VSAM Update 4.3 & 5.1

Stev Glodowski

WAVV 2012

Covington, Kentucky, USA

http://www.ibm.com/zVSE

http://twitter.com/IBMzVSE

Trademarks

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.

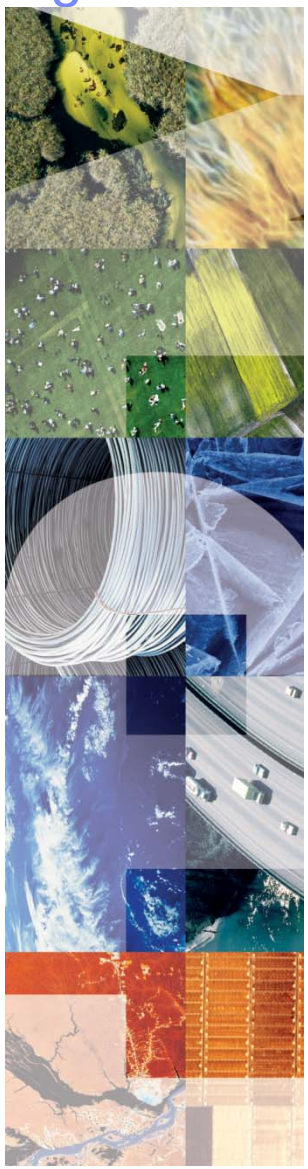
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available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

Agenda

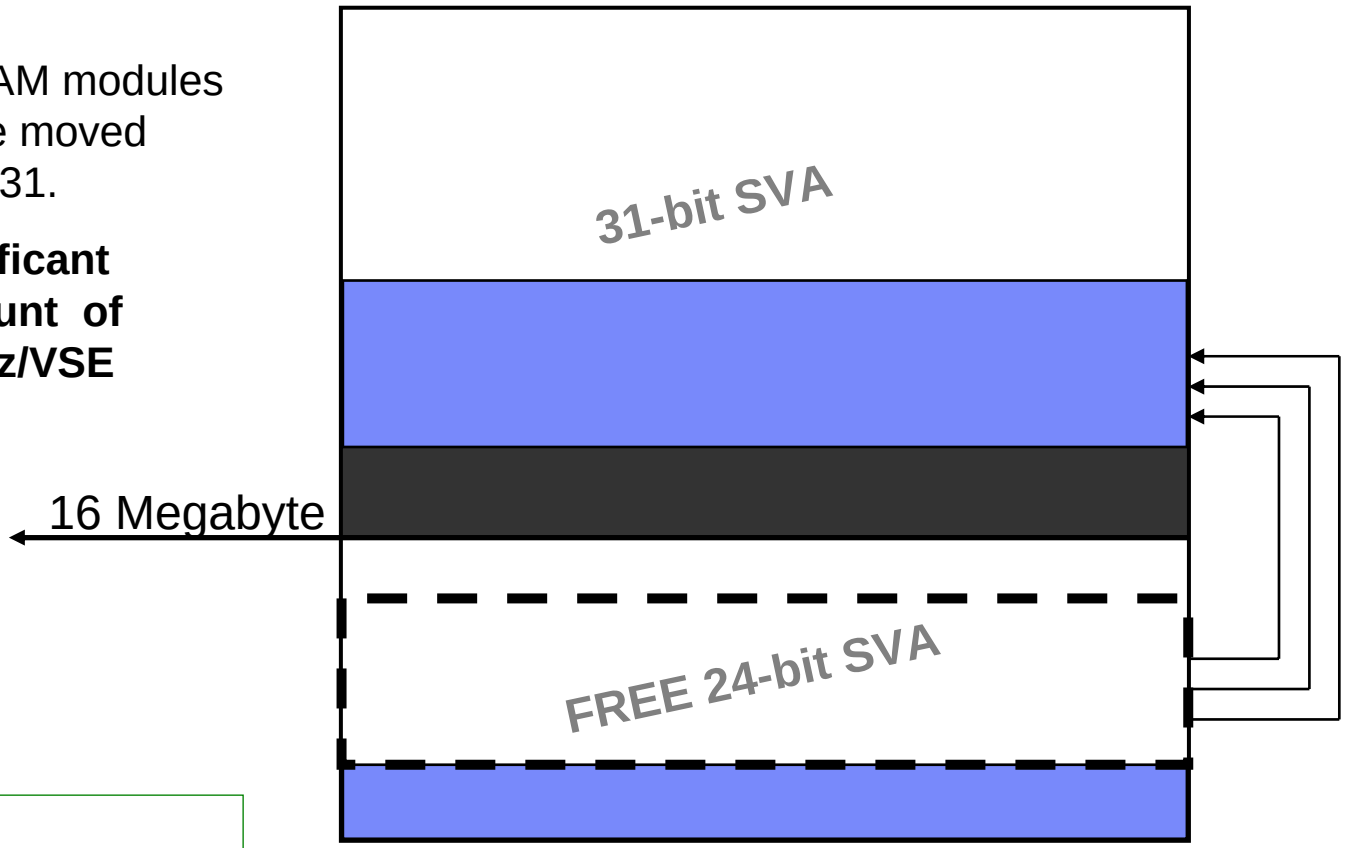


- VSAM 4.3
  - VSAM 24-bit Constraint Relief
  - VSAM SNAP cleanup
  - IDCAMS RECMAP Command Enhancements
  - VSAM using DLBL CYL/BLK
  - New CATLG Interface package
  - VSAM redirector EXCPAD
  - SHOWCB Enhancements
- VSAM 5.1
  - SHOWCB Enhancements
  - IUI improvement for VSAM files
- VSAM Service
  - APARs and PTFs

VSAM 24-bit Constraint Relief

In z/VSE 4.3, VSE/VSAM modules and control blocks were moved from SVA-24 to SVA - 31.

This resulted in **a significant reduction in the amount of SVA - 24 required for z/VSE system phases**.



Advantages:

24-bit GETVIS-shortage relief

VSAM 24-bit Constraint Relief

The following external changes have been made to VSE/VSAM:

1. All modules, except for a couple of small stubs, have been moved into 31-bit SVA.
   Because of the close affinity between VSAM Space Management and BAM, the Space Management modules were converted to AMODE(31), but the modules were left in 24-bit SVA.
   24-bit SVA Savings: 473K, over z/VSE 4.2

2. All VSE/VSAM control blocks, with a few exceptions, have been moved from 24-bit to 31-bit Partition Getvis. **Savings: at least 1.2Meg for 300 files. For non-shared (non-LSR) files, the savings are much greater.**

3. The following $$B-Transients have been loaded into 31-bit SVA in “Move” mode:

- $$BCVS03 Used primarily to call IKQVDUMP for VSAM Traces
- $$BODADE Used to format error messages from VSAM DADSM routines
- $$BODADS See $$BODADE
- $$BTCLOS Called by VSAM TCLOSE macro

VSAM 24-bit Constraint Relief

The following external changes have been made to VSE/VSAM :

4. All Partition GETVIS requests by VSE/VSAM are acquired in GETVIS pools.

   This will allow closer tracking of GETVIS usage by the product

5. VSE/VSAM supports a user-generated control blocks (ACB, RPL, EXLST) as well as the action macros (OPEN, CLOSE, GET, PUT, ERASE, TESTCB, SHOWCB, GENCB) to be located in and executed from 31-bit Partition GETVIS.

   Note: When a 31-bit ACB is passed to either the Open or Close macro, only one ACB may be passed at a time. 24-bit ACBs may be passed in a list, delimited at the end with x'0A'

VSAM 24-bit Constraint Relief

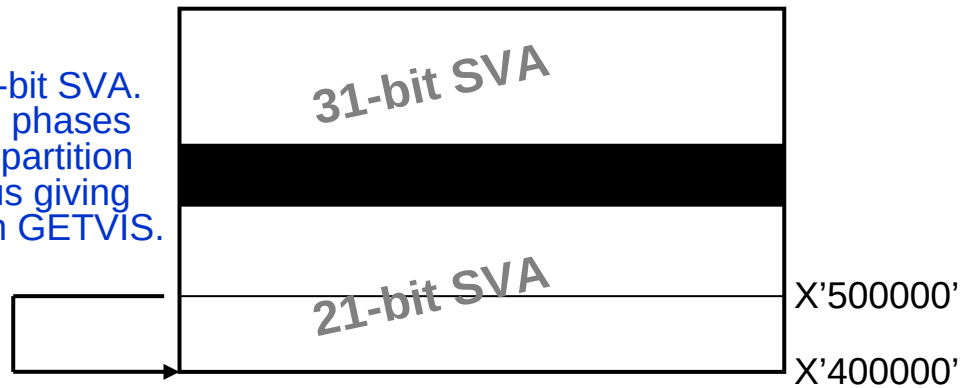
The following external changes have been made to VSE/VSAM:

6. In addition to the above, an option (“BUFDAT=RMODE31”) has been added to the DLBL to allow a legacy application to move the VSE/VSAM data buffers to 31-bit Partition GETVIS.

   Before you use this option, you should ensure your application does not access VSE/VSAM records in "Locate" mode. If so, your application would need to be running in AMODE(31) in order to "see" records residing in 31-bit resident buffers.

Please see description for RPL OPTCD = LOC in Chapter 12 “Descriptions of VSE/VSAM Macros” in "VSE/VSAM Users Guide and Application Programming"

Note: Besides VSAM, that moved code up to 31-bit SVA. Supervisor (Attention routine) moved some large phases into 31-bit area. The net result is that the default partition start was moved from x'500000' to x'400000', thus giving the customer an additional Meg of 24-bit partition GETVIS.



VSAM SNAP trace cleanup

OLD

№	SNAP Trace description
1	Catalog management error code trace + Compression control services trace + Compression management services trace for OPEN and CLOSE ✕
2	Buffer manager trace
3	CLOSE control block dump (at the beginning of CLOSE processing)
4	VSE/VSAM I/O trace
5	I/O error trace
6	OPEN control block dump (when OPEN processing is OK)
7	OPEN error trace (prints control blocks if an error occurs during OPEN processing)
8	Catalog management I/O trace (prints all I/O operations done by VSE/VSAM catalog management)
9	Record management error trace (prints control blocks for any error detected by VSE/VSAM record management)
10	Automatic CLOSE. ✕
11	Managed-SAM control block trace
12	SHAREOPTIONS(4) z/VSE locking activity trace ✕
13	In-core wrap trace for the last sixty file access activities for a file
14	**--- === Future Development === ----**
15	Compression management services control block trace ✕
16	Compression management services trace ✕

NEW

№	SNAP Trace description
1	Catalog management error code trace
2	Buffer manager trace
3	CLOSE control block dump (at the beginning of CLOSE processing) + OPEN control block dump (when OPEN processing is OK) + OPEN error trace (prints control blocks if an error occurs during OPEN processing) + Managed-SAM control block trace
4	VSE/VSAM I/O trace
5	I/O error trace
6	**--- === Future Development === ----**
7	**--- === Future Development === ----**
8	Catalog management I/O trace (prints all I/O operations done by VSE/VSAM catalog management)
9	Record management error trace (prints control blocks for any error detected by VSE/VSAM record management)
10	Redirector trace
11	**--- === Future Development === ----**
12	**--- === Future Development === ----**
13	In-core wrap trace for trace points within VSE/VSAM Record Management
14	Level2 SNAP013 Trace (I/O, EXCPAD and z/VSE Lock Activity)
15	Level3 SNAP013 Trace (Buffer Management)
16	Capture (on SYSLST) a continuous trace of VSE/VSAM Record Access activity.

VSAM SNAP trace cleanup

Customer could enable the following SNAP Traces:

Type:	**Enables**:
0001	Catalog management error code trace
0002	Buffer manager trace
0003	OPEN control block dump (when OPEN processing is complete) OPEN error trace (prints control blocks if an error occurs during OPEN processing) CLOSE control block dump (at the beginning of CLOSE processing)
0004	VSE/VSAM I/O trace
0005	I/O error trace
0008	Catalog management I/O trace (prints all I/O operations done by VSE/VSAM catalog management)
0009	Record management error trace (prints control blocks for any error detected by VSE/VSAM record management)
0010	Redirector Trace
0013	In-core wrap trace for trace points within VSE/VSAM Record Management
0014	Level2 SNAP013 Trace (I/O, EXCPAD and zVSE Lock Activity)
0015	Level3 SNAP013 Trace (Buffer Management)
0016	Produce a printout (PDUMP) each time the SNAP013 Trace Table wraps.

VSAM SNAP trace cleanup

Where to find SNAP Traces OUTPUTs:

SNAP Trace №	SNAP Trace OUTPUT
001	To CONSOLE
002-005	To SYSLST
006, 007	Future Development
009, 010	To SYSLST
011	Future Development
012	Reserved
013-015	In-core
016	To SYSLST

Advantage: These changes were made to enhance VSE/VSAM problem resolution capabilities, while at the same time reducing the trace footprint. This will result in less performance impact on customer applications in those cases when we need to use one of these traces to resolve a problem.

VSAM SNAP trace cleanup

How to Run a SNAP Trace:

To activate IKQVEDA from the system console (SYSLOG) or SYSRDR, enter:

```
// EXEC IKQVEDA,PARM='SYSnnn'
```

where SYSnnn specifies how the SNAP commands are entered:

SYSLOG - SNAP commands are entered from the system console (this is the default)

SYSIPT - SNAP commands are read from SYSIPT

To enable the SNAP trace, enter:

```
ENABLE SNAP=00nn,PART=yy,DDNAME=(list of filenames)
```

To disable the SNAP trace after the program is finished, enter:

```
DISABLE SNAP=00nn,PART=yy
```

EXAMPLE:

```
// EXEC IKQVEDA,PARM='SYSIPT'
    ENABLE SNAP=003
/*
**** run your program ***
// EXEC IKQVEDA,PARM='SYSIPT'
    DISABLE SNAP=003
/*
```

Please find details about SNAP Traces in Appendix F “Diagnosis Tools” in “VSE/VSAM User’s Guide and Application Programming”.

IDCAMS RECMAP Command Enhancements

New parameter DECIMALPOS for IDCAMS RECMAP command is implemented to specify the position of decimal point for decimal numbers. Default is 0.

This parameter can only be applied to the field types:

PACKED, UNPACKED, ZONED, and UNZONED.

```
Examples:
        123.45          has a decimal position of 2
        12345           has a decimal position of 0 (or no decimal position)
        1234500         has a decimal position of -2
```

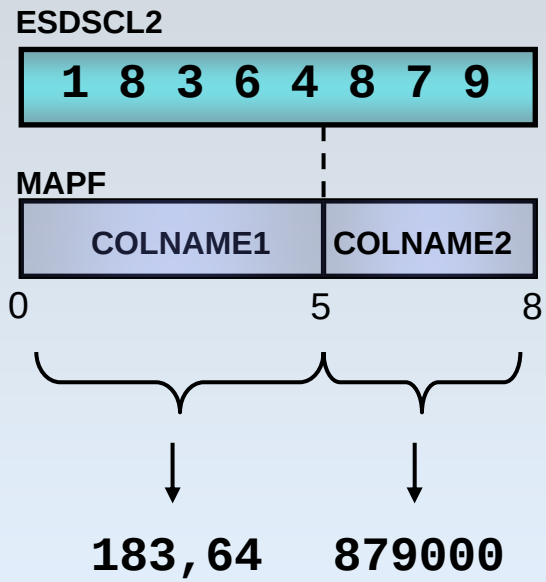
Advantage: Such decimal numbers are used by customer applications on COBOL or PL/1 to do calculations with decimal values.
For example, for storing currency, there are usually 2 digits after the decimal point for cents, e.g. 123.45 €

IDCAMS RECMAP Command Enhancements

RECMAP DECIMALPOS Example:

```
* DEFINE RECMAP MAPF FOR CLUSTER ESDSCLS2 ON UCAT1:
// EXEC IDCAMS,SIZE=AUTO
1S54I  PHASE IDCAMS   IS TO BE FETCHED FROM IJSYSRS.SYSLIB
IDCAMS  SYSTEM SERVICES
RECMAP -
    DEFINE ( MAP(MAPF     )                              -
      MAPCOLUMN(                                        -
       (COLNAME1                                        -
          FIELD(OFFSET(0), LENGTH(5),                  -
          TYPE(PACKED),DECIMALPOS(2) )                  -
        )                                               -
       (COLNAME2                                        -
          FIELD(OFFSET(5), LENGTH(3), TYPE(PACKED), -
          DECPOS(-3) )                                  -
          DESC(DESCRITION_FOR_FIELD_#_2))               -
          ))                                            -
          CATALOG(UCAT1)                                -
          CLUSTER(ESDSCLS2)

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
1S55I  LAST RETURN CODE WAS 0000
```



IDCAMS RECMAP Command Enhancements

OUTPUT of RECMAP LIST command to the defined MAP “MAPF”:

```
 RECMAP LIST (CLUSTERS    ) -
          CATALOG(UCAT1)
CATALOG ----------------- UCAT1
  CLUSTER --------------- ESDSCLS2
    MAP ----------------- MAPF
        COLUMN ---------- COLNAME1
            POSITION --------          1
            LENGTH ----------          5  X'00000005'
            OFFSET ----------          0  X'00000000'
            TYPE ------------ PACKED
            DESCRIPTION -----
            DECIMALPOS ------         +2  X'00000002'
        COLUMN ---------- COLNAME2
            POSITION --------          2
            LENGTH ----------          3  X'00000003'
            OFFSET ----------          5  X'00000005'
            TYPE ------------ PACKED
            DESCRIPTION ----- DESCRITION_FOR_FIELD_#_2
            DECIMALPOS ------         -3  X'FFFFFFFD'
```

IDCAMS RECMAP Command Enhancements

Two new parameters MAPNAMES and CLUSTERNAMES, are implemented for IDCAMS RECMAP LIST command to produce a list of map names.

Example of CLUSTERNAMES:

```
// EXEC IDCAMS,SIZE=AUTO
   RECMAP -
      LIST ( CLUSTERNAMES     )
/*
```

OUTPUT:

```
UCAT1
   CLUSTER.NAME.A
                MAP.NAME.AA
                MAP.NAME.BB
   CLUSTER.NAME.B
                MAP.NAME.CC
                MAP.NAME.DD
UCAT2
   CLUSTER.NAME.V
                MAP.NAME.LL
                MAP.NAME.MM
```

Example of MAPNAMES:

```
// EXEC IDCAMS,SIZE=AUTO
   RECMAP -
      LIST ( MAPNAMES     ) -
      CATALOG(UCAT1)
/*
```

OUTPUT:

```
UCAT1
  MAP.NAME.AA -------- CLUSTER.NAME.A
  MAP.NAME.BB -------- CLUSTER.NAME.A
  MAP.NAME.CC -------- CLUSTER.NAME.B
  MAP.NAME.DD -------- CLUSTER.NAME.B
```

VSAM using DLBL CYL/BLK

The DLBL statement supports the following new operands:

CYL=n CYL=(n,n1)

This operand allows the space allocation on a CKD device to be defined using number of cylinders.

BLK=n BLK=(n,n1)

This operand allows the space allocation on an FBA device to be defined using number of blocks.

n specifies the number of blocks used for the primary allocation,

n1 specifies the number of blocks used for secondary allocations.

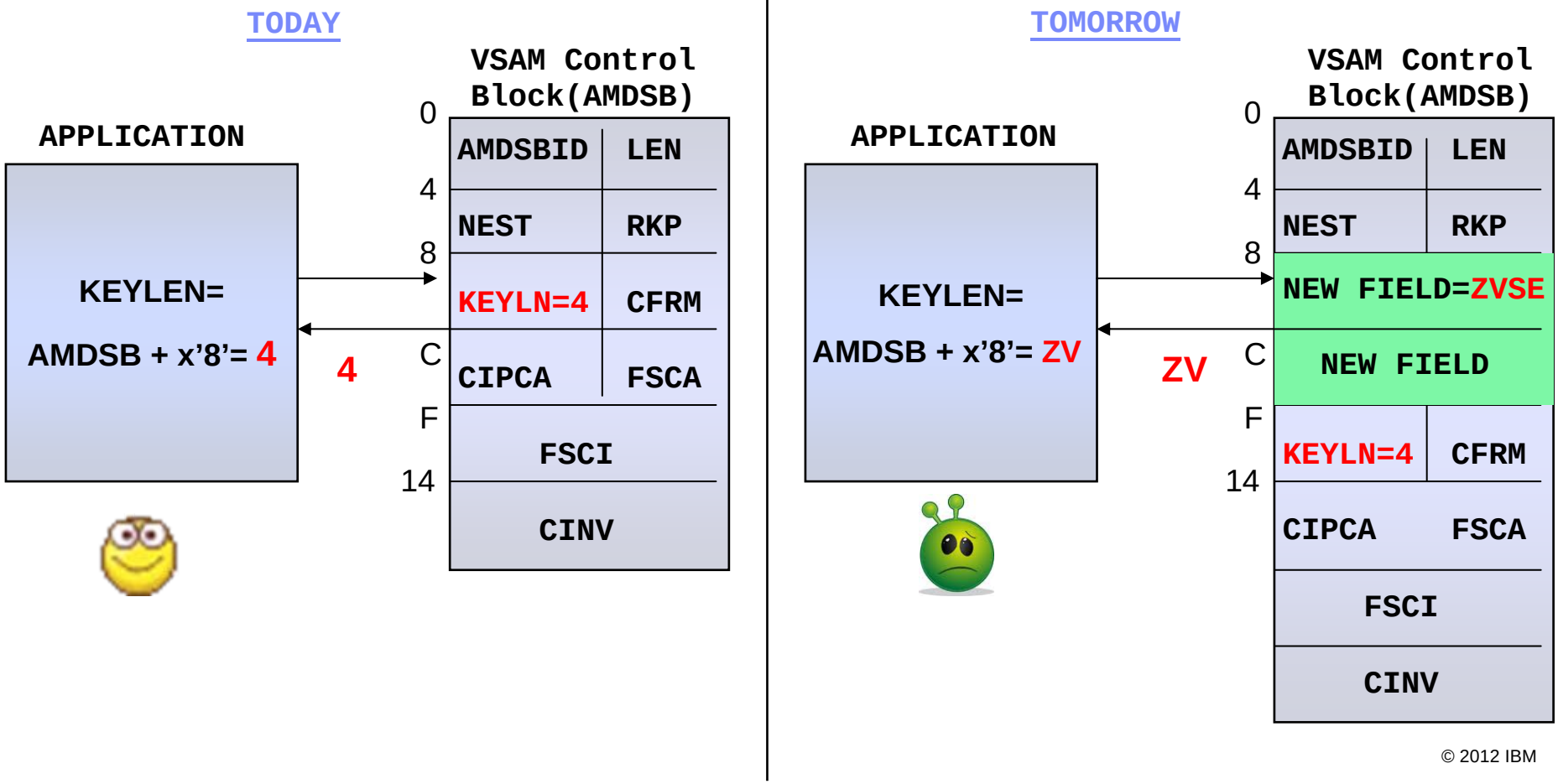
n and n1 can be a decimal number up to 2,147,483,645.

Example of DLBL statements:

```
// DLBL VSMFIL1,'VSMFILE-ID 1',0010,VSAM,CYL=10
// DLBL VSMFIL2,'VSMFILE-ID 2',0020,VSAM,BLK=(20000,10000)
```

New CATLG API

An API is a method of extracting information from VSAM for the vendor application itself. The most commonly used APIs are the VSAM Request Macros: Open, Close, GET, PUT, etc. zVSE Development and Support strongly discourages customer (or vendor) applications from accessing internal VSAM control blocks directly. Control Blocks change, as the VSAM product evolves.



New CATLG API

The most interesting APIs for Vendor Applications are those which supply information about a cluster for a vendor application, without the vendor application needing to know the format of an internal control block. Traditionally, VSE/VSAM has offered:

SHOWCAT (shows information about a cluster prior to Open), Returns Key Position, Key Length, CI Size, Maximum Record Size, and the CI numbers for associated catalog records (Index, Data, Cluster, Alternate Index, Path).

SHOWCB (information after Open, or before/after a record management request).

TESTCB (a special subset of SHOWCB which returns "equal", or "not equal", to allow more efficient assembler coding).

GENCB (allows dynamic creation of a VSE/VSAM external control block (ACB, RPL or Exit list).

MODCB (allows dynamic modification of the options in the ACB, RPL, or Exit list), MODCB of an ACB requires the ACB to be closed. An RPL must not be active.

PRODEXIT (allows monitoring of allocations of VSAM data space extents and the suballocation of VSAM cluster extents).

CATLG (shows general information by calling the Catalog Management phase directly).
***** NEW *****

New CATLG API

The CATLG API is designed to be only **read-only**.

It cannot be used to define Clusters or Volumes, or change the attributes of previously-defined Catalog Entries. The following macros will be provided:

CATLG (Call the Catalog Management entry phase)

CTGPL (Definition of a Catalog Parameter List)

CTGFL (Definition of a Field Parameter List). A CTGPL may point to a series of FPLs, but all FPLs must relate to the same catalog record. Each FPL is used to extract a single piece of data (field) from a particular catalog record.

Prior to executing the CATLG Macro, the vendor application defines the fields which should be retrieved by Catalog Management by defining a Catalog Parameter List and at least one Field Parameter List, using the DSECTs provided by the IKQCTGPL and IKQCTGFL macros. Multiple fields may be retrieved in a single call by chaining an equivalent number of FPLs to the CTGPL.

New CATLG API

An external version of the following macros are being distributed in the z/VSE 4.3 ICCF Library 59 and on the zVSE home page.

- SKCTGPL (Catalog Parameter List => Generates IKQCTGPL.A in PRD2.CONFIG)
- SKCTGFL (Catalog Field Entry => Generates IKQCTGFL.A in PRD2.CONFIG).
- SKCATLG (Invocation Macro => Generates CATLG.A in PRD2.CONFIG)

.

```
IESLIBP                          PRIMARY LIBRARY                          PAGE 12 of 30

PRIMARY (READ/WRITE):     59                                        PREFIX:

OPTIONS:    1 = EDIT      2 = CHANGE     3 = PRINT      4 = COPY       5 = DELETE
            6 = RENAME    7 = SUBMIT     8 = COMPILE    9 = DISPLAY

 OPT   MEMBER NAME    NEW NAME     NEW LIB   LAST ACCESSED   OWNER    PASSW   PRIVATE

  _      SKCOMVAR     _________    _____       06/22/2010    AAAA             _
  _      SKCONSII     _________    _____       06/22/2010    AAAA             _
  _      SKCPSTP      _________    _____       06/22/2010    AAAA             _
  _      SKCRESTP     _________    _____       06/22/2010    AAAA             _
  _      SKCSDFC2     _________    _____       06/22/2010    AAAA             _
  _      SKCSDFIL     _________    _____       06/22/2010    AAAA             _
  *      SKCTGFL      _________    _____       06/22/2010    AAAA             _
  _      SKCTGPL      _________    _____       06/22/2010    AAAA             _
  _      SKDB2SPS     _________    _____       06/22/2010    AAAA             _
  _      SKDB2STR     _________    _____       06/22/2010    AAAA             _

PF1=HELP        2=REFRESH    3=END         4=RETURN                    6=ADD MEMBER
PF7=BACKWARD    8=FORWARD    9=SORT.DATE 10=SORT.NAME 11=SORT.SIZE 12=LIST QUEUE

LOCATE MEMBER/LIST QUEUE PREFIX ==> _________    MEMBER PREFIX (PF2) ==> ________
```

New CATLG API

- The macros are provided in ASSEMBLER format.
- A sample program SKVSMS1 is included. It provides an easy-to-use utility to retrieve any of the supported fields from a catalog, and also serves as an example for customer applications using the CATLG interface.

The provided JCL will compile and catalog this phase into a library / sub-library of the customer's choice. Just update the following statement at the end of the member:

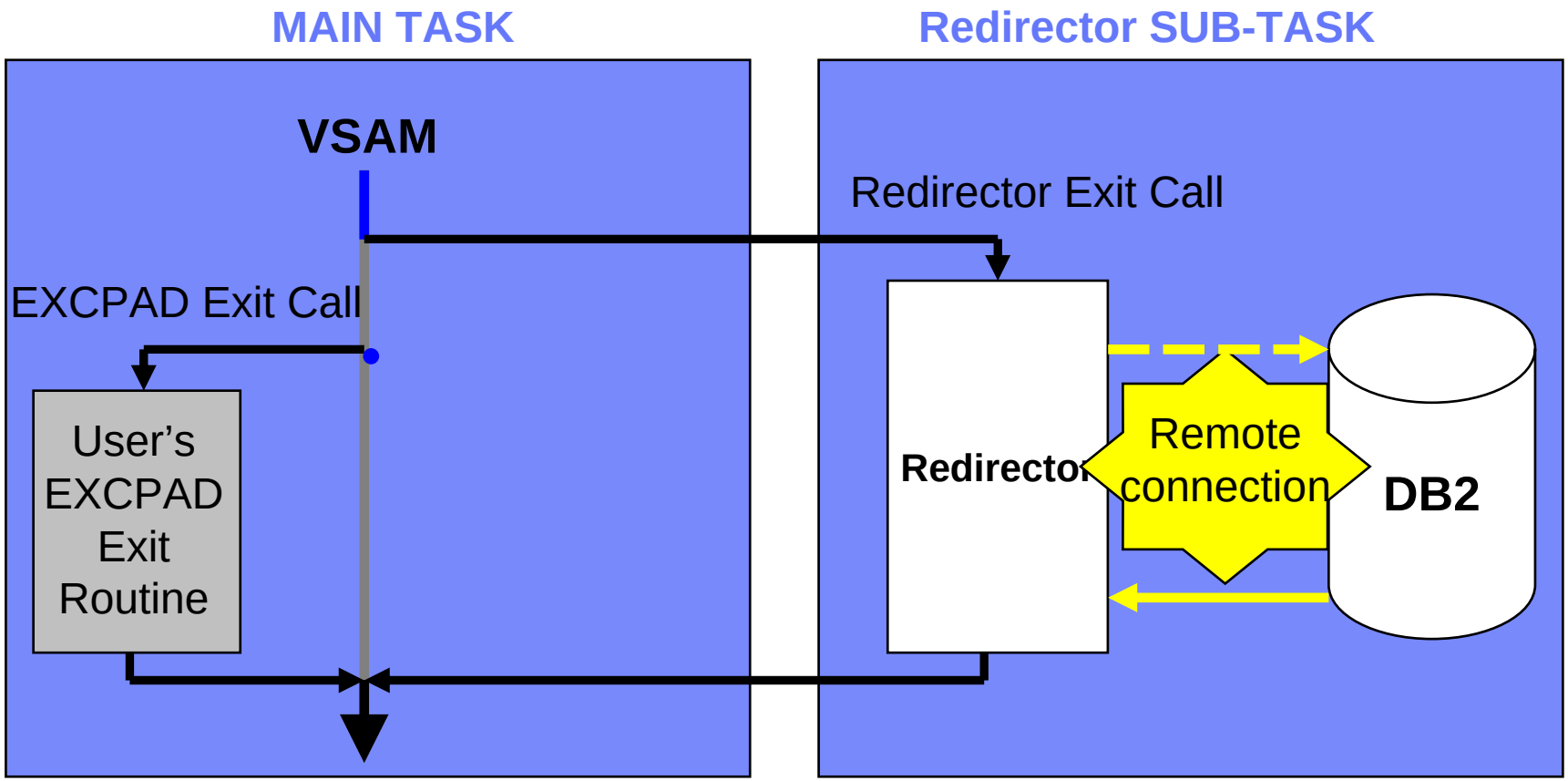
// LIBDEF PHASE,CATALOG=<LIB.SUBLIB>.

A sample program (SKVSMS1) shipped with z/VSE 4.3 as an I.Book and in the ICCF Library 59. The complete API package, along with additional information, is also available for download on the z/VSE web-page.
http://www-03.ibm.com/systems/z/os/zvse/downloads/samples.html#vsam
(You could download it now)

VSAM Redirector EXCPAD

EXCPAD user exit support for VSAM Redirector.

When the EXCPAD exit routine is used, then the Redirector call is performed in a separate subtask. This allows VSAM to continue processing concurrently by returning to the EXCPAD exit routine, while the Redirector task waits for a remote connection.



VSAM Redirector EXCPAD

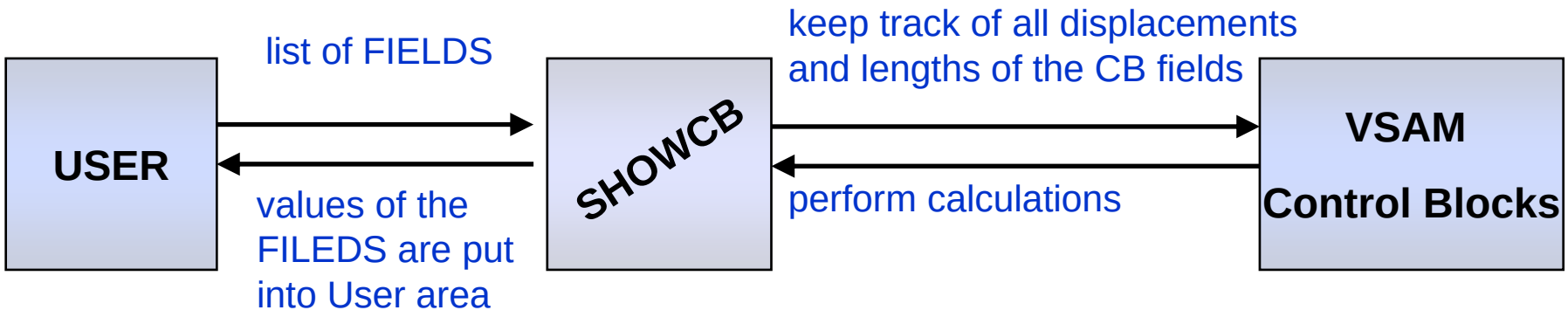
VSAM Redirector EXCPAD details and restrictions.

- Only one Redirector sub-task per partition, even if multiple files are redirected
- Standard EXCPAD interface support
- Designed primarily for CICS/TS
- Redirector Exit Work Area was combined with EXCPAD Exit Work Area.
- Redirector Dispatch Queue implementation.

Advantages: Easy to use - no changes required to CICS/TS Parallel tasking support

SHOWCB Enhancements 4.3

With SHOWCB macro you can examine the contents of fields in an ACB, EXLST, or RPL control block. VSE/VSAM displays the requested fields in a user's area.



Advantage: eliminate customer's dependency on internal VSAM control blocks

SHOWCB Enhancements 4.3

AMDSB and EDB API are extended in order to enable user to obtain status information for open VSAM datasets.

18 NEW AMDSB and EDB FIELDS are supported by SHOWCB ACB.
The following new FIELDS are supported as SHOWCB ACB:

SHOWCB FIELD	Actual FIELD	FIELD Description
ASTRNUM	RPHDASTR	Number of active strings in pool
STRTOT	RPHDSTNO	Total number of strings in pool

SHOWCB FIELD	Actual AMDSB	FIELD Description
ATRB	AMDATTR, AMDATTR3, AMDRCFRM	Dataset Attributes and SAM ESDS record info
BLREC	AMDBLREC	SAM LRECL for SAM ESDS
NSLOT	AMDNSLOT	Number of RRDS slots
SSRBA	AMDSSRBA	RBA of first sequence set index record
SHAREOP	AMDSHOPT	SHARE OPTIONS
SYMU	EDBSYMU	Symbolic unit of the first EDB

SHOWCB Enhancements 4.3

SHOWCB FIELD	Actual AMDSB	FIELD Description
LNEXCP	AMDLEXCP	Local number of EXCPS
LNCIS	AMDLNCIS	Local number of CI SPLITS
LNSSS	AMDLNCAS	Local number of CA SPLITS
LNRETR	AMDLRETR	Local number of retrieved records
LNUPDR	AMDLUPR	Local number of updated records
LNINSR	AMDLIREC	Local number of inserted records
LNDELR	AMDLDELR	Local number of deleted records
LNLOGR	AMDLNLR	Local number of logical records
LAVSPAC	AMDLASPA	Local number of bytes of free space

Please find details about new SHOWCB FIELDS in Chapter 12 “Descriptions of VSE/VSAM Macros” in “VSE/VSAM User’s Guide and Application Programming”.

SHOWCB Enhancements 4.3

SHOWCB AMDSB API Example:

```
SHOWCB   ACB=ACB1,AREA=AREA1,LENGTH=100,FIELDS=(BLREC,NSLOT,    X
        SSRBA,SHAREOP,SYMU,ATRB,NIXL,NSSS,NCIS,LNEXCP,          X
        LNCIS,LNSSS,LNRETR,LNUPDR,LNINSR,LNDELR,LNLOGR,LAVSPAC)
LTR      R15,R15
BNZ      SHOWERR
. . .
AREA1    DS 0F
BLREC    DS F
NSLOT    DS F
SSRBA    DS F
SHAREOP  DS F
SYMU     DS F
ATRB     DS F
ATRB2    DS F
NIXL     DS F
NSSS     DS F
NCIS     DS F
LNEXCP   DS F
LNCIS    DS F
LNSSS    DS F
LNRETR   DS F
LNUPDR   DS F
LNINSR   DS F
LNDELR   DS F
LNLOGR   DS F
LAVSPAC  DS F
```

SHOWCB Enhancements 4.3

SHOWCB LSR API Example:

```
SHOWCB   ACB=ACB2,AREA=AREA2,LENGTH=10,FIELDS=(ASTRNUM,STRTOT)
LTR      R15,R15
BNZ      SHOWERR
. . .
AREA2    DS 0F
ASTRNUM  DS F
STRTOT   DS F
```

Every new FIELD will return 4 bytes except ATRB, which will return 8 bytes:

1 byte	AMDATTR1	ATTRIBUTES (FIRST BYTE)	
2 byte	AMDATTR2	ATTRIBUTES (SECOND BYTE)	
3 byte	AMDATTR3	ATTRIBUTES (THIRD BYTE)	
4 byte	AMDATTR4	ATTRIBUTES (FORTH BYTE)	
5 byte	AMDRCFRM	SAM ESDS RECFM INFORMATION	
6 byte	reserved	- || -	
7 byte	reserved	- || -	
8 byte	reserved	- || -	

SHOWCB Enhancements 5.1

ACB, AMBL and AMDSB API are extended in order to enable user to obtain status information for open VSAM datasets.

9 NEW FIELDS are supported by SHOWCB ACB starting 5.1.
The following new FIELDS are supported as SHOWCB ACB:

SHOWCB FIELD	Actual Control Block Field	Control Block	Length	FIELD Description
IDACB	ACBID	ACB	4	ACB identifier
IDDOS	ACBDOSID	ACB	4	DOS DTF identifier
CDBUF	AMBDBUF	AMBL	4	count of Data Buffers
CIBUF	AMBIBUF	AMBL	4	count of Index Buffers
CNAME	AMBCNAME	AMBL	44	Cluster ID
CIPCA	AMDCIPCA	AMDSB	4	number of CIs per CA
LNEST	AMDLNEST	AMDSB	4	local number of index levels
BFREE	AMDBFREE	AMDSB	4	number of unassigned buffers
OPENOBJ	AMDAMS	AMDSB	4	AMS Flag byte

SHOWCB Enhancements 5.1

SHOWCB Example:

```
SHOWCB   ACB=ACB1,AREA=AREA1,LENGTH=100,FIELDS=(IDACB,IDDOS,      X
        CDBUF,CIBUF,CIPCA,LNEST,BFREE,OPENOBJ,CNAME)
LTR      R15,R15
BNZ      SHOWERR
. . .
AREA1    DS 0F
IDACB    DS F
IDDOS    DS F
CDBUF    DS F
CIBUF    DS F
CIPCA    DS F
LNEST    DS F
BFREE    DS F
OPENOBJ  DS F
CNAME    DS 44CL
```

SHOWCB Enhancements 5.1

LSR Matrix

Local Shared Resource (LSR) information is provided within a new SHOWCB matrix that contains the following information about specific VSAM SHR pools:

For a specified share pool:
- Share Pool Number,
- Total Number of Strings,
- Number of active Strings,
- Number of free Strings,
- High-water-mark of active Strings

For each sub-pool:
- Size of Buffers,
- Type of Buffer,
- Number of Buffers,
- Number of modified Buffers and Number of free Buffers,
- Number of Buffer-reads,
- Number of Retrieval-Requests without I/O,
- Number of User-Initiated writes from Buffer Pool,
- Number of Non-User-Initiated writes from Buffer Pool

For each cluster the following information will be provided:
- Number of Active Strings for this Cluster,
- Size of Data Buffers,
- Number of Data Buffers used,
- Size of Index Buffers,
- Number of Index Buffers used

SHOWCB Enhancements 5.1

LSR Matrix output (header):

Header contains the following information:

- Length of area supplied by User,
- Total length used (required) by VSAM,
- Length of fixed area (Share Pool Statistics Area),
- Number of rows in LSR Pool Buffer Matrix
- Length of rows in LSR Pool Buffer Matrix
- Number of rows in Cluster Matrix
- Length of rows in Cluster Matrix

Length of area supplied by User	Total length used (or required) by VSAM	Length of fixed area	Number of rows in LSR Pool Buffer Matrix
4 bytes	4 bytes	4 bytes	4 bytes

… continued

Len of rows in Buffer Matrix	Number of rows in Cluster Matrix	Length of rows in Cluster Matrix	(reserved)	(reserved)
2 bytes	4 bytes	2 bytes	4 bytes	4 bytes

SHOWCB Enhancements 5.1

LSR Matrix output (Share Pool Statistics Area,fixed area):

For a specified share pool:

- Share Pool Number,
- Total Number of Strings,
- Number of active Strings,
- Number of free Strings,
- High-water-mark of active Strings

share pool #	total # of strings	# of active strings	# of free strings
2 bytes	2 bytes	2 bytes	2 bytes
... continued			
High water mark of active strings	reserved	reserved	reserved
2 bytes	2 bytes	2 bytes	2 bytes

SHOWCB Enhancements 5.1

LSR Matrix output (LSR Pool Buffer Matrix):

For each sub-pool:

- Size of Buffers,
- Type of Buffer,
- Number of Buffers,
- Number of modified Buffers and Number of free Buffers,
- Number of Buffer-reads,
- Number of Retrieval-Requests without I/O,
- Number of User-Initiated writes from Buffer Pool,
- Number of Non-User-Initiated writes from Buffer Pool

Size of buffers	Type of Buffer ("D" or "I")	Flags	Number of buffers	Number of modified buffers	Number of free buffers
2 bytes	1 byte	1 byte	4 bytes	4 bytes	4 bytes

… continued

NUMBER OF BUFFER-READS	NUMBER OF RETR-REQ WITHOUT I/O	NUMBER OF USER-INITIAT WRITES FROM BP	NUMBER OF NON USER-INI WRITES FROM BP
4 bytes	4 bytes	4 bytes	4 bytes

SHOWCB Enhancements 5.1

LSR Matrix output (Cluster Matrix):

For each cluster the following information will be provided:

- DDNAME of the cluster
- Cluster type ('B' if base cluster)
- Number of Active Strings for this Cluster,
- Size of Data Buffers,
- Number of Data Buffers used,
- Size of Index Buffers,
- Number of Index Buffers used

DDNAME	Type of Cluster ('B' if Base Cluster)	Flags	# of Active Strings for this Cluster	Size of Data Buffers	Number of Data Buffers used	Size of Index Buffers
8 bytes	1 byte	1 byte	2 bytes	4 bytes	4 bytes	4 bytes

… continued

Number of Index Buffers used	(reserved)	(reserved)
4 bytes	4 bytes	4 bytes

SHOWCB Enhancements 5.1

LSR Matrix

The new LSR MATRIX and Extent Information MATRIX can be specified using the SHOWCB macro. The syntax of the SHOWCB macro for LSR matrix is given below:

```
name SHOWCB AREA=address,                    X
            LENGTH=number,                   X
            SHAREPL=number,                  X
            FIELDS=(keywords),               X
            MF=form                          X
```

Example of LSR Matrix call:

*SHOWCB AREA=USER_AREA, LENGTH=100, **SHAREPL**=6, FIELDS=(**LSRINF**)*

SHOWCB Enhancements 5.1

Extent Matrix

A second new matrix has been made available by SHOWCB to present information about extents and device characteristics for a specified cluster.

The physical device characteristics for the indicated Cluster are provided.
The data volume information will come first, followed by the index, if applicable:

- Physical Block Size
- Number of Bytes per Track
- Number of Bytes per Control Area
- Number of Physical Blocks per Control Interval
- Number of Physical Blocks per Track
- Number of Tracks per Control Area
- Number of Tracks per Cylinder
- Number of Physical Blocks per Control Area

For each extent (data and index) of the specified cluster the following information is provided:

- Volume Serial Number
- Type of Extent ('D' if Data. “I” if Index)
- Flags
- Low Extent
- High Extent
- Low RBA
- High RBA

SHOWCB Enhancements 5.1

Extent Matrix output (header):

Header contains the following information:

- Length of area supplied by User,
- Total length used (required) by VSAM,
- Length of fixed area (Physical Device Characteristics Area),
- Number of data extents
- Length of data extents row
- Number of index extents
- Length of index extents row

Length of area supplied by User	Total length used (or required) by VSAM	Length of fixed area	Number of data extents (AMDNEXT)
4 bytes	4 bytes	4 bytes	4 bytes

… continued

Len of data extents row	Number of index extents (AMDNEXT)	Len of ind extents row	(reserved)	(reserved)
2 bytes	4 bytes	2 bytes	4 bytes	4 bytes

SHOWCB Enhancements 5.1

Extent Matrix output (Physical Device Characteristics Area,fixed area):

The physical device characteristics for the indicated Cluster are provided.
The data volume information will come first, followed by the index, if applicable:

- Physical Block Size
- Number of Bytes per Track
- Number of Bytes per Control Area
- Number of Physical Blocks per Control Interval
- Number of Physical Blocks per Track
- Number of Tracks per Control Area
- Number of Tracks per Cylinder
- Number of Physical Blocks per Control Area (for FBA only, ignore for ECKD)

Volume id	Type of extent ('D' if Data. "I" if Index	Flags	Physical Block Size	Number of Bytes per Track	Number of Bytes per Control Area	Number of Physical Blocks per Control Interval
6 bytes	1 byte	1 byte	4 bytes	4 bytes	4 bytes	4 bytes

… continued

Number of Physical Blocks per Track	Number of Tracks per Control Area	Number of Tracks per Cylinder	Number of Physical Blocks per Control Area	Reserved	Reserved
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

SHOWCB Enhancements 5.1

Extent Matrix output (Extent information):

For each extent (data and index) of the specified cluster the following information is provided:

- Volume Serial Number
- Type of Extent ('D' if Data. “I” if Index)
- Flags
- Low Extent
- High Extent
- Low RBA
- High RBA

Volser	Type of extent ('D' if Data. “I” if Index	Flags	Low Extent (CCCCHH)	(reserved)	High Extent (CCCCHH)	(reserved)
6 bytes	1 byte	1 byte	4 bytes	4 bytes	4 bytes	4 bytes

… continued

Low RBA	High RBA	(reserved)	(reserved)
8 bytes	8 bytes	4 bytes	4 bytes

SHOWCB Enhancements 5.1

Extent Matrix

The syntax of the SHOWCB macro for Extent Information Matrix is given below:

```
name SHOWCB ACB=address,
            AREA=address,
            LENGTH=number,
            FIELDS=(keywords),
            MF=form
```

Example of Extent Matrix call:

SHOWCB AREA=*USER_AREA*, LENGTH=*300*, *ACB=ACb1*, FIELDS=(**EXTINF**)

