

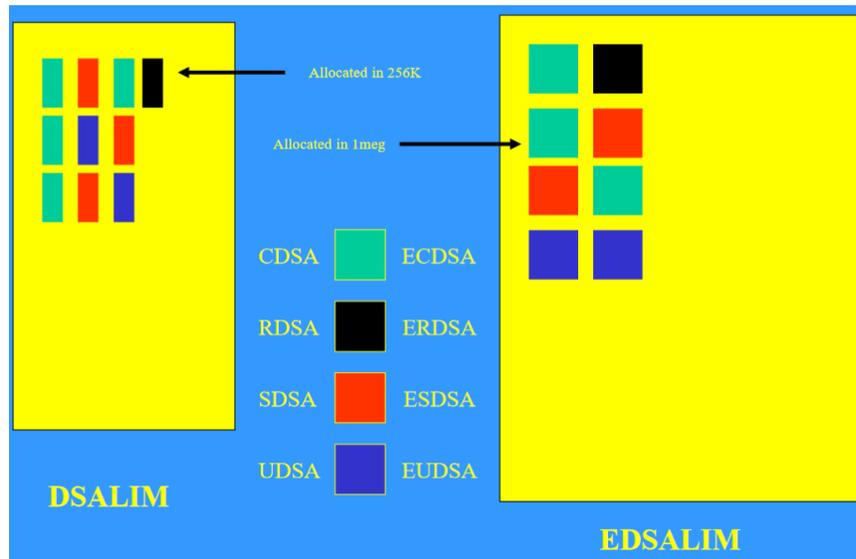


Welcome to the Virtual CICS user group newsletter. The Virtual CICS user group at [www.fundi.com/virtualcics](http://www.fundi.com/virtualcics) is an independently-operated vendor-neutral site run by and for the CICS user community.

## Virtual CICS user group presentation

The latest webinar from the Virtual CICS user group was entitled, "CICS storage violations", and was presented by Colin Pearce.

Colin has been a z/OS and CICS Systems Programmer for over 30 years. He has worked in many countries, mainly for financial institutions. He has also spent a few years working for IBM Singapore. He has written many z/OS and CICS courses and has delivered them in many countries, for many different organizations. He is well versed in both CICS Internals and CICS Transaction Debugging, having written courses in both. He is currently working for a large bank in the UK, supporting a major CICS application written in Assembler.

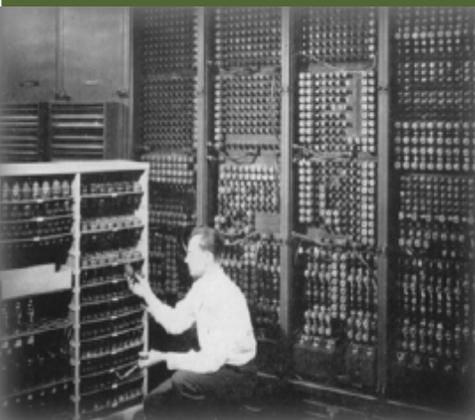


**Figure 1: DSA subpools**

Colin started the session by talking about the storage manager domain. He explained that the storage manager domain acquires storage for the Dynamic Storage Areas (DSAs) during the pre-initialization phase, based on the specifications in the System Initialization Table

### Contents:

Virtual CICS user group presentation	1
Meeting dates	6
Recent CICS articles	7
About the Virtual CICS user group	7



(SIT). As other domains and components require storage, the storage manager domain suballocates storage from the appropriate DSA and manages it as required (see Figure 1).

Four of the DSAs reside in 24-bit mode and are called: the User Dynamic Storage Area (UDSA), the CICS Dynamic Storage Area (CDSA), the Read Dynamic Storage Area (RDSA), and the Shared Dynamic Storage Area (SDSA).

The other DSAs reside in 31-bit mode and are called: the Extended User Dynamic Storage Area (EUDSA), the Extended CICS Dynamic Storage Area (ECDSA), the Extended Shared Dynamic Storage Area (ESDSA), the Extended Read-Only Dynamic Storage Area (ERDSA), the Extended Trusted Dynamic Storage Area (ETDSA), and the Grande Dynamic Storage Area (GDSA). The size of the GDSA is controlled by MEMLIMIT, and is allocated in 2GB extents. It is mainly used for Channels and Containers.

The nine DSAs (minus the GDSA) may all have the same Storage Protect key or may be different depending on whether Storage Protection is in effect. It usually is. In the absence of Storage Protection, the

storage manager domain attempts to keep user storage away from CICS storage in an attempt to reduce the exposure of storage overlays and storage violations.

