  ST_ADDR=0036A000 L=0000004E  SUBPOOL_NAME (EBCDIC) = IJBFBG
              ADDR=00013500  PROGR=""""""""         ---SGETVIS REQUEST---            (HEX)    = C9D1C2C6C2C7
FREEVS 0  BG  C01  TYPE=     RC=00  OPT=220C  --ENTIRE STORAGE RELEASED--
              ADDR=00509C8A  PROGR=$JOBCTLE         PSW=070D0000 00509C8C
GETVIS 0  BG  C01  TYPE=PAR  RC=00  OPT=1000  ST_ADDR=00640080 L=00000000  SUBPOOL_NAME (EBCDIC) = DEFAULT
              ADDR=000CBD90  PROGR=""""""""         PSW=070D0000 000CBD92            (HEX)    = 000000000000
GETVIS 0  BG  C00  TYPE=PAR  RC=00  OPT=1000  ST_ADDR=00640080 L=00003000  SUBPOOL_NAME (EBCDIC) = DEFAULT
              ADDR=0050008C  PROGR=MYPROG           PSW=071D0000 8050008E            (HEX)    = 000000000000
FREEVS 0  BG  C00  TYPE=PAR  RC=00  OPT=1000  ST_ADDR=00640080 L=00001800  SUBPOOL_NAME (EBCDIC) = DEFAULT
              ADDR=0050009E  PROGR=MYPROG           PSW=071D0000 805000A0            (HEX)    = 000000000000
GETVIS 0  BG  C01  TYPE=PAR  RC=00  OPT=1000  ST_ADDR=00643080 L=00003000  SUBPOOL_NAME (EBCDIC) = DEFAULT
              ADDR=061DB978  PROGR=MYPROG           PSW=071D0000 861DB97A            (HEX)    = 000000000000
FREEVS 0  BG  C01  TYPE=PAR  RC=00  OPT=1000  ST_ADDR=00643080 L=00003000  SUBPOOL_NAME (EBCDIC) = DEFAULT
              ADDR=061DB98A  PROGR=MYPROG           PSW=071D0000 861DB98C            (HEX)    = 000000000000
FREEVS 0  BG  C01  TYPE=     RC=00  OPT=220C  --ENTIRE STORAGE RELEASED--
              ADDR=000EE424  PROGR=$IJBSEOT         PSW=070D0000 000EE426
```

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

- EDSALIM is normally easy, the simplest way is to make the partition bigger (assuming that SYS PASIZE allows it) 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*.
- To increase DSALIM you need FREE contiguous GETVIS in units of 256K, and you see that in GETVIS command output.
- The smaller the amount of free GETVIS, the more careful you need to be about how representative the GETVIS command output is.
- For example, if the output shows several MB free, it would be safe to use a reasonable % of that, as another 512K may be all that you need.
- You should use GETVIS xx,RESET as soon as possible after CICS initialises to reset the 24-bit MAX. EVER USED high-water-mark.
- Use GETVIS xx as close as possible to shutdown to get the data that you need.
- Repeat this over several runs of CICS to find a representative amount of free storage if the displayed amount is small.

I suspect a CICS DSA leak, what do I do?

- Monitor using DFH0STAT and/or CEMT I DSA.
- Use CEMT P SNAP to get a baseline.
- Wait until you see a useful amount of growth.
- Use CEMT P SNAP to get new dump.
- DFHPD410 DATA SM=1 on both dumps will show the subpool usage if you want to take a look yourself.
- Send IBM the raw dumps.
- Growth in subpool PGLLE can be caused by code doing EXEC CICS LOAD <HOLD> without an EXEC CICS RELEASE.

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

- In this example, LARGEST FREE shows that more than 5MB of contiguous storage could potentially be used.
- AREA SIZE - MAX. EVER USED of 5,592K shows that LARGEST FREE is slightly bigger than is safe to use, but you must have done a GETVIS xx,RESET to check this.
- Never use all of it or you could see GETVIS errors.
- You could start with 1MB and increase as required.
- Use CEMT I DSA to dynamically change and test the results.

```
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?

- *If it is not a simple case that DSALIM is too small*, set CICS trace with level 1 for all components and at least a 4096K trace table (use 8192K if possible).
- A CICS system dump (e.g. CEMT P SNAP) for comparison when the system has been busy but was not at SOS, a dump at SOS *plus* SYSLOG *plus* SYSLST.
- *A dump after SOS is too late and is often not conclusive.*
- CICS allows you to get an SDUMP on *any* DFH message, the message for SOS is either DFHSM0131 below 16MB, or DFHSM0133 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 can look at the trace and storage in detail (DFHPD410 SM), which will show how CICS is using its own internal subpools.
- Use a PLTPI program to enter SPI SET SYSDUMPCODE commands.

How do I get better use of DSALIM and EDSALIM?

- **This is not an exhaustive list:**
  - Look for PTFs or zaps that reduce 24-bit usage.
  - 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.
  - Are you using SIT TCTUALOC=BELOW when it could be set/deafaulted to ABOVE?
  - Do you really need LE ALL31(OFF)?
  - Check that LE HEAP() uses sizes that are nK - 16 to avoid going over a page boundary in the (E)UDSA once the 16-byte check zones are added by CICS.

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 requested hex size and 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 a Vendor 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    Tran #  Tran     Data Data  Clear Freeze Remote C24 SCA  U24 SCA  C31 SCA  U31 SC
  Address          Token    Key   Loc   Stg   Stg    Tran  Address  Address  Address  Address
 . . .
 0BE2B32C 0014209 07D07D00  CICS Any    No    No     No    0BE2F890 0BE2F7DC 0BE2F728 0BE2F674

==SM: Task subpool summary

 . . .

 SMX Addr Name     Id Loc Acc   Gets  Frees  Elems  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
```

Q & A

- If you have questions after today, feel free to email me.