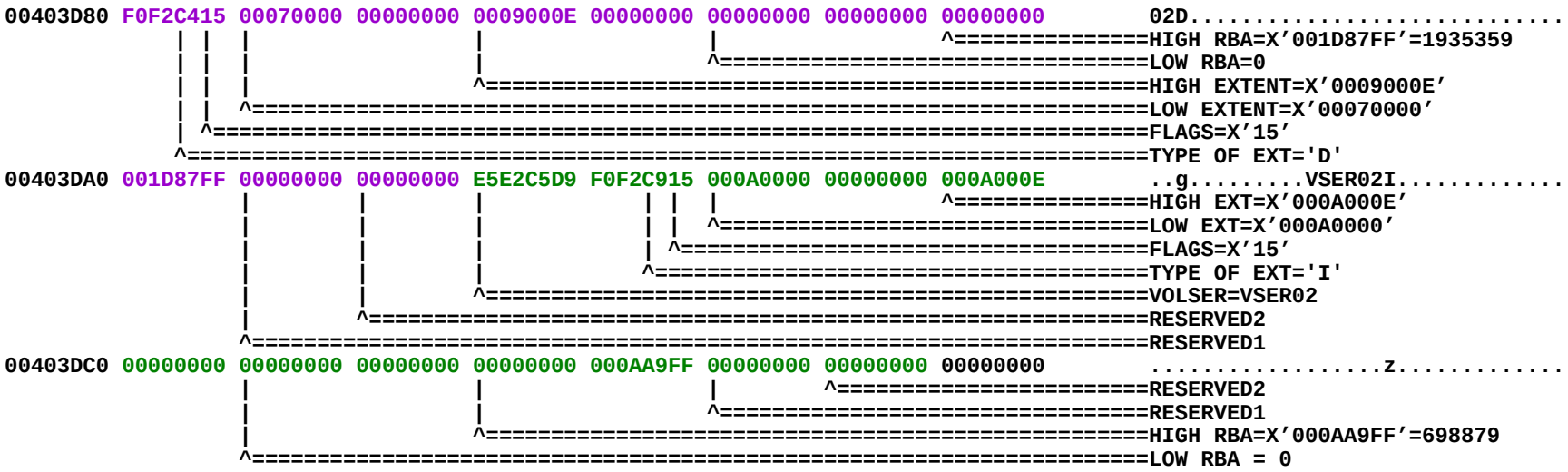
SHOWCB Enhancements 5.1

Example of Extent Matrix output:

```
00403CE0                                                                0000012C   10                                ....
                                                                        ^===============USER'S AREA=X'12C'=300
00403D00 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=96
          ^================================================================================VSAM NEEDS=E0=224
00403D20 F0F2C426 00000800 0000A800 0009D800 00000001 00000015 0000000F 0000000F        02D.......y...Q.................
         | | |    |        |        |        |        |        |        ^===============TRACKS PER CYL=X'F'
         | | |    |        |        |        |        |        ^========================TRACKS PER CA=X'F'
         | | |    |        |        |        |        ^=================================PHYS BLOCKS PER TRACK=X'15'
         | | |    |        |        |        ^=========================================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'
00403D40 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)
00403D60 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'
```

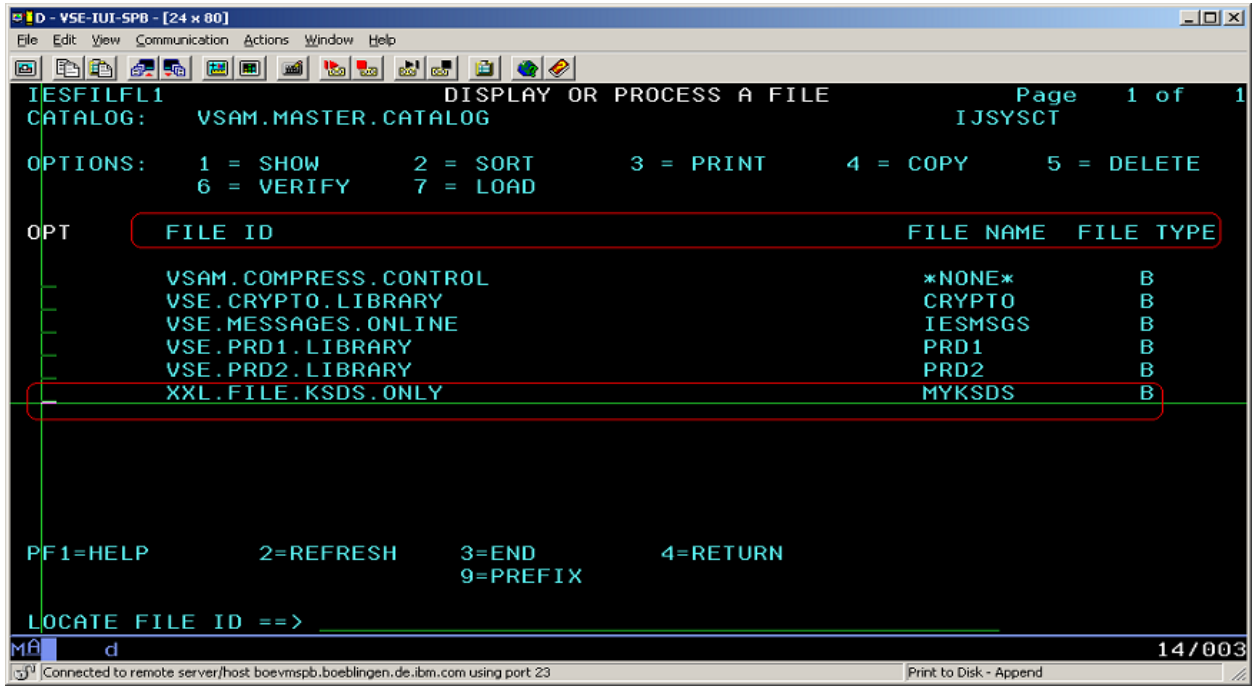
SHOWCB Enhancements 5.1

Example of Extent Matrix output:



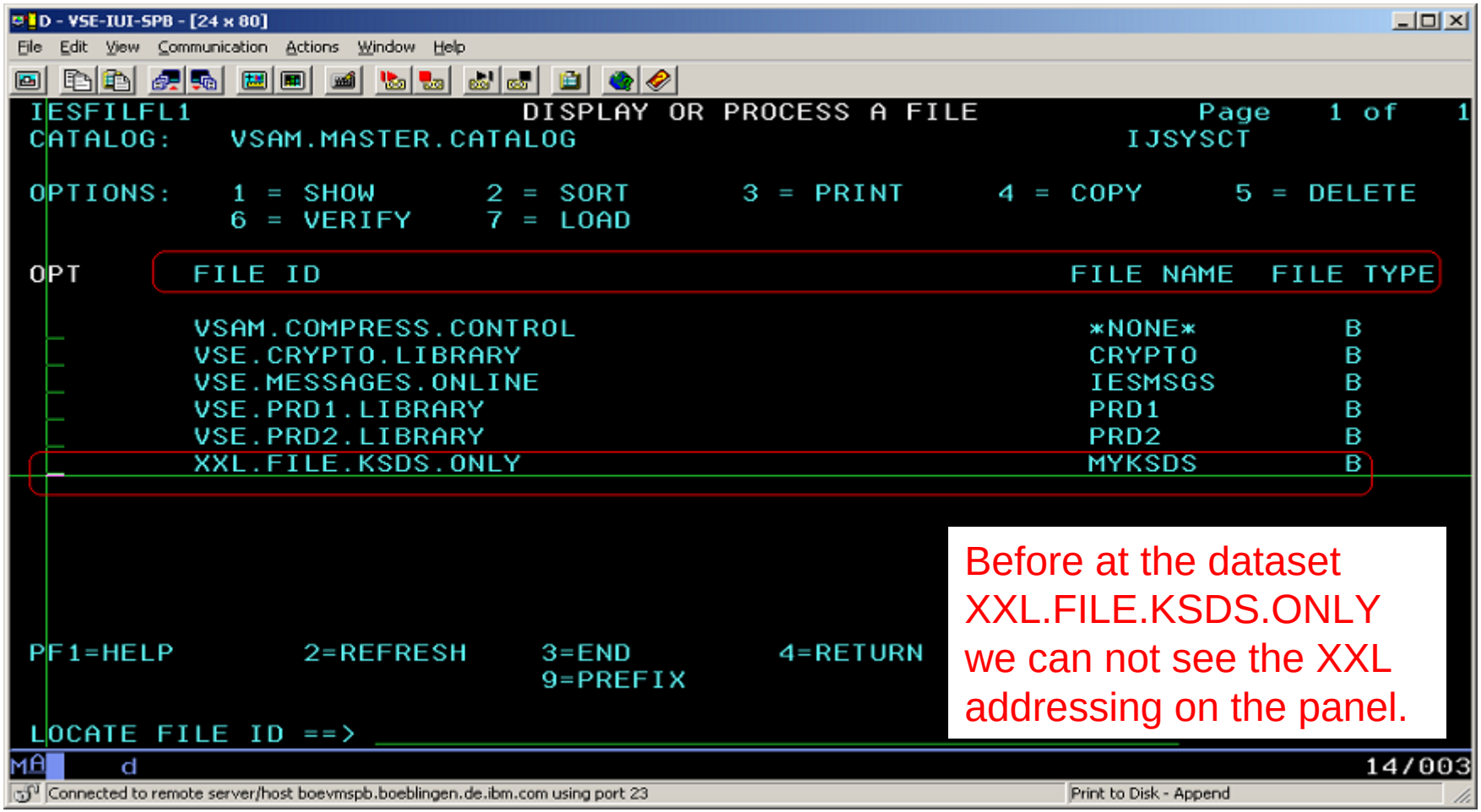
IUI improvements on the VSAM-related panels FILFL1 and FILFL2

- VSAM Addressing Mode listed in IUI
  - Standard or XXL



1. Old View Of The Panel FILFL1

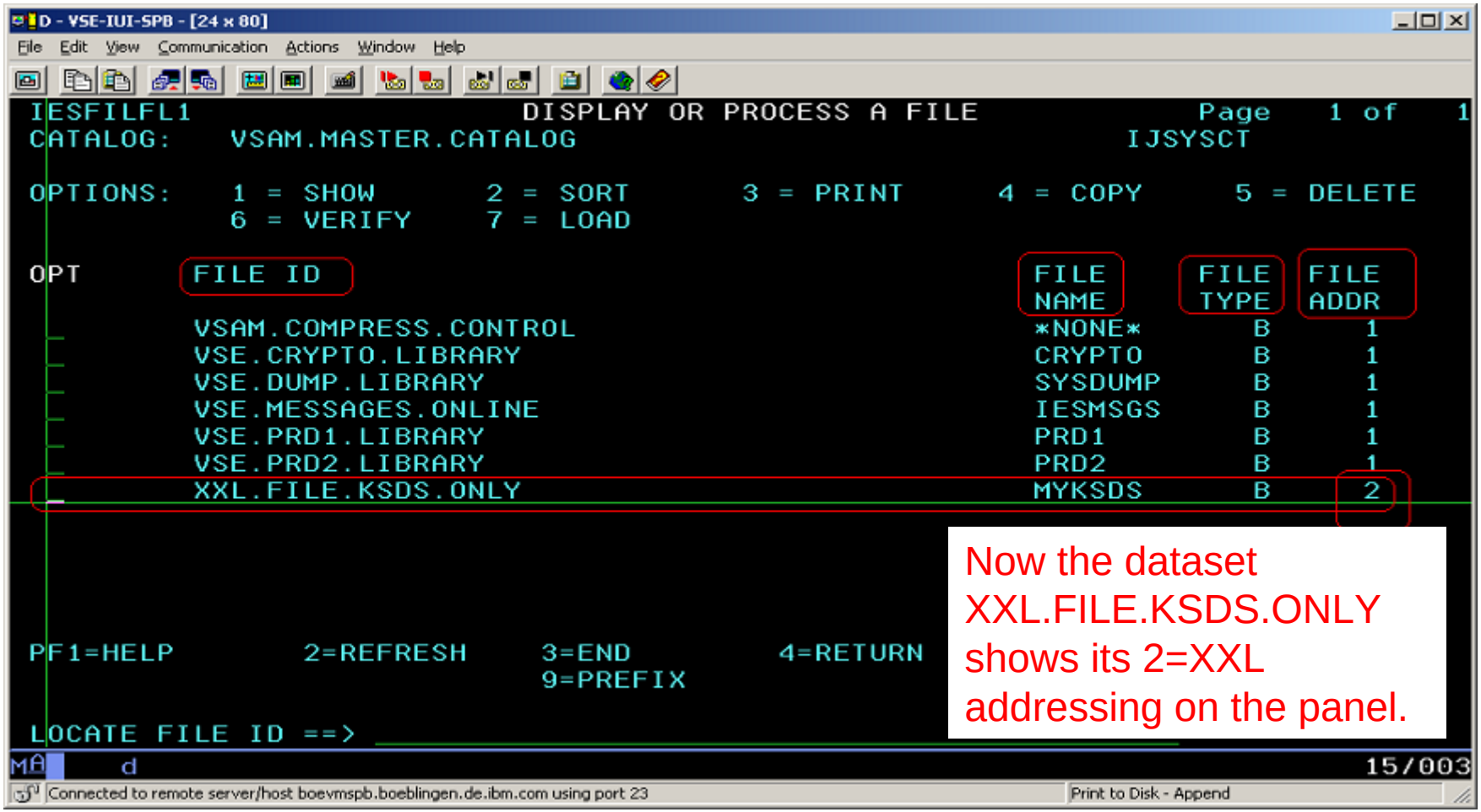
- In z/VSE 4.3 and before the panel FILFL1 looked like:



Before at the dataset XXL.FILE.KSDS.ONLY we can not see the XXL addressing on the panel.

2. New Field 'FILE ADDR' On The Panel FILFL1 – Part 1

- Starting z/VSE 5.1 to show addressing of datasets the panel FILFL1 looks like:

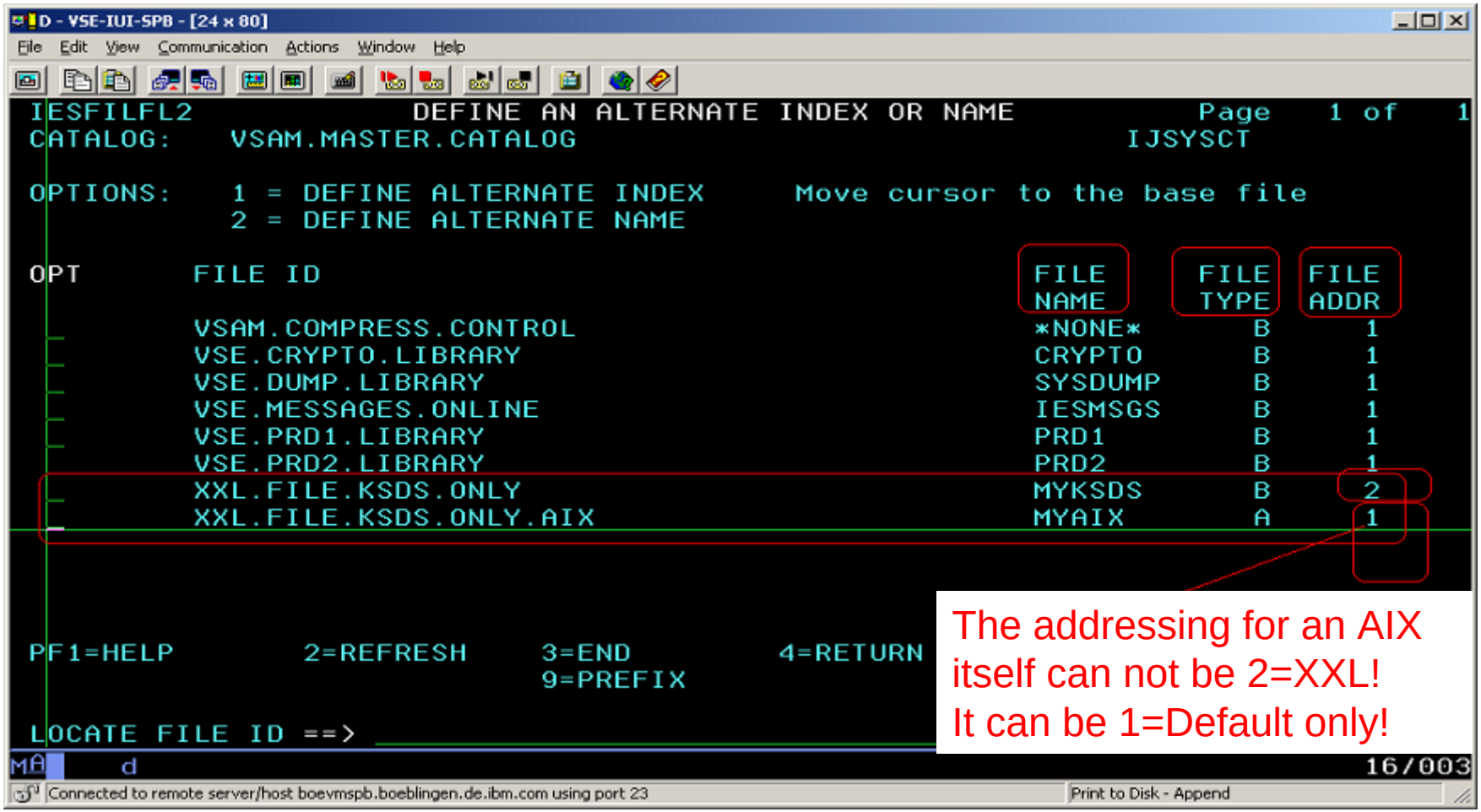


2. New Field 'FILE ADDR' On The Panel FILFL1 – Part 2

- Starting from z/VSE 5.1 the new column 'FILE ADDR' is added to show the corresponded VSAM file addressing:
  - 1 – used for the default addressing,
  - 2 – for XXL addressing (KSDS only).
- Look at the dataset XXL.FILE.KSDS.ONLY to see the XXL addressing samples.
- To accept the new column 'FILE ADDR‘, the panel FILFL1 was re-organized a bit: it contains the same data as before but the existing field 'FILE ID' is shifted to the left; the captions of the columns 'FILE NAME' and 'FILE TYPE' are re-formatted to be more compact on the panel.

3. New Field 'FILE ADDR' On The Panel FILFL2 – Part 1

- The similar changes were done on the panel FILFL2:



3. New Field 'FILE ADDR' On The Panel FILFL2 – Part 2

- To accept the new column 'FILE ADDR‘, the panel FILFL2 was re-organized likes the panel FILFL1: the layout of the panel was re-formatted to show more content.
- Look at the dataset XXL.FILE.KSDS.ONLY for the sample of an XXL dataset on the panel FILFL2.
- NOTE that the addressing for AIXes themselves must have the type 1=Default only for z/VSE 5.1! See for the sample of an AIX addressing at XXL.FILE.KSDS.ONLY.AIX on the panel FILFL2.

PTFs z/VSE 4.3 VSAM (02C)

DY47373	UD53821 IDCAMS VERIFY recovery fix
DY47345	UD53794 PDUMP reduction
DY47335	UD53775 24-BIT GETVIS SHORTAGE
DY47322	UD53774 Recoverable Catalogs PTF
DY47290	UD53761 SHOWCB Fails in Case Fields=HALCRBA and MF=E
DY47262	UD53705 ABEND with Messages 4228I X'B4'(180) (OPNC1-15) CATALOG CLUSTER RECORD NOT FOUND(004,AH,002)
DY47309	UD53749 Restore VSAM Data with CISIZE Less than 8K to SCSI/FBA

PTFs z/VSE 5.1 VSAM (51C)

- **PTF UD53714**
  -RECOVERABLE CATALOGs automatic conversion (DY47322 same as 4.3)
  -SHOWCB correction only regrading the new 5.1 fields (DY47290)

  Is part of the current PSP BUCKET, available for ordering

- **Additional APARs:**
  -EXCPAD 31-bit constraint relief
    - PTF UD53775 for z/VSE 4.3
    - PTF UD53719 for z/VSE 5.1

  -PDUMP reduction, Extent error correction
    - PTF UD53755

  -Backup/Restore SAMESDS
    - PTF UD53813

  -VSAM Clusters created in z/VSE 4.x wih extreme high numbers of extents
    - PTF UD53820

  -IDCAMS VERIFY recovery
    - PTF UD53822

Watch out for future VSAM PTFs at:http://www.ibm.com/zvse/support/vsam.html

Hints and Tips for z/VSE 4.3

- **Several updates on VSAM regarding**
  - Migration
  - Recoverable Catalogs
  - SHOWCB
  - and more

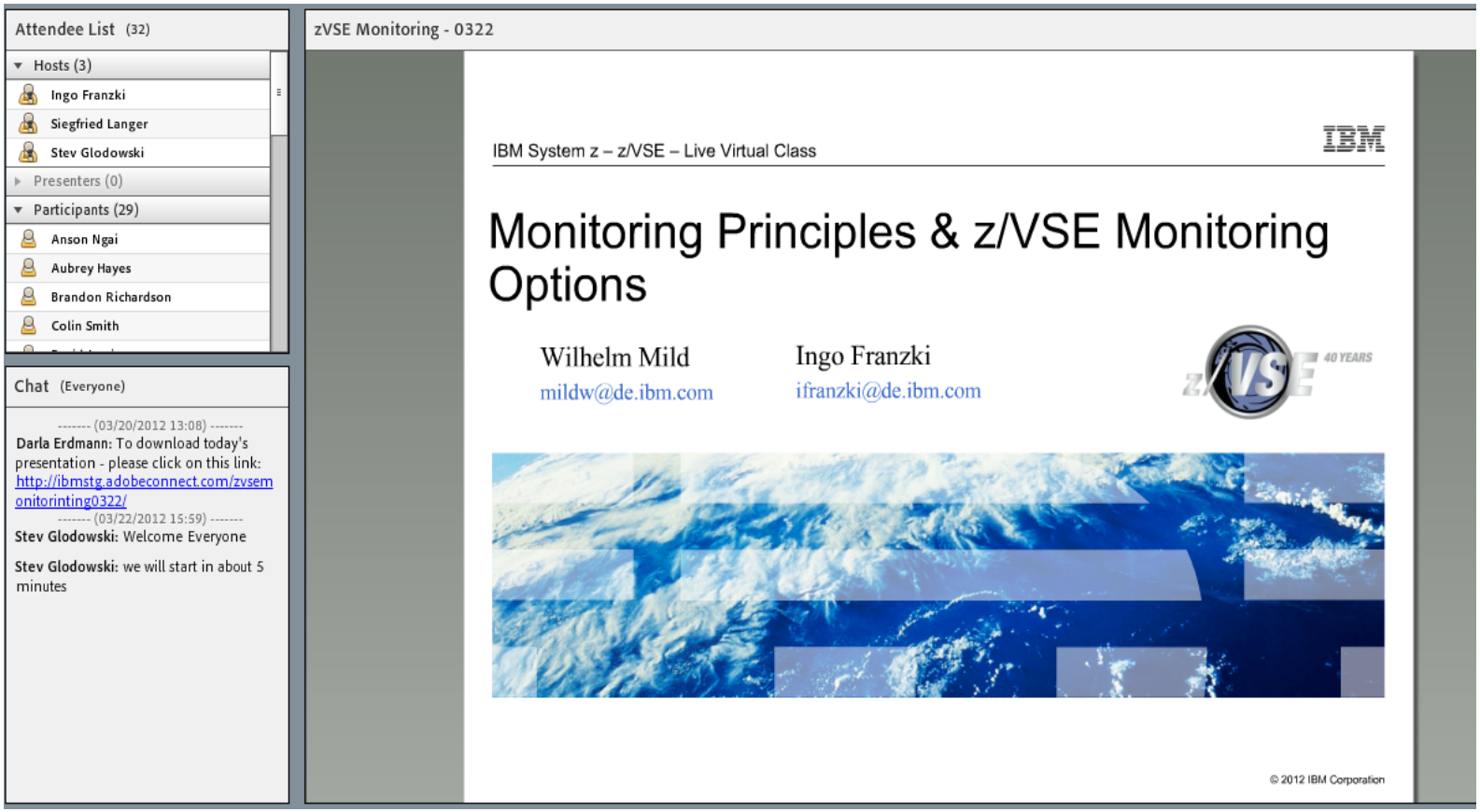
Hints and Tips for z/VSE 4.3

http://www.ibm.com/zvse/documentation/#hints

z/VSE Live Virtual Classes (LVC)

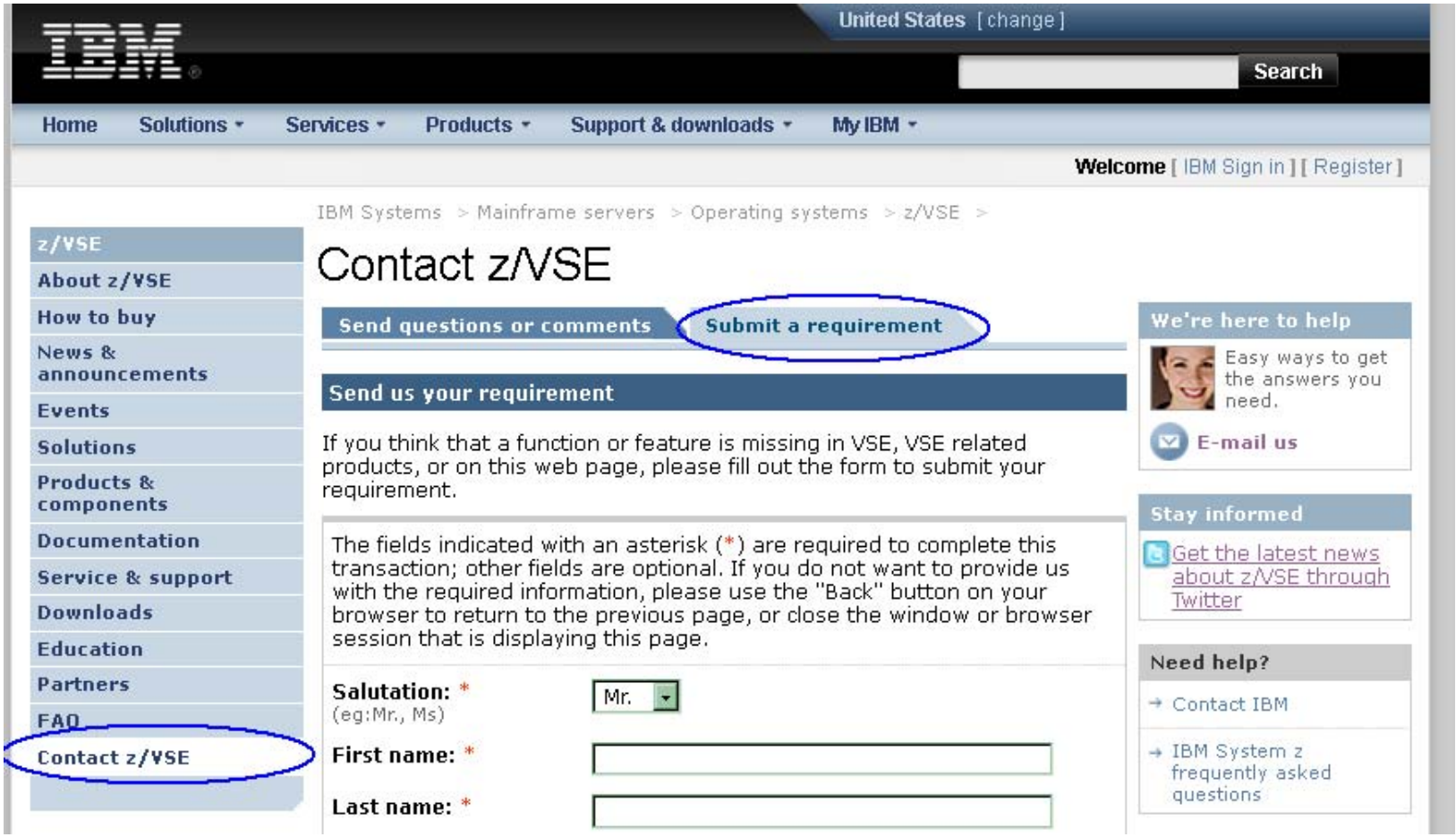
Join in on z/VSE Online Training

Follow IBMzVSE Twitter account and join the LVC mailing list



z/VSE LVCs	**http://www.ibm.com/zvse/education/#completed**
z/VSE & Linux LVCs	**http://www.vm.ibm.com/education/lvc/**

z/VSE Requirements



Thank You

Please forward your questions or remarks to

mailto:sedov-alexei@ru.ibm.com
mailto:stev.glodowski@de.ibm.com