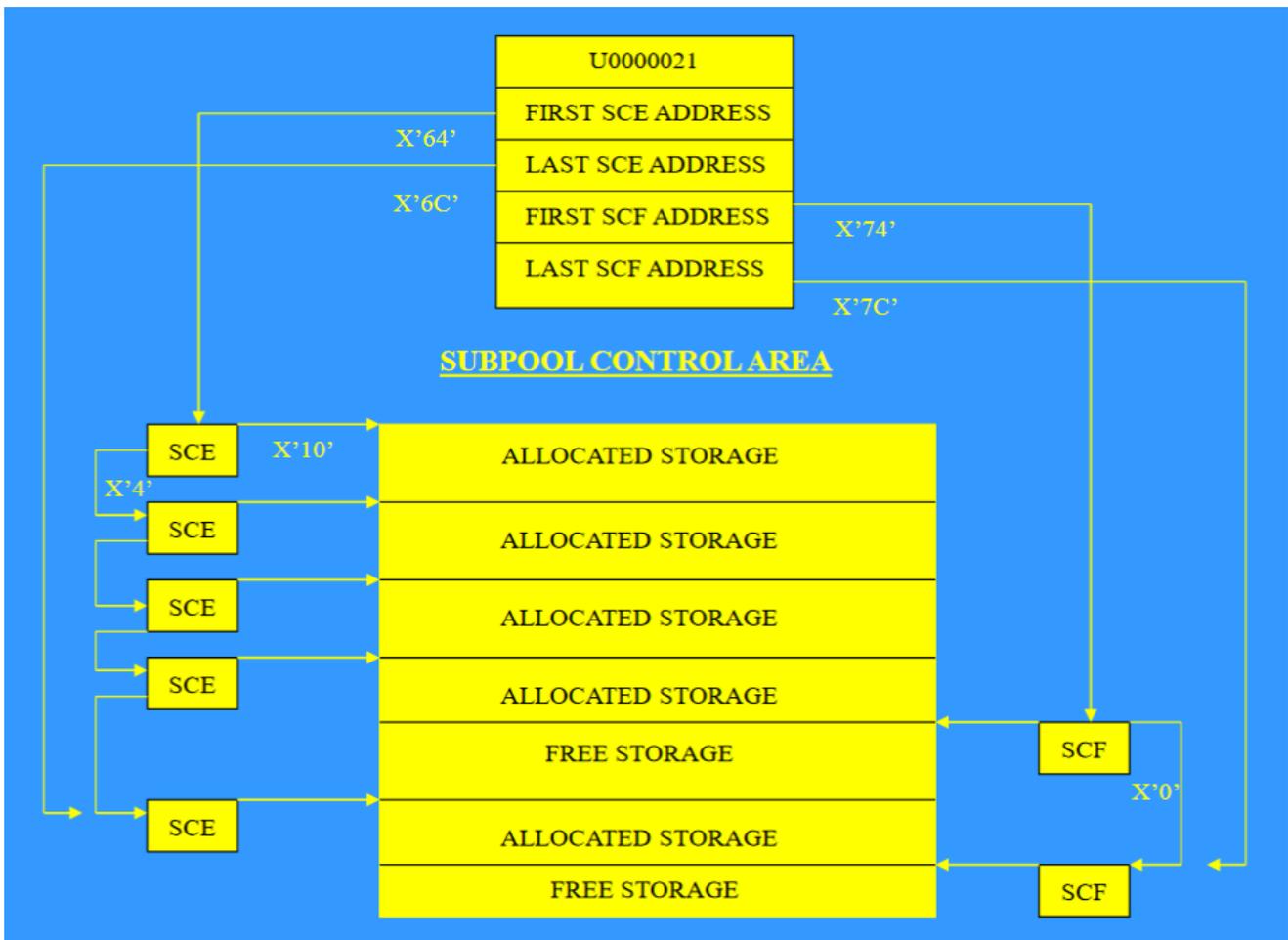
This separation is primarily aimed at programs, however, at the transactions storage level, protection is offered to transaction storage areas, if transaction isolation is defined. Storage protection must be enabled if transaction isolation is to be enabled. Command level requests that require storage are processed by the EXEC Interface Program and routed to DFHESC, which in turn converts the request into a domain call.

