



Session B72

VSE/VSAM Basics

Tom Grossheider

zSeries® EXPO

**FEATURING Z/OS, Z/VM, Z/VSE
AND LINUX ON ZSERIES**

September 19 - 23, 2005

San Francisco, CA

VSE/VSAM Basics

Session B72

- VSE/VSAM Components Page 3
- VSE/VSAM Data Organization Page 7
- VSE/VSAM Catalogs..... Page 33
- Backup / Recovery Page 41

VSE/VSAM Components

File Management History:

- ❖ Operating System:
 - Channel Command Word (CCW), Channel Address Word (CAW), Channel Status Word (CSW), SIO or SSCH

- ❖ PIOCS (Physical Input / Output Control System) (Used by VSE/VSAM)
 - Logical Unit, Command Control Block (CCB), CCW, SVC 0 (EXCP or Start I/O), SVC 7 (Wait on I/O)

- ❖ LIOCS (Logical Input / Output Control System)
 - DTFxx, OPEN, CLOSE, GET, PUT, POINT, DELETE
 - VTOC (Volume Table of Contents)
 - Label Cylinder

VSE/VSAM Components

- ❖ Virtual Storage Access Method (VSE/VSAM)
 - Two versions: MVS/ESA | S/390 | z/OS and VSE/ESA. VM/ESA | z/VM uses VSE/VSAM.
 - Dasd Storage Control
 - Catalog Management
 - Utilities
 - Emulation for SAM and ISAM

- ❖ Hierarchical Database (DL/I, SQL | DB2)
 - Use VSE/VSAM for File Control

- ❖ High Level Languages running under the Language Environment (LE/VSE)
 - Use LIOCS for sequential I/O (Tape, sequential disk, Card and Console)
 - Use VSE/VSAM calls

- ❖ e-Business Connectors
 - Most use VSE/VSAM for File Control on VSE/ESA

VSE/VSAM Components

- ❖ Catalog management
 - Maintains attributes of all files (clusters) defined to VSE/VSAM
 - Allocates dasd space.

- ❖ Open / Close
 - Connects and Disconnects a cluster with an application program.
 - Ensures access integrity

- ❖ Record Management
 - Performs all I/O access to clusters and catalogs
 - Manages Buffer Pools
 - Ensures SHR(4) integrity

- ❖ Space Management Feature
 - Emulates Sequential Access Method (SAM) for non-VSAM applications.
 - Open / Close activity performed by VSE/VSAM.
I/O performed by special VSE Basic Access Method (BAM) phase.
(\$IJGXSrv and its kith and kin)

VSE/VSAM Components

Utilities:

- IDCAMS
 - Catalog Access
 - Data manipulation (REPRO, PRINT)

- Backup / Restore
 - Archival data storage

- IKQVDU
 - VTOC maintenance

- IKQVEDA
 - Trace facility

- IKQVCHK
 - Catalog Corruption Checker

- IKQPRED
 - Compression prediction

- IDCONS
 - Interactive batch interface to IDCAMS written in REXX/VSE.

VSE/VSAM Data Organization

❖ ESDS (Entry-Sequenced Data Set)

- Sequential (Browse) Access
- Direct Access by Relative Byte Address (RBA)
 - Returned in RPL after GET request.
- Insert only at end-of-file.
- Record update only allowed if record length does not change.
- SAM ESDS is a unique sub-set

❖ KSDS (Key-Sequenced Data Set)

- Access by Sequential Browse, RBA, or Key
- Insert in key sequence.
- Contains data and index components.
- “Gobi desert” problem” (See page 21)

❖ RRDS (Relative-Record Data Set)

- Similar to KSDS. No Index component. Records retrieved using Relative Record Number (RRN) as key.
- Access by Browse, RBA, or RRN (key)
- Fixed length records only
- Insert / Update in RRN sequence. (Cannot change length of record)

VSE/VSAM Data Organization

- ❖ VRDS (Variable-Length Relative-Record Data Set)
 - Application access identical to RRDS. Uses RRN as key to access / insert / update records.
 - Allows variable length records.
 - Contains data and index components.

- ❖ Alternate Index
 - KSDS file.
 - Linked via Path to Base Cluster, which may be KSDS (keys) or ESDS (RBA)
 - See write-up on page **Error! Bookmark not defined..**

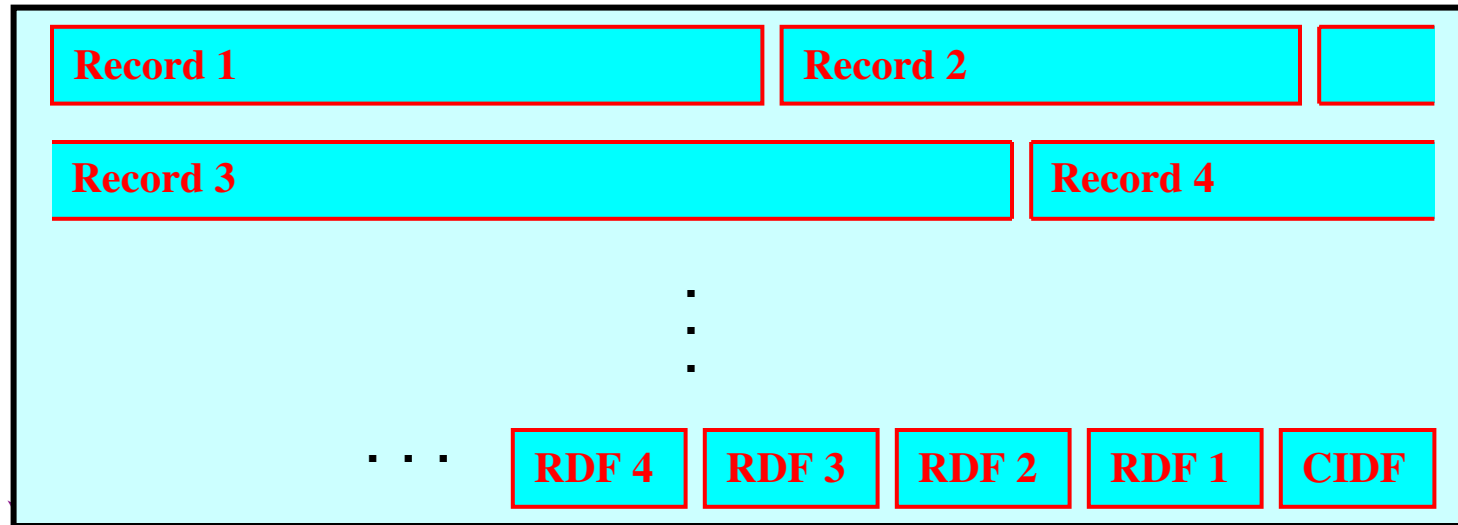
- ❖ Key Ranges
 - Archaic method to isolate I/O requests to different parts of a cluster (file).

- ❖ Cluster Definition:
 - Cluster Name
 - Data Name
 - Index Name (if KSDS or VRDS)

VSE/VSAM Data Organization

Control Interval format

- Logical Records can be:
 - Fixed (Average record size = maximum record size)
 - Variable (Average record size < maximum record size)
 - Spanned (larger than control interval)
 - Compressed (See page 12)



- **Record Definition Field (RDF):** If records have different lengths, one (three bytes) per record. If a series of records have the same length, two RDFs per sequence (nn records of mm length). Spanned record has two RDFs per CI. Identifies position of this segment in logical record.
- **Index:** One index record per CA, with a pointer for each control interval, containing the highest index value in CI.

VSE/VSAM Data Organization

Data Control Interval with 25 fixed 80-byte records:

```

000000 F0F0F1F0 F0F1F4F5 F2F7F0F1 F9F4F7F0 F2F2F2F2 F4F3F6F2 F8F5F2F1 F7F24040 *001001452701947022224362852172 *
000020 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
000040 40404040 40404040 40404040 40404040 F0F0F1F0 F6F9F1F4 F4F6F2F2 F7F5F2F0 * 0010691446227520*
000060 F1F1F4F3 F1F9F6F4 F9F7F9F9 F7F54040 40404040 40404040 40404040 *11431964979975 *
000080 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0000A0 F0F0F1F0 F7F1F3F3 F3F8F9F6 F9F4F4F4 F6F2F5F7 F0F8F0F2 F0F7F6F3 F8F64040 *001071333896944462570802076386 *
0000C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0000E0 40404040 40404040 40404040 40404040 F0F0F1F0 F8F9F6F1 F8F7F6F5 F3F1F5F2 * 0010896187653152*
000100 F6F1F4F9 F0F9F1F2 F2F9F0F6 F3F24040 40404040 40404040 40404040 *61490912290632 *
000120 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
000140 F0F0F1F0 F9F1F6F7 F7F7F8F1 F7F3F9F3 F2F3F9F1 F6F2F8F7 F1F0F0F6 F8F54040 *001091677781739323916287100685 *
000160 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
000180 40404040 40404040 40404040 40404040 F0F0F1F1 F0F5F5F3 F3F3F5F9 F3F8F4F6 * 0011055333593846*
0001A0 F0F7F7F5 F2F4F3F9 F4F7F9F8 F3F84040 40404040 40404040 40404040 *07752439479838 *
0001C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
      .
      .
      .
0007E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000800 to 000FDF same as previous line
000FE0 00000000 00000000 00000000 00000000 00000000 00000800 19 400050 07D00826 * & *
  
```

'08' = Left-hand RDF
'0019' = 25 Records

'40' = Right hand RDF
'0050' = 80-byte Records

'07D0' = Displ to Free Space
'0826' = Length of Free Space

VSE/VSAM Data Organization

Data Control Interval with variable length records.

```

000000 F0F0F1F0 F0F1F4F5 F2F7F0F1 F9F4F7F0 F2F2F2F2 F4F3F6F2 F8F5F2F1 F7F24040 *001001452701947022224362852172 *
000020 40404040 40404040 40404040 40404040 40404040 4040F0F0 F1F0F6F9 F1F4F4F6 * 0010691446*
000040 F2F2F7F5 F2F0F1F1 F4F3F1F9 F6F4F9F7 F9F9F7F5 40404040 40404040 40404040 *22752011431964979975 *
000060 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * *
000080 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * *
0000A0 40404040 40404040 40404040 40404040 0404040 40404040 4040C1E2 C4C6C1E2 * ASDFAS*
0000C0 C4C6C1E2 C4C6C1E2 C4C60000 00000000 00000002 27520F00 00000000 00000000 *DFASDFASDF *
0000E0 0003E840 40404040 40404040 40404040 40404040 4040F0F0 F1F0F7F1 F3F3F3F8 * Y 0010713338*
000100 F9F6F9F4 F4F4F6F2 F5F7F0F8 F0F2F0F7 F6F3F8F6 40404040 40404040 40404040 *96944462570802076386 *
000120 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * *
000140 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * *
000160 40404040 40404040 0404040 40404040 40404040 40404040 4040C1E2 C4C6C1E2 * ASDFAS*
000180 C4C6C1E2 C4C6C1E2 C4C60000 00000000 00000009 69444F00 00000000 00000000 *DFASDFASDF | *
0001A0 0003E840 40404040 40404040 40404040 40404040 40404040 4000E6C5 C9C7C5D3 * Y WEIGEL*
    .
    .
    .
000F60 00000000 00000000 21177F00 00000000 00000000 0003E840 40404040 40404040 * " Y *
000F80 40404000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * *
000FA0 00000000 00000000 0000B900 00280000 5C00004D 00006E00 00950000 4900004F * * ( > n | *
000FC0 00005A00 00AF0000 C900004B 00007C00 00DE0000 5000002D 0000F000 00DE0000 * ! I . @ & 0 *
000FE0 9D00009B 00008E00 00F50000 B60000F1 00006000 00E40000 C0 000036 0F830025 * 5 1 - U c *
  
```

'00' = Single RDF
'00C0' = 192 byte record

'00' = Single RDF
'0036' = 54-byte Record

'0F83' = Displ to Free Space
'0025' = Len of Free Space

VSE/VSAM Data Organization

This page
left
intentionally
blank

VSE/VSAM Data Organization

Compression

- **Hardware or Software**
- **Dictionary:**
- **Compression Control Dataset (CCDS)**
- **Cluster defined using “COMPRESSED” Attribute.**
- **Advantages:**
 - More data stored on dasd extent. Avoid 4 Giga-byte limit.
 - For sequential access, more records per buffer (CI), so fewer I/Os.
 - Some customers report substantial reductions in batch window.

| | | | |
|-------------|---------------------|------------|----------------------------------|
| Flag | Prior to Key | Key | Available for Compression |
|-------------|---------------------|------------|----------------------------------|

- At least 40 bytes per record must be available for compression.
- Requires up to 1Meg additional 31-bit GETVIS per file for compression services.

VSE/VSAM Data Organization

Speaker Notes:

(See “*VSE/VSAM User’s Guide and Application Programming*” (SC33-6632) under “*Chapter 5. Working with Compressed Files*”

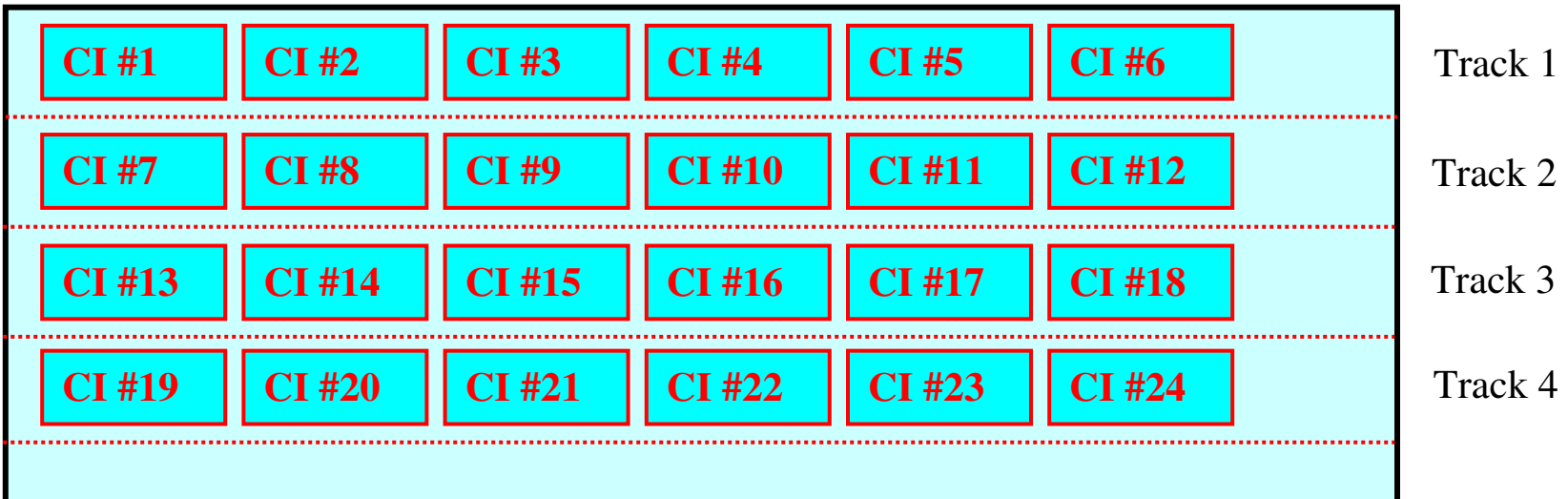
- **Hardware or Software:** Uses either hardware (CMPSC instruction) or software. Software is five times slower.
- **Dictionary:** Utilizes a dictionary of possible data combinations. Optimal dictionary combinations chosen during initial file load using a sampling process (about 2-3 tracks worth of data). Sufficient data must be loaded during initial file open to complete sampling.
- **Compression Control Dataset (CCDS):** Selected dictionary token identifiers are stored in CCDS. This is a separate KSDS file pre-defined in all VSE/ESA distributed catalogs, and added to new user catalogs defined via the Interactive Interface (II).

| | | | |
|-------------|---------------------|------------|----------------------------------|
| Flag | Prior to Key | Key | Available for Compression |
|-------------|---------------------|------------|----------------------------------|

Flag: Byte 1 = x'40',
Bytes 2-3 = uncompressed length of record (non-spanned records)
Bytes 2-5 = uncompressed length of record (spanned records)

VSE/VSAM Data Organization

Control Area format



- IMBED'ed Index
- SAM ESDS
- CISIZE vs track utilization

| 3390: | <u>CI Size</u> | <u>CIs per Track</u> | <u>Track Capacity</u> |
|-------|----------------|----------------------|-----------------------|
| | 2K | 21 | 42K |
| | 4K | 12 | 48K |
| | 8K | 6 | 48K |
| | 18K | 3 | 54K |

VSE/VSAM Data Organization

Track 4

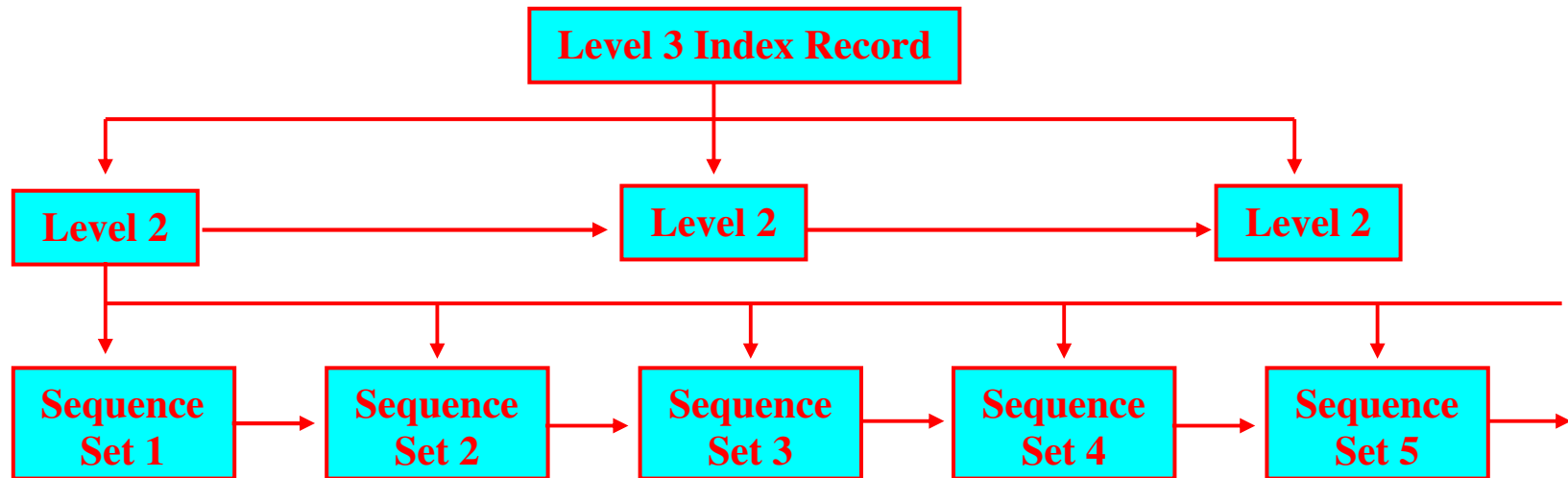
Speaker Notes:

- **IMBED'ed Index:** The first track of each CA is reserved for the sequence set record. If **REPLICATE**'d, the track is filled with instances of this index record.
- **SAM ESDS:** non-CA format. CI can be split at end of track.
- **CISIZE vs track utilization:** (see "VSE/VSAM User's Guide ..." (SC33-6632) under "*Performance: CI Size*")

VSE/VSAM Data Organization

KSDS / VRDS Index Record Structure

Typical Tree Structure, with horizontal chain:



Each Index Sequence Set describes a data control interval.

VSE/VSAM Data Organization

Key Compression in Index Records

- Both front and rear compression
- Very efficient. 100,000 records in 750 tracks required 4 tracks index.

| | <u>Highest Key in Control Interval</u> | | <u>F</u> | <u>L</u> | <u>CI</u> |
|--------|--|----------|----------|----------|-----------|
| CI#1: | 001305263036769318131188297363 | '001305' | 00 | 06 | 00 |
| CI#2: | 001562894381711315138840480100 | '562' | 03 | 03 | 01 |
| CI#3: | 001760223013561240734555111685 | '76' | 03 | 02 | 02 |
| CI#4: | 001949473536319918934062070610 | '94' | 03 | 02 | 03 |
| CI#5: | 002124234113651725528014615241 | '212' | 02 | 03 | 04 |
| CI#6: | 002327971654539736328740665031 | '327' | 03 | 03 | 05 |
| CI#7: | 002601508617068350649999452349 | '60' | 03 | 02 | 06 |
| CI#8: | 002867546478217498748460933800 | '867' | 03 | 03 | 07 |
| CI#9: | 003074228334141857900076103930 | '30' | 02 | 02 | 08 |
| CI#10: | 003390321861192839726291987529 | '3903' | 03 | 04 | 09 |
| CI#11: | 003621387672634913128100175623 | '621' | 03 | 03 | 0A |
| CI#12: | 003841434775673551831484414832 | '841' | 03 | 03 | 0B |

“F” Front Compression
“L” Length

VSE/VSAM Data Organization

Sequence Set (Level 1 Index) record showing key compression:

| | | | | | | | | | | | | | | | | | | | | | |
|-------|----------|----------|----------|---------------------|-------------------------|-----------|---------------|-----------------|-------------|--------|------|--------|-----|-----|---|--|--|--|--|-------------|---|
| 00000 | 0FF90301 | 00000000 | 00001000 | 00000000 | 01000072 | 0DFE0EFA | B3B2B1B0 | AFAEADAC | * 9 | | | | | | | | | | | * | |
| 00020 | ABAAA9A8 | A7A6A5A4 | A3A2A1A0 | 9F9E9D9C | 9B9A9998 | 97969594 | 93929190 | 8F8E8D8C | * zyxwvuts | | | | | | | | | | | * rqpnmklj | |
| 00040 | 8B8A8988 | 87868584 | 83828180 | 7F7E7D7C | 7B7A7978 | 77767574 | 73727170 | 6F6E6D6C | * ihgfedcba | "='@#: | | | | | | | | | | * ?>_%* | |
| 00060 | 6B6A6968 | 67666564 | 63626160 | 5F5E5D5C | 5B5A0000 | 00000000 | 00000000 | 00000000 | *, | | | | | | | | | | | * /-^;)*\$! | |
| 00080 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | * | | | | | | | | | | | * | |
| | ▪ | | | | | | | | | | | | | | | | | | | | |
| | ▪ | | | | | | | | | | | | | | | | | | | | |
| | ▪ | | | | | | | | | | | | | | | | | | | | |
| 00DE0 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 0000F2F3 | F7F70104 | * | | | | | | | | | | | * 2377 | |
| 00E00 | 59F4F1F8 | 030358F3 | F2F10203 | 57F9F703 | 0256F7F9 | F3030355 | F5F70302 | 54F3F3F2 | * 418 | 321 | 97 | 793 | 57 | 332 | | | | | | | * |
| 00E20 | 030353F1 | F5030252 | F2F0F1F5 | 020451F6 | 030150F4 | F5F20303 | 4FF1F1F2 | 02034EF7 | * 15 | 2015 | 6 | &452 | 112 | +7 | | | | | | | * |
| 00E40 | 03014DF6 | F103024C | F4F1F003 | 034BF2F1 | F203034A | F2F0F0F0 | 010449F7 | F9F10303 | * (61 | <410 | .212 | >2000 | 791 | | | | | | | | * |
| 00E60 | 48F5F203 | 0247F9F1 | F5020346 | F8F80302 | 45F6F103 | 0244F303 | 0143F8F0 | F1020342 | * 52 | 915 | 88 | 61 | 3 | 801 | | | | | | | * |
| 00E80 | F8F1F403 | 0341F5F2 | 030240F7 | F2F00203 | 3FF8F303 | 023EF5F1 | F503033D | F3F30302 | *814 | 52 | 720 | 83 | 515 | 33 | | | | | | | * |
| 00EA0 | 3CF6F1F1 | 02033BF8 | F3F50303 | 3AF6F103 | 0239F3F1 | 030238F5 | F0F10203 | 37F8F003 | * 611 | 835 | 61 | 31 | 501 | 80 | | | | | | | * |
| | ▪ | | | | | | | | | | | | | | | | | | | | |
| | ▪ | | | | | | | | | | | | | | | | | | | | |
| | ▪ | | | | | | | | | | | | | | | | | | | | |
| 00F80 | F2020314 | F8F30302 | 13F5F9F2 | 030312F3 | F2030211 | F5F0F402 | 0310F8F7 | 03020FF5 | *2 | 83 | 592 | 32 | 504 | 87 | 5 | | | | | | * |
| 00FA0 | F7F40303 | 0EF4F203 | 020DF4F1 | F5F00204 | 0CF8F4F1 | 03030BF6 | F2F10303 | 0AF3F9F0 | *74 | 42 | 4150 | 841 | 621 | 390 | | | | | | | * |
| 00FC0 | F3030409 | F3F00202 | 08F8F6F7 | 030307F6 | F0030206 | F3F2F703 | 0305F2F1 | F2020304 | *3 | 30 | 867 | 60 | 327 | 212 | | | | | | | * |
| 00FE0 | F9F40302 | 03F7F603 | 0202 | F5F6F2030301 | F0F0F1F3F0F50006 | 00 | 000FF9 | 0FF90000 | *94 | 76 | 562 | 001305 | 9 | 9 | | | | | | | * |

'562' = Concatenated Key
 03 = Front compression
 03 = Concatenated Key Len
 01 = Data Control Interval 2

'001305' = Concatenated Key
 00 = No front compression
 06 = Concatenated Key Len
 00 = Data Control Interval 1

RDF CIDF

VSE/VSAM Data Organization

“Gobi Desert” problem

- ❖ Can affect any KSDS file (including VSAM catalogs)
- ❖ Add at end, delete from beginning
- ❖ Index High Key not changed by delete
- ❖ Empty data CIs are never reused
- ❖ Impact:
 - Performance degradation
 - Cluster (catalog) growth
- ❖ Resolution:
 - Define keys (or cluster names) so that they are random
 - Frequent reorganization of file (or catalog).

VSE/VSAM Data Organization

Speaker Notes:

- ❖ **Add at end, delete from beginning.** This condition is caused by adding records (clusters, in the case of catalogs) at the end of a range of keys, and deleting records / clusters from the beginning.
- ❖ **Index High Key not changed by delete.** Once a data control area is allocated, an index sequence set record is assigned. This record contains the key of the highest record in each CI of the new CA. The high key recorded in the sequence set for a specific data CI is not changed if that record is deleted, or even if all data records are deleted from this data CI (or even CA).
- ❖ **Empty data CIs are never reused.** An empty data CI is only reused if a record is added in the range defined in the index as belonging to that CI. If new records (clusters) are added at the end of the extent, these empty data CIs are never reused.
- ❖ **Impact:**
 - **Performance degradation:** When accessing the first record in the file via browse. Since VSE/VSAM does not know that the data CI is empty, it must read every data CI in the front of the file, until it locates the first non-empty data CI, and then returns the first record.
 - **Cluster (catalog) growth:** The cluster or catalog will grow in size, even though the number of records (clusters) remains constant. Specifically in the case of a catalog, the high key range will eventually exceed 15 extents, preventing any additional catalog growth.
- ❖ **Resolution:**
 - **Define keys (or cluster names) so that they are random.** Consider beginning key or cluster name with hardware timestamp, bytes 4 & 5 (start counting with byte 0).

VSE/VSAM Data Organization

JCL to reorganize catalog to correct “Gobi desert” problem:

```
// JOB REPRO FILE TO DISK
// DLBL OFILE, 'REPRO.OUT', 0, SD
// EXTENT SYS005, SYSWK2, 1, 0, 1515, 75
// ASSGN SYS005, DISK, VOL=SYSWK2, SHR
// DLBL IFILE, 'VSESP.USER.CATALOG', , VSAM, CAT=VSESPUC
// EXEC IDCAMS, SIZE=AUTO
  REPRO INFILE (IFILE)           -
        OUTFILE (OFILE          -
              ENV ( BLOCKSIZE(18432) -
                    RECFM(V)))
IF LASTCC = 0 THEN             -
  DO
  REPRO INFILE (OFILE          -
        ENV ( BLOCKSIZE(18432) -
              RECFM(V)))       -
        OUTFILE (IFILE)
  END
/*
/ &
```

VSE/VSAM Data Organization

Size Limits: (rc x'1C' "No more available extents")

- ❖ **123 volumes per cluster component (data and index).**
- ❖ **16 volumes from default model:**
- ❖ **4.3 Giga-byte**
 - **4 byte RBA (Relative Byte Address):**
 - **5800 cylinders on D/T3390**
 - **Compression can help**
 - **'ExtraLargeDataset' or 'XXL'**
- ❖ **Clusters are limited to 123 extents.**
 - This is normally only a problem if you specify a very small secondary extent.
 - VSE/VSAM will sub-allocate an extent up to 5 times.
- ❖ **Catalogs, reusable files, and unique files are limited to 16 extents.**

VSE/VSAM Data Organization

Size Limits: (cont)

- ❖ **No single allocation over 16Meg records:**
- ❖ **64K control areas limitation for SHR(4)**
 - **Maximum SHR(4) file size is 51 GB**
- ❖ **16 million records per extent:** When defining files using “**RECORDS**”, you can specify up to 16 million records.
 - **“RECORDSIZE”:** You can request more data by specifying a larger maximum “**RECORDSIZE**” for the file.
 - **Compression can help:** Remember, if the file is compressed, VSAM uses the un-compressed maximum record length to calculate how much space to reserve for the file. This may give you more space than you actually need.

