IBM Software Group

# CICS Storage 102 - Debugging CICS Short on Storage

Sarah Bertram (bertrams@us.ibm.com) Erich Hoppe (ephoppe@us.ibm.com)
CICS Level 2 Support
10 August 2011

WebSphere® Support Technical Exchange

# Agenda

- Review: What you should already know
- What is Short on Storage?
- Storage Sympathy Sickness
- Actions taken by CICS
- General Problem Analysis
- Debugging Short on Storage dumps
- Storage Tuning Parameters

# Review: What You Should Already Know

- Prereq: CICS Storage 101 – A look into CICS Dynamic Storage Areas (WebSphere Technical Exchange, November 2010), currently available for replay at:
  - http://www-01.ibm.com/support/docview.wss?uid=swg27020167
- Topics Discussed:
  - Various DSAs in CICS
  - SIT parameters that affect storage
  - Control blocks that manage DSA
- In this presentation we will be using some of these control blocks and concepts to debug CICS short on storage (SOS) conditions.

# Short on Storage Preface

- There are many different flavors of SOS conditions and many different ways to approach them.
- In this presentation, we will show you steps we take to diagnosis these types of problems, but note that the examples are relatively simple compared to some of our customer's complex environments.
- When we are diagnosing SOS dumps, one difficulty we face is not knowing what normal storage usage is in the CICS region.

# What is short on storage?

- When a getmain request for CICS DSA storage cannot be satisfied.
- CICS will issue one of the following messages when it goes short on storage:
  - DFHSM0131 - *applid* CICS is under stress (short on storage below 16MB).
  - DFHSM0133 - *applid* CICS is under stress (short on storage above 16MB).
- CICS can go short on storage independently in any DSA. You may see tasks suspended on any of the following resource types: CDSA, SDSA, RDSA, UDSA, ECDSA, ESDSA, ERDSA or EUDSA.

# Short on Storage - continued

- CICS will report a individual DSA as being SOS, however this does not necessarily mean *that* DSA is the source of the SOS.
- Examples:
  1. The UDSA could be flagged as SOS, but the CDSA is using the majority of the DSA.
     - Start by examining the CDSA storage usage first.
  2. The EUDSA is reporting SOS, but when you look at the dump, you find plenty of free storage. There is 250M of unallocated EDSA.
     - This would indicate that the getmain request is greater than 250M of storage. Investigate if this is an incorrect length.

# Storage Sympathy Sickness

- Many times when CICS goes SOS, it is due to other situations.
  - Looping task.
  - Network slow down.
  - Resource contention.
  - Storage overlay.
  - Poor Tuning.

# Actions taken by CICS

- It will look at the particular DSA to see if the largest free area can satisfy the request.
- If available will allocate new unused extent(s) to the particular DSA.
- Steal an extent from another DSA to satisfy the request.
  - The extent must be completely free to be eligible to be stolen.
- Perform program compression.
  - Any not-in-use, nonresident programs will be deleted possibly freeing an extent.

# Actions taken by CICS – Continued

- If these steps fail to free up storage allowing the getmain to satisified, then CICS goes SOS.
- When SOS, no new tasks will be attached.
- CICS will allow tasks currently in the system to continue running.
  - This allows tasks to complete, which would free up storage related to that task, which could possibly satisfy the outstanding getmain request.
- When CICS is no longer short on storage, one of the following messages will be issued:
  - DFHSM0132 - *applid* CICS is no longer short on storage below 16MB.
  - DFHSM0134 - *applid* CICS is no longer short on storage above 16MB.

# General Dump Analysis

1. Get a dump preferably on the DFHSM0131 or DFHSM0133 message, but a console dump at the time will also suffice.
2. Format out the SM domain (VERBX DFHPD660 'SM')
3. Examine the current limit and total for the DSA/EDSA that is flagged as SOS.
4. Capture the size of storage allocated to the individual DSA/EDSAs.
5. You will want to focus on the individual DSA that is using the most storage

# General Dump Analysis – notes

- Use the CICS command CEMT SET SYDUMPCODE(SM0131) SYSDUMP MAXIMUM(1) ADD to indicate that a dump should be taken the first time a DFHSM0131 message is issued.
- If you are out of storage below the line, capture the size of the:
  - UDSA, CDSA, SDSA, RDSA
- If you are out of storage above the line, capture the size of the:
  - ECDSA, EUDSA, ESDSA, ERDSA
- If no DSA stands out, this is a case where its helpful to understand what is *normal* for the region. This may be a case of sympathy sickness and storage is victim of another problem.