The CDSA (key 8) is always allocated below the 16MB line, and is referred to as CICS-KEY storage. The UDSA (key 9, providing storage protection is enabled) is allocated below the 16MB line and is referred to as USER-KEY storage. The ECDSA (key 8) is allocated above the 16MB line, and is referred to as CICS-KEY storage. The EUDSA (key 9, providing storage protection is enabled) is allocated above the 16MB line, and is referred to as USER-KEY storage. The ERDSA is allocated above the 16MB line, either from READ-ONLY (key 0) or (key 8) –

depending on the Linkedit RENT and ANY parm and the RENTPGM=PROTECT option. This option affects only the storage key, not whether the ERDSA is allocated. The 24bit RDSA is also READ-ONLY (key 0) or (key 8). The CICS Nucleus is allocated in the ERDSA. The SDSA (key 8) and ESDSA (key 8) are for user programs.

CICS executes in key 8, along with, of course, any programs that execute as a result of being invoked by a transaction. This has led to CICS code being exposed to rogue transactions because there was no storage protection offered. In order to overcome this, Supervisor key 0 could have been used, but this would have left the z/OS system exposed to CICS. So, another solution was found – the ability to have more than one storage protect key available to a single address space or region. This has required special hardware microcode to be implemented with the newer processors.

With storage protection active, CICS key 8 is known as CICS-KEY, and key 9 is known as USER-KEY. Key 9 is known as the Open or CICS wide access key. That is, any application running in any other key can modify this storage. However key 9 applications can only



**Figure 2: Storage manager control blocks**

modify key 9 storage, and can read any other key. If an application running in key 9 attempts to modify any other key, it will be abended with an S0C4 ASRA abend.

CICS invokes storage protection providing the STGPROT option has been specified in the SIT.

During the initialization process, many of the CICS components issue the ADD\_SUBPOOL Domain call to the Storage Manager to define the characteristics and names of their subpools.

With every ADD\_SUBPOOL, Storage Manager creates a control block called a SUBPOOL CONTROL AREA (SCA), which contains these subpool characteristics.

Each subpool is assigned a unique 1-byte hex identifier called the SUBPOOL ID. The storage manager returns an 8-byte token representing the subpool to the caller. This token is included in subsequent requests, enabling the storage manager to quickly locate the correct SCA,

which in turn identifies the subpool. Storage is assigned in 4k pages, therefore a subpool will occupy a 4k page, no matter how much is requested.

Storage manager creates task subpools for the exclusive use of that task. The subpools' names are 8 characters in length and begin with a one-letter id that identifies which DSA the storage is allocated is to. The remaining 7 characters identify the task number assigned to the task. In later CICS releases, the subpools

begin with just the DSA id, M,B,C, or U, followed by the taskid.

There are four task subpools that can be associated with each task or transaction. Two are below the line and two above, and they each reside in a separate DSA. They are:

Mnnnnnnn – this subpool is used to satisfy internal 24-bit CICS requests for storage in the CDSA for services issued by the task or transaction.

Bnnnnnnn – this subpool is used to satisfy application issued 24-bit EXEC CICS GETMAIN commands for storage in the UDSA.

Cnnnnnnn – this subpool is used to satisfy internal 31-bit CICS requests for storage in the ECDSA for services issued by the task or transaction.

Unnnnnnn – this subpool is used to satisfy application issued 31-bit EXEC CICS GETMAIN commands using the FLENGTH option for storage in the EUDSA.

In each case the nnnnnnn is the task number assigned to the task. In addition, each subpool will have its own Subpool Control Area (SCA).

The Task Subpool Summary is part of the storage manager domain output in IPCS. It's displayed with the VERBX 'SM=3' command. The SMX is the Transaction

Storage Area. Its role is to link the storage manager domain with the transaction manager domain. One is created for each task subpool associated with the task.

In the output, there are four transactions. The LOC and ACC headings give the location (24-bit or 31-bit), and the key 8 or 9, depending on storage protection. The number of GETMAINS and FREEMAINS in each DSA is also given.

Because storage is required in the various DSAs, CICS will allocate new 'blocks' of storage below and above the line, within the DSALIM and EDSALIM specifications. If the storage is allocated below the line, it will be allocated in 256k segments. If the storage is allocated above the line, it will be allocated in 1MB segments. This allocation needs to be monitored because the longer the CICS region is up, the more likely it is to suffer a Short-on-Storage. CEMT INQ DSAS can be issued to review the allocation of DSAs. However, if transaction isolation is enabled for the transaction(s) then the EUDSA is allocated in 1MB segments.

All of the storage manager control blocks reside outside of the DSAs in extended storage. This is to reduce the impact of storage violations.

Most of them begin with an eyecatcher or prefix. The major control blocks are anchored from the Storage Manager Anchor Block. This is the Storage Manager Anchor (SMA). Most of the frequently-used control blocks are chained as free control blocks from the SMA. If there are no free blocks, a new 4k page is acquired using an MVS GETMAIN.

The SMA anchor block, which is pointed to from the Domain Gate Header table at offset x'10', contains global storage and statistics and pointers to other storage manager control blocks. The prefix for the SMA begins with a half-word length followed by the constant: DFHSMANCHOR.

There is an SCA created for each subpool when the ADD\_SUBPOOL Domain call is issued. SCAs are chained from the SMANCHOR (the Storage Manager Anchor block). In CICS 5.2, the first SCA is at offset x'20C'. SCAs have forward and backward pointers to a chain of SCAs with the last one pointing back to the SMA. For the task subpools, the 8 bytes would take the form of ynnnnnnn, as already mentioned. Four SCAs are created with each transaction. When a task ends its Task Control Area and the associated SCAs are FREEMAINED, SCAs are either active or free.

Each SCA keeps track of the allocated/unallocated storage areas for the subpool, as well as statistical information. These 8 bytes that are at the beginning and end of every piece of allocated storage are called STORAGE CHECK ZONES (SCZS)

Allocated storage for any subpool that is defined has a Storage Element Descriptor (SCE). They are chained, and each storage allocated is called an ELEMENT. The SCEs are anchored from the SCA at offset x'64' for the subpool in use. SCEs reside outside of the DSAs (see Figure 2).

Free storage for subpools is controlled by Free Storage Descriptors (SCFs) anchored from the SCA at offset x'74' for the subpool. SCFs are also located outside of the DSAs. SCEs are x'30' and SCFs are x'20' bytes in length.

A storage violation is the writing of data into a storage area that is not owned or not properly addressed by the executing transaction. Storage violations can be prevented, detected, or undetected. Each storage area allocated to the transaction will have an 8-byte SCZ at the beginning of the storage and a duplicate 8-byte SCZ at the end. CICS TS allows for separation between CICS and user applications. The

use of the ERDSA can also offer a measure of write protection. The ability to detect a storage violation has been improved with each successive release of CICS with the introduction of SCEs and SCFs to monitor storage and SCZs as well as the existing SAAs (Storage Accounting Areas). Undetected storage violations can result in bad data being written to files and terminals. With SAAs and SCZs, it is important to remember that the violation will only be discovered when the storage areas are checked. This usually occurs at the end of the transaction.

Transaction isolation allows for the separation of user storage, so that each transaction has only write access to its own Task Lifetime Storage, and is not allowed access to any other task's storage. Of course, write access must be allowed to shared storage such as Temporary Storage areas and TIOAs. z/OS offers subspace grouping, which allows for the logical partitioning of an address space. Each address space is also a base space. Each base space can be divided into many subspaces. Each subspace takes 2.5 pages per transaction. Each page is 1MB. Transaction isolation is very expensive in terms of real storage, especially with a high Max task setting

(MXT \* 2.5 pages). Storage protection (STGPROT in the SIT) must also be enabled.

CICS will issue messages to the CICS system log and invoke an SDUMP to the z/OS dynamic dump dataset. One message you may see is:

```
DFHSM0102 applid A
storage violation (code
x'cccc') has been detected
in module name
```

where cccc is the exception trace point id. This can be interrogated by using CICS/TS Diagnosis Handbook – LX33-6093-00

Message DFHME0116 is also issued to offer more information on a system wide basis, such as the latest maintenance, that has been applied.

If a protection exception is encountered attempting to access a protected DSA, message DFHSR0622 will be sent to the console indicating which DSA the application attempted to access.

- If the DSA is either CDSA or ECDSA, Storage Protection is active.
- If the DSA is RDSA or ERDSA, RENTPGM=PROTECT has been specified, regardless of whether storage protection has been enabled.

- If the DSA is UDSA or EUDSA, both storage protection and transaction isolation have been enabled.

The transaction will be abended with an ASRA and message DFHAP0001/DFHSR0001, which will also be sent to the console. A system dump will be taken unless suppressed by the master terminal operator.

Storage violations may be detected when the overlaid storage is explicitly FREEMAINED or implicitly FREEMAINED at the end of task. When the violation is detected by an explicit FREEMAIN, you'll see an exception trace record, usually a code of 030B. This will be part of message DFHSM0102 sent to the console. If the SAA or SCZ is not repaired, the transaction will be abended with an AEXZ.

When the violation is detected at the end of task, the code 0F0C exception trace is written to the console with message DFHSM0102. If STGRCVY=YES in the SIT has been requested, repair of the SAA or SCZ will be attempted. A storage violation dump will be taken to the z/OS Dynamic Dump dataset. When analysing a storage violation, the following questions need to be addressed:

- When was the storage violation detected?
- Is it possible to ascertain when the storage was last known to be good?
- What tasks were in control between these two points?
- Any task that fits into the above criteria is suspect.

Be suspicious of the following:

- COMMAREAs that don't match between linking programs
- Overruns of table arrays – a mismatch of copybooks
- Absolute values specified with a CICS GETMAIN command

Storage violations can be harder to analyse if the violation was detected at the end of the transaction.

In CICS/TS 5.2, the SCE is x'30' and SCF are x'20' in length. Offset x'10' in the SCE contains the address of the storage area and offset x'14' contains the length of this storage area inclusive of the SCZs. In CICS/TS 4.1 the SCE is x'18' and SCF are x'18' in length. The address of the storage area is at offset x'8' and the length is at offset x'C'

In a storage violation situation, the SCE becomes extremely important because

it not only contains the address of storage (which should match up with the \*EXC entry in the trace), but also the length of the storage that has been found to be corrupted.

Storage Manager will write an Exception Trace Entry into the trace table. The Abbrev Trace Table entry will contain just the exception entry – it is the Full Trace Table entry that contains all the details of the trace, which should match up with the 030B or 0F0C in the CICS Trace Table manual.

Colin Pearce spent some time explaining how to go through a trace and find the information needed to identify the problem..

A copy of Colin Pearce's presentation is available for download from the Virtual CICS user group Web site at [www.fundi.com/virtualcics/presentations/CICSStorageViolationSep17.pdf](http://www.fundi.com/virtualcics/presentations/CICSStorageViolationSep17.pdf).

You can see and hear the whole user group meeting by downloading the WMV file from [www.fundi.com/virtualcics/presentations/2017-09-12meeting.wmv](http://www.fundi.com/virtualcics/presentations/2017-09-12meeting.wmv).

## Meeting dates

The following meeting dates have been arranged for the Virtual CICS user group:

- 14 November 2017, when Mary Abdill will be discussing, “Back to basics”.
- The following meeting is on 16 January 2018, when Ezriel Gross will be presenting.

We will be using Citrix GoToMeeting for the user group meetings.

## Recent CICS articles

*Introduction to CICS – new IBM Redbooks video course now available!* by D Harris on CICS DevCenter (22 September 2017). You can find the article at: <https://developer.ibm.com/cics/2017/09/22/introduction-cics-new-ibm-redbooks-video-course-now-available/>.

*IBM Redbooks ‘Introduction to CICS’ Video Course* by Ashleigh Denholm on CICS DevCenter (22 September 2017). You can find the article at: <https://developer.ibm.com/cics/2017/09/22/ibm-redbooks-introduction-cics-video-course/>.

*Developing an MQ JMS application for CICS Liberty* by Phil Wakelin on CICS DevCenter (6 September 2017). You can find the article at: <https://developer.ibm.com/cics/2017/09/06/developing-jms-application-cics-liberty/>.

*Monitoring and Administration Needed for CICS Effectiveness* by Jim Schesvold in *IBM Systems Magazine* (September 2017). You can find the article at: <http://ibmsystemsmag.com/mainframe/administrator/cics/monitor-administration/>.

*EJBs in Liberty JVM server - a Web Shop example* by Alexander D Brown on CICS DevCenter (21 August 2017). You can find the article at: <https://developer.ibm.com/cics/2017/08/21/cics-ejb/>.

*New event processing capabilities with CICS TS V5.4* by Catherine Moxey on CICS DevCenter (1 August 2017). You can find the article at: <https://developer.ibm.com/cics/2017/08/01/cics-ep-new/>.

*Developing a JMS application for CICS Liberty* by Phil Wakelin on CICS DevCenter (28 July 2017). You can find the article at: <https://developer.ibm.com/cics/2017/09/06/developing-jms-application-cics-liberty/>.

*CICS Tools Identify Performance Issues* by Jim Schesvold in *IBM Systems Magazine* (August 2017). You can find the article at: <http://ibmsystemsmag.com/mainframe/administrator/cics/performance-tools/>.

## About the Virtual CICS user group

The Virtual CICS user group was established as a way for individuals using IBM’s CICS TS systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at [www.fundi.com/virtualcics](http://www.fundi.com/virtualcics) provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM CICS practitioners. Anyone with an interest in CICS is welcome to join the Virtual CICS user group and share in the knowledge exchange.

To share ideas, and for further information, contact [trevor@itech-ed.com](mailto:trevor@itech-ed.com).

The Virtual CICS user group is free to its members.