VSE/VSAM Data Organization

Speaker Notes:

- ❖ **16 volumes from default model:** For implicitly defined files, VSE/VSAM will extract up to 16 volumes from the default model
- ❖ **4.3 Giga-byte**
 - **4 byte RBA (Relative Byte Address):** Every record in a VSAM file is ultimately accessed using its relative byte position from the beginning of the file. This is stored in a 4-byte field, which means that you can only have 4.3 billion (4,294,967,295) bytes of data in a single VSAM file. The actual limit is 1 control area less than 4GB.
 - **5800 cylinders on D/T3390.** Don't forget that when the allocation goes to a new volume, a primary allocation is taken first, then secondaries.
 - **Compression can help:** increases the number of data records per control interval, so also increases the amount of user data that can be "packed" into 4GB. See page 12.
 - **'ExtraLargeDataset' or 'XXL':** A cluster can be defined as 'XXL', which allows up to 493GB for an extended KSDS or ESDS dataset. See "VSE/VSAM Commands" (SC33-6731-00) under "*Define Cluster*" / "*Extralargedataset*"
- ❖ **No single allocation over 16Meg records:** Any single allocation (primary or secondary) cannot contain more than 16Meg records, if the user intends on using IDCAMS Backup.
- ❖ **64K control areas limitation for SHR(4):** The lock used by Shareoption(4) processing has a two byte field for the control area id (or index control interval id for KSDS files). Thus, a Shareoption(4) file may not have more than 64511 (64K-1024) control areas, and the maximum file size would be 64511 times your control area size.
 - **Maximum SHR(4) file size is 51 GB:** Assuming a CFSIZE of 18K on 3390, and 123 extents of CYL(525,525) resulting in 64511 one-cylinder CAs.

VSE/VSAM Data Organization

Large DASD Support in VSE/ESA 2.6

- ❖ Supports dasd with up to 10017 cylinders.
- ❖ Implementation is transparent to existing applications and JCL. Dasd is flagged in LISTCAT as “BIG-3390”.
- ❖ Automatic
- ❖ Allocations converted to **CYLINDERS**
- ❖ Minimum data CISIZE increased to 1024 (depending on key size)
- ❖ BUFSPACE parameter may be increased
- ❖ Note: **RECOVERABLE** catalogs are not supported on “large dasd”.

VSE/VSAM Data Organization

Speaker Notes:

- ❖ **Large DASD Support in VSE/ESA 2.6** See “[VSE/ESA 2.6 Release Guide](#)” (SC33-6718-02)
 - **Supports dasd up to 10017 cylinders:** Increased from 64K tracks. Supports larger devices such as the IBM 3390 Model 9
 - **Implementation is transparent to existing applications and JCL.**
 - **Automatic:** During Define Space, VSE/VSAM automatically detects a “large DASD”, and implements the required catalog changes.
 - **Minimum data CISIZE increased to 1024 (depending on key size)**
 - A key length between 7 and 35 bytes requires at least 1024 bytes CI size.
 - A key length between 36 and 55 bytes requires at least 2048 bytes CI size.
 - A key length greater than 55 bytes requires at least 4096 bytes CI size.
 - **BUFSPACE parameter may be increased**

Must be at least as large as two Data Control Intervals and one Index Control Interval.
IDCAMS RESTORE or IMPORT may fail with msgIDC31337I.
Intransit APAR corrects problem by automatically increasing BUFSPACE to accommodate cluster definition.
 - **Allocations converted to CYLINDERS:** In most cases, clusters defined on “large DASD” will have their allocations converted from **TRACKS** or **RECORDS** to multiples of **CYLINDERS**. Catalogs defined on “large DASD” also have their default allocations modified.
 - **DY46249** is latest maintenance level.

VSE/VSAM Data Organization

Recommendations:

- ❖ **Do not use IMBED or REPLICATE:** These options were originally designed for slower, smaller dasd, and are not applicable to today's systems. They waste one track per control area, and complicate catalog management due to duplicate index and data extents. This option has been removed from VSE/ESA 2.6, although downward compatibility has been maintained.
- ❖ Maximize size of Control Area
- ❖ Use reasonably large data Control Intervals
- ❖ Let index Control Interval size default.
- ❖ Compression will save I/Os, but will cost CPU
- ❖ Additional buffering will save I/Os
 - For sequential processing, use largest possible data CIs, and multiple data buffers.
 - For direct processing, use smallest possible index CIs, and multiple index buffers.

VSE/VSAM Data Organization

CI Split process:

1. Set Split-in-Progress bit in data CI to be split, and write it out.
 2. Move all records higher than record being inserted (sequential vs direct processing) into a new data CI, and write it out.
 3. Update index sequence set and write it out.
 4. Remove C-I-P bit from old CI, and write it out.
 5. If no room for another data CI in this CA, or if sequence set record is full, causes CA split.
- ❖ CI splits not very costly in terms of system overhead (four I/Os, a bit of CPU processing overhead)
 - ❖ Recommendation: Do not specify CI free space. Do not reorganize files just based on CI split numbers.
 - ❖ Between steps 2 and 4, duplicate records exist in the database. If an error occurs (split is interrupted) at this point, CI with s-i-p bit is already in database.
 - Next time this CI is read, in keyed update mode, the split will be completed.
 - If the access is not keyed and not update, rc x'00' with feedback x'1C' is returned.
 - If the access is not keyed, but get-for-update, rc x'08' with feedback x'9C' is returned.

VSE/VSAM Data Organization

CA Split process:

1. Set Split-in-Progress bit in index sequence set, and write it out.
2. Format a new CA at high-used RBA in current extent. This may involve 150 I/Os for 4K data CIs.
3. Read all CIs from old CA. Turn on Split-in-Progress bit and write them back out (up to 300 I/Os).
4. Read all CIs higher than record being inserted (sequential vs direct processing), turn off c-i-p bit
5. Write CIs to new new data CA. (up to another 300 I/Os).
6. Create new index sequence set and write it out.
7. Read and write higher-level index records (if required)
8. Read all CIs from old CA, turn off c-i-p bit. Clear, if required. Write back out to old CA (up to another 300 I/Os).
9. Update old index sequence set record, indicate free CIs, turn off c-i-p bit and write out.
10. If no room for another data CA in this extent, allocate a new extent. If no additional extents possible, or if catalog full, reject record update / insert request with feedback code x'1C'. If adding new extent would result in > 4Gig, set feedback code x'D8'

VSE/VSAM Data Organization

CA Splits Recommendations:

- ❖ CA splits are quite expensive (up to a thousand I/Os) when they occur, but do not substantially impact future processing.
- ❖ If inserts are heavily clustered, CA splits may be more efficient than CA Free Space.
 - Reorganization will consolidate the cluster, removing free space created by splits, which may have to be added back in.
- ❖ Consider preformatting file, when new record keys are predictable and are inserted in direct mode.
- ❖ Do not reorganize after a certain number of CA splits.
- ❖ Define CA free space (at least 20%) for on-line files. **FREESPACE(0,20)**
- ❖ Between steps 5 and 8, duplicate records exist in the database. If an error occurs (split is interrupted) at this point, CI with s-i-p bit is already in database.
 - Next time this CI is read, in keyed update mode, the split will be completed.
 - If the access is not keyed and not update, rc x'00' with feedback x'1C' is returned.
 - If the access is not keyed, but get-for-update, rc x'08' with feedback x'9C' is returned.

VSE/VSAM Catalogs

Master Catalog (IJSYSCT):

- ❖ One Master catalog per system. Defined during system installation, normally on DOSRES. Assigned (via DEF SYSCAT) during IPL.
- ❖ VSE Messages Online File,
- ❖ Definitions for further user catalogs,
- ❖ Definition for VSE/VSAM –managed libraries (PRD1, PRD2)

User Catalogs:

- ❖ Optional. As many as required
- ❖ Requires JCL specification
- ❖ May be shared by multiple VSE/ESA systems
- ❖ Only one catalog per volume
- ❖ Can own space on multiple volumes.
- ❖ Multiple catalogs can own space on one volume (not recommended, except for DOSRES and SYSWK1)

VSE/VSAM Catalogs

VSESPUC User Catalog (VSESP.USER.CATALOG):

- On-line System Files:
 - VSE.CONTROL.FILE (IESCNTL)
 - CICS Start-up Dataset (CSD)
 - Restart Dataset (RSD)
 - Global Catalog (GCD, CICS TS)
 - Local Catalog (LCD, CICS TS)
 - Transient data, Intra-partition dataset (TD.INTRA)
 - Temporary Storage (DFHTEMP)
 - Data Management Facility (DMF) file
 - Transaction Abend Dump Library (DFHDMPA / DFHDMPB)
 - On-line Problem Determination File (IESPRB)
 - VSE Primary Library (Alternate ICCF Library)
- System Work Files
- PTF.FILE (Used to apply PTFs from disk)
- Text Repository File (IESTRFL)
- On-line Messages File (IESMSGs)
- VSE/VSAM Record Mapping Definitions (See e-business connectors)
- CICS REXX files (RFSDIR1, RFSPOL1, RFSDIR2, RFSPOL2)
- CICS Listener (EZACONF, EZACACH)

VSE/VSAM Catalogs

Catalog contents:

- ❖ Self-describing records (including cluster definitions for catalog itself)
- ❖ Volume (space) definitions
- ❖ Cluster definitions (including data, index, aix, and path)
- ❖ Recoverability definitions (not recommended, see below)
- ❖ Compression information (CCDS dataset)
- ❖ Internal Format like VSAM KSDS key range file
- ❖ Can be shared with VM/ESA | zVM in read-only mode.
Not shareable with MVS | OS/390 | z/OS.
- ❖ Three parts:
 - True-name (High-key) range. Contains index of 44-character names to internal catalog CI#
Subject to “Gobi Desert” problem (see description on page 21).
 - Low-key range:
 - Volume descriptors
 - Cluster descriptors
 - Index (Used only for True-name records)

VSE/VSAM Catalogs

Catalog Recommendations:

- ❖ NOIMBED, NOTRECOVERABLE
- ❖ Name clusters (and catalogs, where applicable) to include application names.
- ❖ Name all cluster components (spec. data and index) explicitly. Volume list in LISTCAT includes data and index, not cluster name.
- ❖ Exploit partition and system independent naming (% or %%) (SAM ESDS files only)
- ❖ With RAMAC virtual arrays, consider multiple volumes and catalogs, perhaps by application
- ❖ Place static (once defined, multi-access) and dynamic files (frequently redefined) in separate catalogs.
- ❖ Place batch vs on-line files in separate catalogs.
- ❖ Do not put all your eggs in one basket.
- ❖ Define catalog with “DEDICATE” at cluster level, and catalog allocation at “DATA” level.

VSE/VSAM Catalogs

Catalog Recommendations: (cont)

- ❖ Catalogs are limited to 16 extents, and can only expand on original volume.
 - Monitor this closely by checking self-descriptor cluster listing.
See “VSE/VSAM Commands” (SV33-6631) under “*Appendix A. