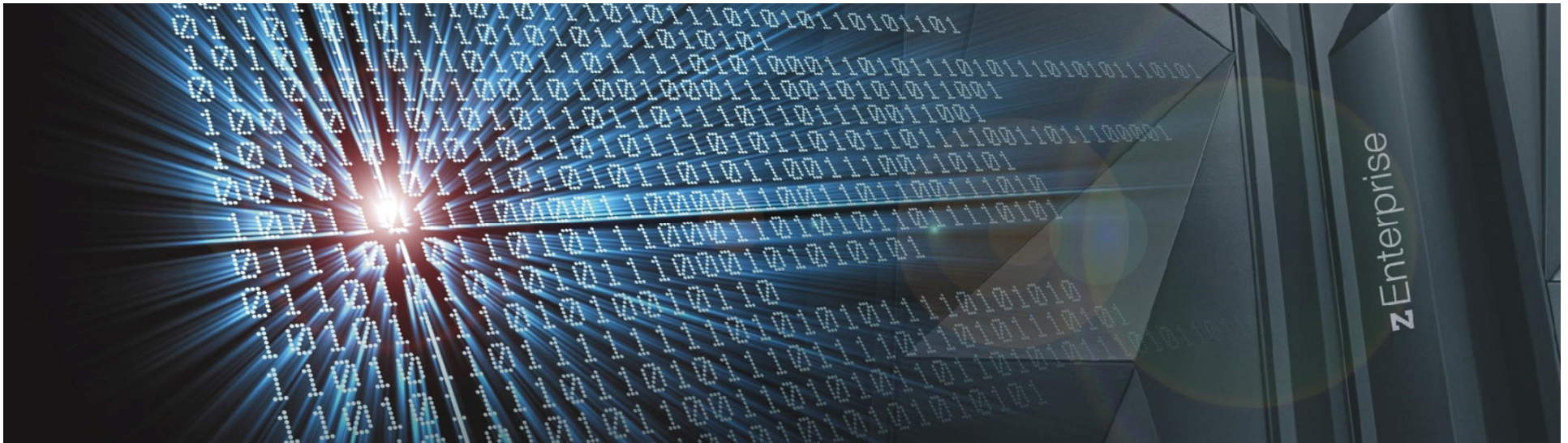


z/VSE VSAM Enhancements

Mikhail Zaslonko



<http://www.ibm.com/zVSE>

<http://twitter.com/IBMzVSE>

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business (logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT").

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

- ✓ IDCAMS Commands Security
- ✓ DLBL CISIZE parameter for SAM-ESDS Implicit Definition
- ✓ Catalog Management Trace
- ✓ Additional Functional Enhancements
- ✓ SHOWCB Enhancements

IDCAMS Commands Security

IDCAMS tool provides a number of cluster management and catalog maintenance commands which can be destructive to data. To prevent cases of data destruction, system administrators can restrict the usage of IDCAMS commands with the help of a security manager, for example the Basic Security Manager (BSM) provided with z/VSE.

The administrator can control access to IDCAMS commands by using the 'IDCAMS.GENERAL' BSM resource profile of the resource class FACILITY.

IDCAMS commands access control is designed for batch processing only. Thus if batch security is not active or IDCAMS function is executed in ICCF pseudo partition, then no RACROUTE calls performed and all IDCAMS commands are executed as requested. Same happens when batch security is active but the profile 'IDCAMS.GENERAL' was not defined to the BSM.

It is up to the administrator to define and setup 'IDCAMS.GENERAL' resource profile of the resource class FACILITY. It can be done either with BSTADMIN utility in batch or via Interactive Interface dialogs (fast-path 28).

IDCAMS Security – READ authorization level

If batch security is active and the corresponding IDCAMS.GENERAL profile is defined, then an ID statement has to be supplied in the job to authenticate a user.

```
// ID USER=OPER, PWD=QWERTY
```

IDCAMS commands were split into three groups. Each group corresponds to the specific authorization level to IDCAMS.GENERAL profile – **Read**, **Update** or **Alter**

Users having **Read** authorization level are permitted to perform the following set of IDCAMS commands:

- LISTCAT - lists entries contained in a catalog
- PRINT - lists a part or the whole VSAM file
- BACKUP - produces a backup copy of one or more VSAM objects

IDCAMS Security – UPDATE authorization level

Users having **Update** authorization level are permitted to perform commands for the **Read** authorization level plus the set of IDCAMS commands listed below.

- DEFINE CLUSTER|AIX|PATH|NONVSAM - defines cluster, alternate index or path
- DELETE CLUSTER|AIX|PATH|NONVSAM - deletes cluster, alternate index or path
- EXPORT/IMPORT - exports/imports cluster or alternate index
- REPRO - copies data from one dataset to another
- RESTORE - defines cluster (if required) and fills it with the data from the backup medium
- BLDINDEX - builds one or more alternate indexes
- VERIFY - verifies and corrects (if required) end-of-file information

Note:

1. The scope of using of the DEFINE and DELETE commands is limited to cluster, alternate index, path and non-VSAM object.
2. EXPORT CONNECT and IMPORT DISCONNECT are not allowed for this authorization level.

IDCAMS Security – ALTER authorization level

Users having **Alter** authorization level are permitted to perform commands for the **Read** and **Update** authorization level plus the set of IDCAMS commands listed below.