# General Dump Analysis – continued

6. Use Storage Control blocks to further understand the storage usage in the suspect DSA.
   A. If the suspect DSA is UDSA/EUDSA, then the problem involves task related storage.
      a) You will want to look at the Task subpool summary to analyze storage associated with each task.

   B. Otherwise, you'll need to look at the Domain subpool summary for the suspect DSA to see if any particular subpool is using the majority of that DSA's size.

# Debugging Short on Storage Dumps

# Example 1

- We will look at a system dump taken for message: DFHSM0133
- This example will show a classic short on storage condition caused by particular subpool that has grown very large.

# Example 1 – Storage Manager Summary

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

Current DSA limit:                      6144K
Current DSA total:                      1280K
Currently SOS below 16M:                   NO

Current EDSA limit:                      150M
Current EDSA total:                      149M
Currently SOS above 16M:                  YES
```

# Example 1 – Storage Manager Summary Notes

- When debugging a dump taken for a short on storage, the first domain I look at is the storage domain using:
  - Verbx DFHPD660 ‘SM’
- The summary shows us that the EDSALIM has been set to 150M and we have allocated 149M and it is currently SOS.
  - This shows us that there is a 1M extent of EDSA that has not been allocated yet, this implies the task must be getmaining storage larger than 1M since the request could not be satisfied.
- We will now look at the ECDSA, EUDSA, ESDSA and ERDSA.

# Example 1– Individual DSA Summaries

```
==SM: ECDSA Summary
   Size:                          103424K
   Current free space:               260K  ( 0%)
   Largest free area:                260K
   Currently SOS:                     NO
==SM: EUDSA Summary
   Size:                           17408K
   Current free space:              1664K  ( 9%)
   Largest free area:                192K
   Currently SOS:                    YES
==SM: ESDSA Summary
   Size:                            1024K
   Current free space:               956K  (93%)
   Largest free area:                956K
   Currently SOS:                     NO
==SM: ERDSA Summary
   Size:                           30720K
   Current free space:               980K  ( 3%)
   Largest free area:                392K
   Currently SOS:                     NO
```

# Example 1 – Individual DSA Summaries notes

- We extract the pertinent information from the dump regarding the individual DSA summaries:
  - size, current free space, largest free space and currently SOS flag
- From this, we can clearly see that the ECDSA is using the greatest amount of storage even though it is currently *not* SOS.
- 101M of storage has been allocated to the ECDSA of the 150M available.
- The EUDSA is currently SOS which indicates that someone is requesting task storage. Since the largest free area in this DSA is 192K, we know the request is greater than that.
- The largest free area in any of the DSAs is less 1M which means there are no free extents that can be stolen.

# Example 1 – Subpool Summaries

```
==SM: Domain subpool summary (ECDSA)

   Name        Id Chn  Initf Bndry Fxlen Q-c    Gets   Frees  Elems  Elemstg Pagestg
   TSMAIN    01A1          4K    64    64  Y    45165       0  45165 79690560  79060K
   TSTSI     011F          4K     8     8  Y    45165       0  45165  9961320   9844K
```

The description of these subpool names can be found in the performance guide. We can tell that these 2 subpools involve temporary storage as the performance guides lists these subpools as follows:

- TSMAIN - contains storage for temporary storage main storage.
  The subpool might be reduced by using auxiliary temporary storage.

- TSTSI - contains TS item descriptors.

# Example 1 – Subpool Summaries notes

- Since the ECDSA is the largest, we went to the domain subpool summary for the ECDSA to identify the largest users.
- We found that the largest users were the following subpools:
  - TSMAIN and TSTSI
- These 2 temporary storage subpools are using a combined 87M of the 101M allocated to the ECDSA
- This leads us to believe that this is the culprit of the SOS.
  - Temporary Storage needs to be managed by the application and is not automatically cleaned up by CICS.
- The EUDSA being flagged as SOS is a victim of the extensive temporary storage usage.

# Example 1 – Temporary Storage

```
=TS: Main statistics
  * Put/Putq main storage requests:          1245165
  * Get/Getq main storage requests:                0
  * Current storage for TS main:            31129125
  * Peak storage for TS main:               31129125

==TS: Queue summary

Queue      Total  Total  Tran  Creation
Name        Size  Items  Id    Date/Time
TSQ 0000   82365   4845  TSQL  12/07/11 18:59:48
TSQ 0001   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0002   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0003   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0004   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0005   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0006   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0007   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0008   82688   4864  TSQL  12/07/11 18:59:48
TSQ 0009   82688   4864  TSQL  12/07/11 18:59:48
```

# Example 1 – Temporary Storage notes

- Since we know the problem is related to temporary storage, we went to the temporary storage domain by issuing the following
  - verbx dfhpd660 'ts'
- We can see in the TS: Main statistics, that the current storage matches the peak which implies that the storage keeps growing.
- In Queue summary, we can see a pattern that shows transaction TSQL has written to many queues. We would recommend this application be investigated to be sure it is deleting its temporary storage queues.

# Example 2

- This dump was taken for a DFHSM0133 message.
- We will see a build up of tasks that accounts for the increased storage usage leading to the SOS.

# Example 2 – Storage Domain Summary

```
SM Domain status:                  INITIALISED
Storage recovery:                  YES
Storage protection requested:      NO
Storage protection active:         NO
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:                   200M
Current EDSA total:                   199M
Currently SOS above 16M:               YES
```

# Example 2 – Individual DSA Summaries

```
==SM: ECDSA Summary
   Size:                          15360K
   Current free space:              740K  ( 4%)
   Largest free area:               700K
   Currently SOS:                     NO
==SM: EUDSA Summary
   Size:                         156672K
   Current free space:            10368K  ( 6%)
   Largest free area:               192K
   Currently SOS:                   YES
==SM: ESDSA Summary
   Size:                           1024K
   Current free space:              956K  (93%)
   Largest free area:               956K
   Currently SOS:                     NO
==SM: ERDSA Summary
   Size:                          30720K
   Current free space:              980K  ( 3%)
   Largest free area:               392K
   Currently SOS:                     NO
```

# Example 2 – Individual DSA Summaries notes

- We extract the pertinent information from the dump regarding the individual DSA summaries.
- From this, we can clearly see that the EUDSA is using the greatest amount of storage and is currently SOS.
- 153M of storage has been allocated to the EUDSA of the 200M total available.
- Notice that there is 10M of free storage in the EUDSA. However, the largest contiguous free area in this DSA is 192K. Because of this, CICS will not be able to satisfy a getmain request for more than 192K even though there is 10M free.

# Example 2 – EUDSA Extents

```
Extent list:      Start      End          Size     Free

                  15600000   156FFFFF     1024K    312K
                  16F00000   170FFFFF     2048K      0K
                  17100000   172FFFFF     2048K      0K
. . .
. . .
                  19900000   19AFFFFF     2048K      0K
                  19B00000   19CFFFFF     2048K    192K
                  19D00000   19EFFFFF     2048K    192K
                  19F00000   1A0FFFFF     2048K    192K
                  1A100000   1A2FFFFF     2048K    192K
                  1A300000   1A4FFFFF     2048K    192K
                  1A500000   1A6FFFFF     2048K    192K
                  1A700000   1A8FFFFF     2048K    192K
                  1A900000   1AAFFFFF     2048K    192K
```

# Example 2 – EUDSA Extents notes

- Because there was 10M of free storage in the EUDSA, we wanted to look at the extent list which shows the amount of free storage per extent.
- We see a pattern of 2M extents, each with 192K of free storage.

# Example 2 – Suspend queue

```
==SM: Suspend queue summary

   KE Task   Tran #   Susptok   Subpool   DSA     Request

15044700 0000149 068C0001 U0000149 EUDSA   1835024
```

- The suspend queue summary shows what task number is requesting storage and for what subpool name. In this case task 00149 is requesting decimal 1835024 bytes of task subpool U0000149 storage.
- The amount 1835024 (1.75 M) requires a contiguous 2M extent or greater to satisfy this request.

# Example 2 – Task subpool summary

```
==SM: Task subpool summary

   Current number of tasks:     103


SMX Addr Name       Id Loc Acc    Gets   Frees  Elems  Elemstg Pagestg Tran
20834064 M0000003 0001  B   C        0       0      0        0      0K CSOL
         C0000003 0003  A   C        2       0      2     2000      4K
         B0000003 0002  B   C        0       0      0        0      0K
         U0000003 0004  A   C        0       0      0        0      0K
208340A8 M0000005 0001  B   C        0       0      0        0      0K CEPM
         C0000005 0003  A   C        2       0      2     2352      4K
         B0000005 0002  B   C        0       0      0        0      0K
         U0000005 0004  A   C        0       0      0        0      0K
...
...
20834680 M0000073 0001  B   C        0       0      0        0      0K SET1
         C0000073 0003  A   C        0       0      0        0      0K
         B0000073 0002  B   C        1       0      1      928      4K
         U0000073 0004  A   C        4       2      2  1836608   1920K
208346C4 M0000074 0001  B   C        0       0      0        0      0K SET1
         C0000074 0003  A   C        0       0      0        0      0K
         B0000074 0002  B   C        1       0      1      928      4K
         U0000074 0004  A   C        2       0      2  1836608   1920K
20834708 M0000075 0001  B   C        0       0      0        0      0K SET1
         C0000075 0003  A   C        0       0      0        0      0K
         B0000075 0002  B   C        1       0      1      928      4K
         U0000075 0004  A   C        2       0      2  1836608   1920K
```

# Example 2 – Task subpool Summary notes

- Looking at the task subpool summary, first thing we noticed was a fairly large number of tasks in the system, 103.
- Scanning down and focusing on the Pagestg column, we are looking for a task using an unusually large amount of storage.
- Rather than 1 task using a large amount of storage, we see there are many tasks (transaction SET1) using the same amount of storage, 1920K.
- There are 76 SET1 tasks that each have 1920K of storage which totals 142M.

# Example 2 – XM Domain

```
==XM: MXT SUMMARY

   Maximum user tasks (MXT):                150
   System currently at MXT:                 No
   Current active user tasks:               88
   Current queued user tasks:               0
 * Peak active user tasks:                  88
 * Peak queued user tasks:                  0
 * Times at MXT limit:                      0
```

# Example 2 – XM Domain notes

- Since we suspect the SOS is due to the volume of tasks in the system, we went to the XM domain to check if we are max task.
  - verbx dfhpd660 'xm=1'
- The MXT Summary at the bottom shows us that we are not at max task though the current active users is the peak number which indicates that there could be more tasks than normal.

# Example 2 – DS Summary

```
RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT      DTA      M  SUSPAREA XM_TXN_TOKEN
TYPE                        SUSPEND      DUE          (DSTSK)
SOCKET   RECEIVE         S 18:21:44.061      -       208C5B00 SO 208C5B00 16E4E9000000150C
SOCKET   RECEIVE         S 18:17:28.333      -       208C6080 SO 208C6080 16E373000000133C
SOCKET   RECEIVE         S 18:17:29.381      -       208C6200 SO 208C6200 16E375000000134C
SOCKET   RECEIVE         S 18:16:30.429      -       208C6380 SO 208C6380 16E377000000135C
SOCKET   RECEIVE         S 18:15:31.478      -       208C6500 SO 208C6500 16E379000000136C
EUDSA                    S 18:29:45.110      -       16E4E700 QR 208C5980 16E4E7000000149C
SOCKET   RECEIVE         S 18:19:32.527      -       208C6680 SO 208C6680 16E37B000000137C
SOCKET   RECEIVE         S 18:19:33.583      -       208C6800 SO 208C6800 16E37D000000138C
SOCKET   RECEIVE         S 18:18:34.625      -       208C6980 SO 208C6980 16E421000000139C
SOCKET   RECEIVE         S 18:23:35.672      -       208C6B00 SO 208C6B00 16E423000000140C
SOCKET   RECEIVE         S 18:18:36.721      -       208C6C80 SO 208C6C80 16E425000000141C
```

```
Dump was taken at:
TIME OF DAY CLOCK: C80F4693 780BD60C  07/12/2011 19:29:46.148541 local
TIME OF DAY CLOCK: C80F392A 3DCBD60C  07/12/2011 18:29:46.148541 GMT
```

# Example 2 – DS Summary Notes

- Since we now believe the SOS is due to the volume of the tasks, we went to the dispatcher domain to see what resources the tasks are suspend on and for how long
  - verbx dfhpd660 ‘ds=1’
- We noticed that theses tasks are suspended in a socket receive wait for over 10 minutes. This would indicate a possible network problem or a problem with the client.
  - Use IP ST SYS to get the time of the dump

# Example 2 – Summary

- To recap, this dump shows us that we are SOS in the EUDSA. This was due to a large number of tasks in the system.
- Some suggestions for this type of SOS
  - To address the socket receive, check for network or client issues.
  - To throttle the number of SET1 transaction, put SET1 in a Transaction class limiting the number that can be run at one time.
  - To increase the amount of EDSA, adjust your EDSALIM allowing more tasks to run at once.

# Example 3

- Operator receives complaints from users about their terminals hanging.
- The operator noticed the region is short on storage below the line and takes a console dump of the region.

# Example 3 – Storage Manager Summary

```
SM Domain status:                    INITIALISED
Storage recovery:                    YES
Storage protection requested:        YES
Storage protection active:           YES
Reentrant program option:            PROTECT
Transaction isolation requested:     YES
Transaction isolation active:        YES

Current DSA limit:                     5120K
Current DSA total:                     5120K
Currently SOS below 16M:                 YES

Current EDSA limit:                     150M
Current EDSA total:                      51M
Currently SOS above 16M:                  NO
```

# Example 3 – Individual DSA Summaries

```
==SM: UDSA Summary
   Size:                          1024K
   Current free space:            1020K  (99%)
   Largest free area:             1016K
   Currently SOS:                   NO
==SM: CDSA Summary
   Size:                           512K
   Current free space:              72K  (14%)
   Largest free area:               24K
   Currently SOS:                   NO
==SM: SDSA Summary
   Size:                          3328K
   Current free space:              92K  ( 2%)
   Largest free area:               44K
   Currently SOS:                  YES
==SM: RDSA Summary
   Size:                           256K
   Current free space:              40K  (15%)
   Largest free area:               40K
   Currently SOS:                   NO
```

# Example 3 – Domain Subpool Summary

```
==SM: Domain subpool summary (SDSA)

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

   APECA     00BC              8     8          10       4      6       48      4K
   DFHAPU24  0093             16                1       0      1      512      4K
   LDPGM     0038             16                0       0      0        0      0K
   LDRES     0034             16                0       0      0        0      0K
   SMSHRU24  00C0   Y          16              102       0    102  3256656   3228K
   ZCTCTUA   0165   Y          16                0       0      0        0      0K
```

SMSHRU24 is used for many control blocks of SHARED_USER24 class storage.

# Example 3 – SMSHRU24 Control Blocks

```
SCE.SMSHRU24 13FE1098 Storage Element Descriptor

   0000  13FE1068 13FC455C 0052D980 00004310

SCE.SMSHRU24 13FE1068 Storage Element Descriptor

   0000  13FE1050 13FE1098 00522320 0000B660

SCE.SMSHRU24 13FE1050 Storage Element Descriptor

   0000  13FE1038 13FE1068 004FA2B0 00004310

SCE.SMSHRU24 13FE1038 Storage Element Descriptor

   0000  13FE1020 13FE1050 00516CC0 0000B660

SCE.SMSHRU24 13FE1020 Storage Element Descriptor

   0000  13FDFFB0 13FE1038 004F5FA0 00004310

SCE.SMSHRU24 13FDFFB0 Storage Element Descriptor

   0000  13FE1020 13FE1050 004EA940 0000B660
```

# Example 3 - SMSHRU24 Element Control Blocks notes

- Find on SCA.SMSHRU24.
- Looking at the Storage Element Descriptor (SCE) control blocks, we can see that there is a pattern.
- The 4th word in the SCE is the size of the element in hex bytes. We see the following sizes repeated: x'4310' and x'B660'

# Example 3 – Suspend Queue

```
==SM: Suspend queue summary

   KE Task   Tran #   Susptok   Subpool   DSA        Request

   1529E700 0000122 0A90000B SMSHRU24 SDSA         46688
```

- This shows us that task 122 was issuing a shared getmain for 46688 (x'B660') bytes.
  - This size matches one of the lengths seen in the SCE control blocks.

# Example 3 – PG domain

```
==PG: PTA SUMMARY FOR TRAN NUM : 00122,  PTA ADDRESS : 140FE810

LOG-LVL : 1         SYS-LVL : 0         TASK-LLE : 00000000  PLCB : 152A7060

=PG: TASK PLCB SUMMARY

PROG BELOW    LVL   1 PLCB 152A7060 LD 16D10000 ENT 96D10000 LEN 0002A8
```

- To get an idea of what programs may be involved with task 00122, we go to the PG domain by issuing:
  - verbx dfhpd660 ‘pg’
- The transaction is currently at link level 1 with program BELOW. We see that it is loaded above the line.

# Example 3 – PG domain continued

- Since we are suspect of program BELOW, we display its load point in browse mode to see its eyecatcher to see when it was last compiled:

```
|DFHYA660.0...00}.00..00.......*F|
|ILLIN*...00..BELOW   (U)07/08/11|
|  21.33 zOS660  ..}...x....'...0|
```

- Issuing IP ST SYS, the dump was taken at:

```
07/10/2011 21:36:08.862136 local
07/10/2011 20:36:08.862136 GMT
```

- This shows us that the program was recently recompiled.

# Example 3 – Summary

- In this dump, we saw a pattern of storage usage in the SDSA in subpool SMSHRU24.
- A transaction was suspended for SDSA storage for a size matching the pattern found.
- The PG domain shows the current program(s)  associated with the task.
- In this case there is only one program called BELOW. We found it was recently recompiled and believe it is doing the shared getmain for below the line storage.
  - It is important to remember that shared storage is not freed by CICS. The application will need to freemain this storage when it is no longer needed to ensure a buildup of storage usage like this does not occur.

# Storage Tuning Parameters

- Taskdataloc
- Datalocation
- Autodst
- Ruwapool
- LE Run-time Options
  - ALL31
  - HEAP
  - STACK

# Storage Tuning Parameters - notes

- Taskdataloc – Transaction definition attributes. This tells CICS where to put control blocks it obtains for the task, such as the EIB, TCA, and TWA. It defaults to below. If all the programs are 31 bit addressable, then you should specify ANY.
- Datalocation – Program definition attribute. Tells CICS where to obtain the storage on EXEC CICS commands that have a SET option. It defaults to below. Again, you should specify ANY for 31 bit programs.
- Autodst – A SIT parameter. This is Language Environment's automatic dynamic storage tuning feature in CICS. LE captures the storage usage for each program and will use this storage setting when the program is invoked next time. This is really for performance, to reduce the number of GETMAINs issued. But note, this value never decreases, which can cause adverse effects.
  - http://www-01.ibm.com/support/docview.wss?uid=swg21197278
- Ruwapool – A SIT parameter. Specifies the option for allocating a storage pool the first time a program is linked to under Language Environment.

# Storage Tuning Parameters - notes

- Use CLER to display your Language Environment Region Level Run-Time Options.
  - CLER is provided by Language Environment in group CEE.
- ALL31 – At the CICS region level we want to see ALL31 on.
- HEAP - controls the allocation of the heap size.
  - Working storage is allocated from HEAP.
  - Recommended CICS settings: ((4K,4080,ANYWHERE,KEEP,4K,4080),OVR)
- STACK - controls the allocation of the thread's stack storage.
  - Save areas are allocated from STACK.
  - Recommended CICS settings:
    - ((4K,4080,ANYWHERE,KEEP,4K,4080),OVR)

# Additional Product Resources

- IBM_CICS technical support news on Twitter
  http://www.ibm.com/support/docview.wss?uid=swg21384915
- WebSphere and CICS Support Blog
  http://www.ibm.com/developerworks/mydeveloperworks/blogs/aimsupport/
- CICS Featured documents (quarterly Support Newsletter)
  http://www.ibm.com/support/docview.wss?uid=swg27006900
- Technical support emails from My Notifications subscription
  http://www.ibm.com/software/support/einfo.html
- Webcasts for CICS products
  http://www.ibm.com/support/docview.wss?uid=swg27007244
- CICS Transaction Server support Web page
  http://www.ibm.com/support/entry/portal/Overview/Software/Other_Software/CICS_Transaction_Server

# Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe
2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com
3. **Be connected!**
   Connect with us on Facebook
   Connect with us on Twitter

# Questions and Answers