- DEFINE MASTERCATALOG|USERCATALOG|SPACE - defines master catalog, user catalog, or space
- DELETE MASTERCATALOG|USERCATALOG|SPACE - deletes master catalog, user catalog, or space
- IMPORT DISCONNECT - disconnects user catalog from master catalog
- EXPORT CONNECT - connects user catalog to master catalog
- ALTER - changes attributes of catalog entries

IDCAMS Security – Messages and codes

If the user's authorization level to IDCAMS.GENERAL profile is enough to execute a specific IDCAMS function, then the command is executed without any extra messages.

If the user's authorization level to IDCAMS.GENERAL profile is **not** enough to execute a specific IDCAMS function, IDCAMS function is interrupted and the following messages are displayed in SYSLST:

```
IDC32240I RACROUTE (AUTH) FAILED WITH RETURN CODE 8 REASON 0
IDC32241I SAF RETURN CODE 8 FOR RACROUTE (AUTH)
```

At the same time message BST120I from BSM is displayed on the console. It shows which BSM resource class and resource profile are affected.

```
BG 0000 BST120I USER(OPER )
      BST120I IDCAMS.GENERAL CL(FACILITY)
      BST120I INSUFFICIENT ACCESS AUTHORITY
      BST120I FROM IDCAMS.GENERAL
      BST120I ACCESS INTENT(UPDATE ) ACCESS ALLOWED(READ )
```

Job isn't cancelled, IDCAMS processing continues with the next command specified.

IDCAMS Security – IDCAMS.GENERAL Profile Setup in Batch

The JCL sample below shows how to use BSTADMIN utility for defining the IDCAMS.GENERAL resource profile in BSM. This profile setup allows everyone to use the 'read-only' commands and grants user USR1 **update** authorization level and user USR2 **alter** authorization level to the IDCAMS.GENERAL profile.

```
// EXEC BSTADMIN
  ADD FACILITY IDCAMS.GENERAL UAC(READ)
  PERMIT FACILITY IDCAMS.GENERAL ID(USR1) ACCESS(UPD)
  PERMIT FACILITY IDCAMS.GENERAL ID(USR2) ACCESS(ALT)
  PERFORM DATASPACE REFRESH
  LIST FACILITY IDCAMS.GENERAL
/*
```

BSTADMIN LIST command output in SYSLST:

```
FACILITY    IDCAMS.GENERAL
UNIVERSAL ACCESS
-----
          READ

INSTALLATION DATA
-----
          NONE

AUDITING
-----
FAILURES(READ)

USER        ACCESS
-----
USR1        UPDATE
USR2        ALTER
```

IDCAMS Security – IDCAMS.GENERAL Profile Setup in UI

Adding new IDCAMS.GENERAL resource profile of the class FACILITY (fastpath 2819)

```

IESADMBSLE                MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS: FACILITY      (START is Case Sensitive)      STATUS: ACTIVE
START....
OPTIONS:  1 = ADD              2 = CHANGE              5 = DELETE              6 = ACCESS LIST

  OPT  PROFILE NAME                DESCRIPTION                UNIVERSAL AUDIT
      >                                ACCESS VALUE
  =    DFHRCF.BRSLPU                >                                12
  -    DFHRCF.BRSL00                >                                12
  -    DFHRCF.BRSL01                >                                12
  -    DFHRCF.BRSL02                >                                12

```

```

IESADMBSAE                MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:      FACILITY

Add Profile:

  PREFIX..... _____      CICS region
  RESOURCE NAME.....          Maximum length is 39 characters.
  ..... IDCAMS.GENERAL
  GENERIC..... 1              (1=yes, 2=no)
  UNIVERSAL ACCESS... 2       (_=None, 2=Read, 3=Update, 4=Alter)
  AUDIT-LEVEL 1 .....        (_=None, 1=Failure, 2=Success, 3=All)
  ACCESS-LEVEL 1 .....        (2=Read, 3=Update, 4=Alter, _=default)
  AUDIT-LEVEL 2 .....        (_=None, 1=Failure, 2=Success, 3=All)
  ACCESS-LEVEL 2 .....        (2=Read, 3=Update, 4=Alter, _=default)
  DESCRIPTION.....          Optional remark
PF1=HELP                    3=END                    5=UPDATE

RESOURCE NAME FIELD IS CASE SENSITIVE. ENTER DATA AS REQUIRED.

```

IDCAMS Security – IDCAMS.GENERAL Profile Setup in UI

Configuring IDCAMS.GENERAL resource profile access list (fastpath 2819)

```

IESADMBSLA          MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS: FACILITY      (START is Case Sensitive)      STATUS: ACTIVE
START.... DFHRCF.RSL24
OPTIONS:   1 = ADD                2 = CHANGE                5 = DELETE                6 = ACCESS LIST

  OPT   PROFILE NAME                DESCRIPTION                UNIVERSAL AUDIT
                                     >                          ACCESS VALUE
  --   DFHRCF.RSL24                >                          12
  --   IBMVSE.JCL.ASSGN.PERM        >                          12
  --   IBMVSE.JCL.LIBDEF.PERM      >                          12
  --   IBMVSE.JCL.LIBDROP.PERM     >                          12
  --   IBMVSE.JCL.OPTION.PARSTD    >                          12
  --   IBMVSE.JCL.OPTION.STDLABEL  >                          12
  6   *IDCAMS.GENERAL                >                          2          12

```

```

IESADMBSLA          MAINTAIN ACCESS LIST
BSM CLASS: FACILITY
PROFILE: *IDCAMS.GENERAL
START....
OPTIONS:   1 = ADD                2 = CHANGE                5 = DELETE
NUMBER OF ENTRIES ON LIST: 00000

  OPT   NAME      ACC
  1

```

```

IESADMBSAA          MAINTAIN ACCESS LIST
BSM CLASS: FACILITY
PROFILE: *IDCAMS.GENERAL

Add Userid or Groupid:

  NAME..... USR1                Userid or Groupid
  ACCESS..... 3                (_=None,
                                2=Read, 3=Update, 4=Alter)

```

IDCAMS Security – IDCAMS.GENERAL Profile Setup in UI

Rebuilding BSM Security Information (fastpath 283)

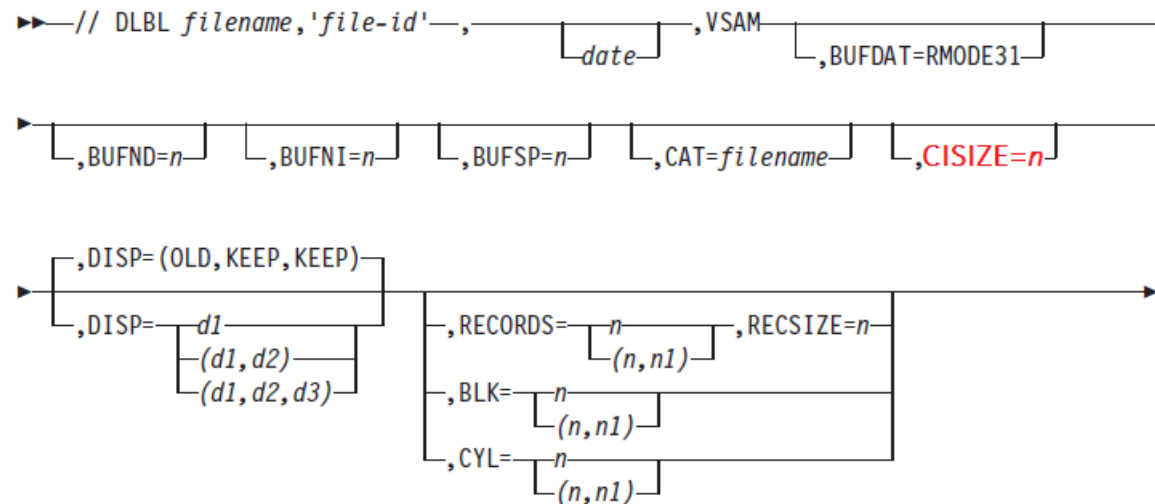
```
IESADMSL.IESEBSEC                SECURITY MAINTENANCE                APPLID: DBDCCICS
Enter the number of your selection and press the ENTER key:

  1  BSM Resource Profile Maintenance
  2  BSM Group Maintenance
  3  BSM Security Rebuild
  4  Maintain Certificate - User ID List
  5  Define Transaction Security (DTSECTXN)
  6  BSM Cross Reference Report
  7  Unified BSM Resource Profile Maintenance

PF1=HELP                3=END                4=RETURN                6=ESCAPE (U)
                        9=Escape (m)
SECURITY INFORMATION WAS SUCCESSFULLY REBUILT.
==> 3_                Path: 28
```

DLBL CISIZE parameter for SAM-ESDS Implicit Definition

Existing DLBL CISIZE parameter now allowed not only for SD files but also for VSAM files. This is to enable customer to specify for SAM-ESDS implicit define via DLBL, a CISIZE other than the default assigned by VSAM.



CISIZE=n

For VSE/VSAM this parameter specifies a control interval size for SAM-ESDS dataset. The size overrides that specified (or defaulted) in the respective DTF macro. The specified size must be a number from 1 to 32,768. VSAM will round the value up to the multiple of 512 bytes or multiple of 2K (if specified value is greater than 8K) but greater than the SAM logical block length.

Catalog Management Trace

New Catalog Management Trace was developed to support the investigation and resolution of Catalog Management problems by dumping the partition in trouble as close as possible to the point where the error occurred.

Most of the Catalog Management modules are setting at exit their return and reason codes accompanied with two characters module IDs. This diagnostic information is included in syslist or syslog error messages, e.g. :

```
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLBN-6
```

or

```
4228I FILE DFHTEMP      OPEN  ERROR X'B4'(180) CAT=VSESPUC ( 4,AH, 10)
```

Those return codes are processed only after coming back from catalog management. Even if an SDUMP is issued, all Catalog Management data structures are partly destroyed or freed by that time.

We have enhanced existent Catalog Management SNAP TRACE 001 in IKQVEDA tool. The new DUMP parameter was added, which specifies a return_code, module_id, and reason_code combination to issue SDUMP as soon as the specified module completed with specified return and reason codes.

Catalog Management Trace - Invocation

IKQVEDA SNAP 001 trace command format is shown below:

```
ENABLE SNAP=001, PART=partition, DUMP=(return_code, module_id, reason_code)
```

PART=*partition* specifies partition in which the specified SNAP001 trace is enabled. If SNAP001 was already enabled for that partition, new parameters are overlapping the previous values.

DUMP=(*return_code*, *module_id*, *reason_code*) specifies the *return_code*, *module_id*, and *reason_code* combination which is to cause SDUMP. This parameter can be specified only with the SNAP=001 trace request. Application continues execution after the SDUMP is completed. SDUMPs are not issued if OPTION NODUMP is in effect.

return_code is any decimal number from 0 to 255.

module_id is a two characters suffix of a Catalog Management module name.

reason_code is a decimal number from 0 to 255.

A sample of the IKQVEDA SNAP command is shown below:

```
// EXEC IKQVEDA, PARM='SYSIPT'  
      ENABLE SNAP=001, PART=F8, DUMP=(4, AH, 10)
```

Unless otherwise stated, SDUMP will be stored in the SYSDUMP library.

To avoid a waterfall of dumps, an SDUMP is issued only for the first occurrence of the matched codes combination per job step.

Additional Functional Enhancements

Deletion of the KSDS cluster with ERASE attribute after unsuccessful RESTORE

When KSDS cluster cannot be extended on the RESTORE, then there might be an error during follow on deletion attempt if that cluster has been defined with ERASE attribute.

```
IDC01304I SUCCESSFUL DEFINITION OF TEST.CLUSTER
IDC31338I CANNOT EXTEND TEST.CLUSTER
IDC31334I CANNOT DELETE OLD VERSION OR ASSOCIATION OF TEST.CLUSTER
IDC31316I ** VSAM CATALOG RETURN CODE IS 250 - REASON CODE IS IGG0CLGB-52
```

DEFINE SPACE CANDIDATE on FBA/SCSI disks.

An attempt to define data space with CANDIDATE option on FBA/SCSI device ended up with the following error:

```
IDC0511I SPACE ALLOCATION STATUS FOR VOLUME SCSI00 IS 68
IDC3020I INSUFFICIENT SPACE ON USER VOLUME
```

Remove duplicate VOLSERs during DEFINE CLUSTER

IDCAMS permits definition of a cluster with duplicate Volser's:

```
VOLUMES (SYSWK2, SYSWK2, SYSWK3, SYSWK3, SYSWK3)
```

However that could lead to the following error if this volume ever needs to be removed (via ALTER REMOVEVOLUME) :

```
IDC3012I ENTRY TEST.KSDS1.DATA NOT FOUND
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLBN-6
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12
```

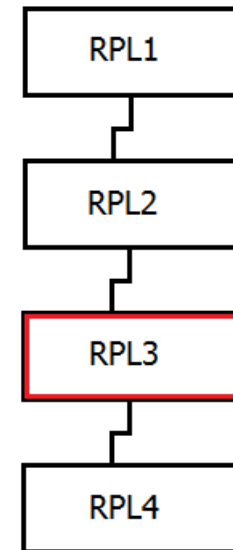
Now IDCAMS eliminates duplicate Volser's for DEFINE CLUSTER and DEFINE ALTERNATEINDEX, but does not bring a non-zero return code. This is to ensure existing user JCL, which currently includes duplicate Volser's, will not fail.

Enhancements in Chained RPL processing

Sometimes VSE/VSAM applications can save system overhead (minimize number of IO operations) by “chaining” VSAM requests.

All Request Parameter Lists (RPLs) in the chain are seized during initialization in Record Management. If an RPL in the middle of the chain experiences an error, then all subsequent requests won't be carried out. However, rest of the RPLs on the chain might not be released, causing problems with some applications.

Now the entire RPL chain will be released if an error is detected in one RPL in the chain.



SHOWCB Enhancements

The SHOWCB macro displays fields from an ACB, EXLST, or RPL control blocks. SHOWCB places these fields in an area that you provide. This is essential to use an official interfaces like SHOWCB to obtain values from VSAM internal control blocks. Overtime displacements and format of VSAM internal control blocks could be changed, which will affect all applications/products which obtain information from control blocks directly

Starting with z/VSE V5.1, the SHOWCB macro also provides the following information:

- LSR (Local Shared Resources) matrix which contains string statistics information, information about each buffer sub-pool defined for the specified LSR pool, and LSR string and buffer statistics for each cluster within a specified share pool.
- Extent matrix which contains characteristics of physical devices on which the specified cluster resides and information about all extents for the specified cluster.
- Nine new keywords for the FIELDS parameter.

All attributes supported by the SHOWCB macro are described in the “*SHOWCB Macro Section*” of Chapter 12 “*Descriptions of VSE/VSAM Macros*” in the “*VSE/VSAM User’s Guide and Application Programming*”.

New FIELDS supported by SHOWCB Macro

Nine new keywords for the FIELDS parameter are supported by SHOWCB in order to enable user to obtain status information for open VSAM datasets.

SHOWCB FIELD	Length	Description
IDACB	4	The ACB identifier is equal to x'A0'.
IDDOS	4	The DOS identifier is equal to x'28'.
CDBUF	4	Number of data buffers
CIBUF	4	Number of index buffers
CNAME	44	Name of the cluster
CIPCA	4	Number of control intervals per control area
LNEST	4	Local number of index levels
BFREE	4	Number of unassigned buffers
OPENOBJ	4	AMS flag byte. With the AMS flag you can determine whether the opened object is a path, a base cluster, or an alternate index: x'80'=alternate index x'40'=access via path x'20'=access via base cluster

Example of a SHOWCB Call

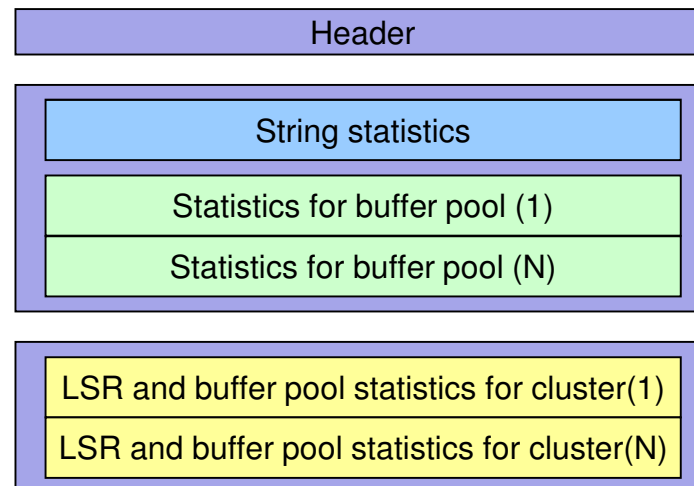
```
SHOWCB ACB=ACB1,AREA=AREA1,LENGTH=100,FIELDS=(IDACB,IDDOS, X
CDBUF,CIBUF,CIPCA,LNEST,BFREE,OPENOBJ,CNAME)
```

LSR Matrix

Originally SHOWCB returns 1 value for each specified keyword in FIELDS parameter. Starting 5.1 SHOWCB also support 2 new keywords, which allow user to obtain a set of values (matrix). The size of this matrix depends on your system definitions and will return you all information about the specified Local Share Pool.

Returned LSR matrix consists of three parts:

- Header.
- Share Pool Statistics Area: string statistics area which contains total number of LSR strings and buffer matrix which contains buffer statistics for the requested share pool.
- Cluster Matrix Area: contains LSR string and buffer statistics for each VSAM cluster assigned to a specified share pool.



LSR Matrix (Header layout)

Information in the header shows the 'map' of the retrieved data. It should be used to parse the returned values and to provide large enough area to obtain all information.

Header has a fixed size of 32 bytes and contains the following fields:

Field	Length
Length of area supplied by user	4 bytes
Total length used (or required) by VSAM	4 bytes
Length of string statistics area	4 bytes
Number of rows in buffer matrix	4 bytes
Length of rows in buffer matrix	2 bytes
Number of rows in cluster matrix	4 bytes
Length of rows in cluster matrix	2 bytes
(reserved)	4 bytes
(reserved)	4 bytes

LSR Matrix (Header layout)

Header fields description:

- **Length of area supplied by user** contains the length of the area passed by user in the LENGTH parameter of the MACRO call in bytes.
- **Total length used (or required) by VSAM** contains the length of the area actually used by or needed for VSAM to display string statistics, buffer matrix, and cluster matrix, including length of the header, in bytes.
- **Length of string statistics area** contains the length of fixed string statistics area (first part of share pool statistics area) in bytes.
- **Number of rows in buffer matrix** contains the number of fixed size rows that are displayed in the user's area. This number also indicates the number of sub-pools in a specified share pool.
- **Len of rows in buffer matrix** contains the length of each row in buffer matrix in bytes.
- **Number of rows in cluster matrix** contains the number of fixed size rows that are passed to the user. This number also indicates the number of clusters in a specified share pool, including base clusters opened via a path.
- **Length of rows in cluster matrix** contains the length of each row in cluster matrix in bytes.

LSR Matrix (Share Pool Statistics Area)

String Statistics Area

This area contains the following information:

Field	Length
Share pool number	2 bytes
Total number of strings	2 bytes
Number of active strings	2 bytes
Number of free strings	2 bytes

Buffer Matrix Area

Following string statistics area is a series of thirty two byte rows that contain descriptors for each buffer pool defined for this share pool. The fields in a row are the following:

Field	Length
Size of buffer	2 bytes
Type of buffer	1 byte
Flags	1 byte
Number of buffers	4 bytes
Number of modified buffers	4 bytes
Number of free buffers	4 bytes
Number of buffer reads	4 bytes
Number of retry requests without I/O	4 bytes
Number of user-initiated writes	4 bytes
Number of non-user-initiated writes	4 bytes

LSR Matrix (Cluster Matrix Area)

Cluster Matrix Area contains LSR string and buffer statistics for each cluster within a specified share pool.

This part contains fixed size rows, number of which equals the number of clusters associated with a specified share pool. The length of a row and the current number of rows are contained in the header.

Cluster Matrix Area

Field	Length
DDNAME	8 bytes
Type of cluster	1 byte
Flags	1 byte
Number of active strings for this cluster	2 bytes
Size of data buffers	4 bytes
Number of data buffers used	4 bytes
Size of index buffers	4 bytes
Number of index buffers used	4 bytes
(reserved)	4 bytes
(reserved)	4 bytes

Example of an LSR Matrix Call

SHOWCB AREA=USER_AREA, LENGTH=800, **SHAREPL=1**, FIELDS=(**LSRINF**)

Example of LSR Matrix output (part 1)

This part of the output shows the Header and Share Pool Statistics Area.

```

00000320 00000190 00000010 00000002 00200000 00080024 10 .....
|          |          |          |          |          |          |          ^=====LENGTH of cluster row=x'24'=36
|          |          |          |          |          |          |          ^=====NUM of cluster rows=8
|          |          |          |          |          |          |          ^=====len of buffer row=x'20'=32
|          |          |          |          |          |          |          ^=====NUM OF buffer ROWS=2
|          |          |          |          |          |          |          ^=====LEN OF FIXED AREA=x'10'=16
|          |          |          |          |          |          |          ^=====AREA NEEDED FOR VSAM=x'190'=400
|          |          |          |          |          |          |          ^=====AREA SUPPLIED BY USER=x'320'=800
^=====D.....
00000000 00000000 00010014 00030011 00000000 00000000 0200c400 00000004 .....
|          |          |          |          |          |          |          |          ^=====NUMBER OF BUFFERS=4
|          |          |          |          |          |          |          |          ^=====FLAGS=RESERVED=0
|          |          |          |          |          |          |          |          ^=====TYPE OF BUF='D'
|          |          |          |          |          |          |          |          ^=====SIZE OF BUFFERS=x'200'=512
|          |          |          |          |          |          |          |          ^=====RESERVED ( 6 bytes)=0
|          |          |          |          |          |          |          |          ^=====RESERVED
|          |          |          |          |          |          |          |          ^=====NUMBER OF FREE STRINGS=x'11'=17
|          |          |          |          |          |          |          |          ^=====N OF ACTIVE STRINGS=3
|          |          |          |          |          |          |          |          ^=====TOTAL NUMBER OF STR=x'14'=20
|          |          |          |          |          |          |          |          ^=====SHR POOL NUMBER=1
|          |          |          |          |          |          |          |          ^=====RESERVED2=0
|          |          |          |          |          |          |          |          ^=====RESERVED1=0
^=====D.....
00000000 00000004 00000000 00000000 00000000 00000000 2000c400 00000014 .....
|          |          |          |          |          |          |          |          |          ^=====NUMBER OF BUFFERS=x'14'=20
|          |          |          |          |          |          |          |          |          ^=====FLAGS=RESERVED=0
|          |          |          |          |          |          |          |          |          ^=====TYPE OF BUF='D'
|          |          |          |          |          |          |          |          |          ^=====SIZE OF BUFFERS=x'2000'=8192
|          |          |          |          |          |          |          |          |          ^=====NUM OF NON-USER-INIT WR=0
|          |          |          |          |          |          |          |          |          ^=====NUM OF USER-INIT WRITES=0
|          |          |          |          |          |          |          |          |          ^=====NUM OF RETR-REQ WITHOUT IO=0
|          |          |          |          |          |          |          |          |          ^=====NUM OF BUFFER-READS=0
|          |          |          |          |          |          |          |          |          ^=====NUM OF FREE BUFFERS=4
|          |          |          |          |          |          |          |          |          ^=====NUM OF MOD BUF=0
^=====

```

Example of LSR Matrix output (part 2)

This part of the output shows Cluster Matrix Area.

```

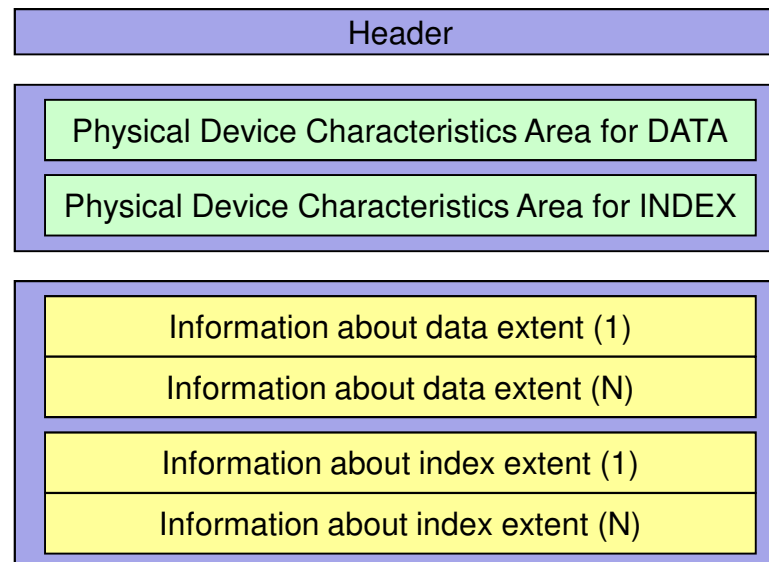
00000000 00000008 0000000B 00000015 00000009 00000000 D2E2C4E2 40404040 .....KSDS
|         |         |         |         |         |         ^=====DDNAME=KSDS
|         |         |         |         |         |         ^=====NUM OF NON-USER-INIT WR=0
|         |         |         |         |         |         ^=====NUM OF USER-INIT WRITES=9
|         |         |         |         |         |         ^=====NUM OF RETR-REQ WITHOUT IO=x'15'=21
|         |         |         |         |         |         ^=====NUM OF BUFFER-READS=x'B'
|         |         |         |         |         |         ^=====NUM OF FREE BUFFERS=8
|         |         |         |         |         |         ^=====NUM OF MODIFIED BUFFERS=0
^=====
C2000001 00002000 00000007 00002000 00000007 00000000 00000000 D2E2C4E2 B.....KSDS
|         |         |         |         |         |         |         ^=====DDNAME=KSDS3
|         |         |         |         |         |         |         ^=====RESERVED2=0
|         |         |         |         |         |         |         ^=====RESERVED1=0
|         |         |         |         |         |         |         ^=====NUM OF INDEX BUFFERS=7
|         |         |         |         |         |         |         ^=====SIZE OF INDEX BUFFERS=x'2000'=8192
|         |         |         |         |         |         |         ^=====NUM OF DATA BUFFERS=7
|         |         |         |         |         |         |         ^=====SIZE OF DATA BUFFERS=x'2000'=8192
|         |         |         |         |         |         |         ^=====NUM OF ACTIVE STRINGS=1
|         |         |         |         |         |         |         ^=====TYPE OF CLUSTER=B
^=====
F3404040 C2000001 00002000 00000005 00002000 00000005 00000000 00000000 3 B.....
|         |         |         |         |         |         |         ^=====RESERVED2=0
|         |         |         |         |         |         |         ^=====RESERVED1=0
|         |         |         |         |         |         |         ^=====NUM OF INDEX BUFFERS=5
|         |         |         |         |         |         |         ^=====SIZE OF INDEX BUFFERS=x'2000'=8192
|         |         |         |         |         |         |         ^=====NUM OF DATA BUFFERS=5
|         |         |         |         |         |         |         ^=====SIZE OF DATA BUFFERS=x'2000'=8192
|         |         |         |         |         |         |         ^=====NUM OF ACTIVE STRINGS=1
|         |         |         |         |         |         |         ^=====TYPE OF CLUSTER=B
^=====
D2E2C4E2 F2404040 C2000001 00002000 00000003 00002000 00000003 00000000 KSDS2 B.....
00000000 D2E2C4E2 F4404040 C2000000 00002000 00000000 00002000 00000000 ....KSDS4 B.....
00000000 00000000 D2E2C4E2 F6404040 C2000000 00002000 00000000 00002000 .....KSDS6 B.....
00000000 00000000 00000000 D2E2C4E2 F7404040 C2000000 00002000 00000000 .....KSDS7 B.....
00002000 00000000 00000000 00000000 D2E2C4E2 F1F64040 C2000000 00002000 .....KSDS16 B.....
00000000 00002000 00000000 00000000 00000000 D2E2C4E2 F2F04040 C2000000 .....KSDS20 B...

```

Extent Matrix

The output matrix consists of the three parts:

- Header.
- Physical Device Characteristics Area which contains information about the volumes where related dataset is allocated. **Note:** VSAM requires that all extents for a specific cluster component reside on the same type of DASD. For KSDS and VRDS clusters, the data and index can reside on different types of DASD, so there will be two sets of Physical Device Characteristics, one set used for data and the other used for index.
- Extent Information Area which contains information about each extent for the requested VSAM cluster. Data extents will be listed first, marked with 'D', followed by the index extents, marked with 'I'.



Extent Matrix (Header layout)

Information in the header shows the 'map' of the retrieved data. It should be used to parse the returned values and to provide large enough area to obtain all information.

Header has a fixed size of 32 bytes and contains the following fields:

Field	Length
Length of area supplied by user	4 bytes
Total length used (or required) by VSAM	4 bytes
Length of physical device characteristics area	4 bytes
Number of data extents	4 bytes
Length of data extents row	2 bytes
Number of index extents	4 bytes
Length of data extents row	2 bytes
(reserved)	4 bytes
(reserved)	4 bytes

Extent Matrix (Header layout)

Header fields description:

- **Length of area supplied by user** contains the length of the area passed by user in the LENGTH parameter of the MACRO call in bytes.
- **Total length used (or required) by VSAM** contains the length of the area actually used by or needed for VSAM to display physical device information and extent information, including length of the header, in bytes.
- **Length of physical device information area** contains the length of this area in bytes.
- **Number of data extents** contains the number of data extents for the specified cluster (ACB). It also indicates the number of fixed size rows that are displayed in the user's area.
- **Length of data extents row** contains the length of each row in the extent area in bytes.
- **Number of index extents** contains the number of index extents for the specified cluster (ACB). This number also indicates the number of fixed size rows that are displayed in the user's area.
- **Length of index extents row** contains the length of each row in the extent area in bytes.

Extent Matrix (Physical Device Characteristics Area)

This part contains the physical device characteristics for the indicated cluster. Data volume information is displayed first and is followed by index, if applicable.

Each 48 bytes contain the following fields:

Field	Length
Volume ID	6 bytes
Type of extent	1 byte
Flags	1 byte
Physical block size	4 bytes
Number of bytes per track	4 bytes
Number of bytes per control area	4 bytes
Number of physical blocks per control interval	4 bytes
Number of physical blocks per track	4 bytes
Number of tracks per control area	4 bytes
Number of tracks per cylinder	4 bytes
Number of physical blocks per control area	4 bytes
(reserved)	4 bytes
(reserved)	4 bytes

Extent Matrix (Extent Information Area)

This part shows information about all extents for a specified file. This part consists of fixed size rows, number of which equals number of extents associated with a specified cluster. The length of each row equal to 48 bytes and can be found in the header.

Field	Length
Volser	6 bytes
Type of extent	1 byte
Flags	1 byte
Low extent (CCHH)	4 bytes
(reserved)	4 bytes
High extent (CCHH)	4 bytes
(reserved)	4 bytes
Low RBA	8 bytes
High RBA	8 bytes

Example of an Extent Matrix Call

SHOWCB AREA=*USER_AREA*, LENGTH=300, ACB=*ACB1*, FIELDS=(**EXTINF**)

Example of Extent Matrix output (part 1)

This part of the output shows the Header and Physical Device Characteristics Area.

```

0000012C  10  .....
^=====USER'S AREA=X'12C'=300
000000E0 00000060 00000001 00300000 00010030 00000000 00000000 E5E2C5D9 .....VSER
^=====VOLID=VSER02
RESERVED2
RESERVED1
=====
LEN OF INDEX EXT ROW
IND EXTENTS=1
LEN OF DATA EXT ROW
DATA EXTENTS=1
=====
FIXED AREA LEN=X'60'=96
VSAM NEEDS=X'E0'=224
F0F2C426 00000800 0000A800 0009D800 00000001 00000015 0000000F 0000000F 02D.....y...Q.....
^=====TRACKS PER CYL=X'F'=15
TRACKS PER CA=X'F'=15
PHYS BLOCKS PER TRACK=X'15'=21
PHYS BLOCKS PER CI=X'1'
NUM BYTES PER CA=X'9D800'
NUM BYTES PER TRACK=X'A800'
PHYS BLOCK SIZE=X'800'
FLAGS=X'26'
TYPE OF EXT='D'
0000A800 00000000 00000000 E5E2C5D9 F0F2C926 00000E00 0000B600 0000B600 ..y.....VSER02I.....
^=====NUM BYTES PER CA=X'0000B600'
NUM BYTES PER TRACK=X'0000B600'
PHYS BLOCK SIZE=X'E00'
FLAGS=X'26'
TYPE OF EXT='I'
VOLID=VSER02
RESERVED2
RESERVED1
=====
NUM PHYS BLOCKS PER CA(FBA only)

```

Example of Extent Matrix output (part 2)

This part of the output shows Extent Information Area.

```
00000001 0000000D 00000001 0000000F 0000B600 00000000 00000000 E5E2C5D9 .....VSER
|         |         |         |         |         |         |         ^=====VOLSER=VSER02
|         |         |         |         |         |         |         ^=====RESERVED2
|         |         |         |         |         |         |         ^=====RESERVED1
|         |         |         |         |         |         |         ^=====NUM PHYS BLOCKS PER CA(FBA only)
|         |         |         |         |         |         |         ^=====TRACKS PER CYL=X'F'
|         |         |         |         |         |         |         ^=====TRACKS PER CA=X'1'
|         |         |         |         |         |         |         ^=====PHYS BLOCKS PER TRACK=D
|         |         |         |         |         |         |         ^=====PHYS BLOCKS PER CI=X'1'
^=====
F0F2C415 00070000 00000000 0009000E 00000000 00000000 00000000 00000000 02D.....
|         |         |         |         |         |         |         ^=====HIGH RBA=X'001D87FF'=1935359
|         |         |         |         |         |         |         ^=====LOW RBA=0
|         |         |         |         |         |         |         ^=====HIGH EXTENT=X'0009000E'
|         |         |         |         |         |         |         ^=====LOW EXTENT=X'00070000'
|         |         |         |         |         |         |         ^=====FLAGS=X'15'
|         |         |         |         |         |         |         ^=====TYPE OF EXT='D'
^=====
001D87FF 00000000 00000000 E5E2C5D9 F0F2C915 000A0000 00000000 000A000E ..g.....VSER02I.....
|         |         |         |         |         |         |         ^=====HIGH EXT=X'000A000E'
|         |         |         |         |         |         |         ^=====LOW EXT=X'000A0000'
|         |         |         |         |         |         |         ^=====FLAGS=X'15'
|         |         |         |         |         |         |         ^=====TYPE OF EXT='I'
|         |         |         |         |         |         |         ^=====VOLSER=VSER02
|         |         |         |         |         |         |         ^=====RESERVED2
|         |         |         |         |         |         |         ^=====RESERVED1
^=====
00000000 00000000 00000000 00000000 000AA9FF 00000000 00000000 00000000 .....z.....
|         |         |         |         |         |         |         ^=====RESERVED2
|         |         |         |         |         |         |         ^=====RESERVED1
|         |         |         |         |         |         |         ^=====HIGH RBA=X'000AA9FF'=698879
|         |         |         |         |         |         |         ^=====LOW RBA = 0
```

Thank You



Questions

Please forward your questions or remarks to
zvse@de.ibm.com
zaslonko.mikhail@ru.ibm.com

z/VSE Live Virtual Classes

z/VSE

@ <http://www.ibm.com/zvse/education/>

LINUX + z/VM + z/VSE

@ <http://www.vm.ibm.com/education/lvc/>

Read about upcoming LVCs on @ <http://twitter.com/IBMzVSE>

Join the LVC distribution list by sending a short mail to alina.glodowski@de.ibm.com

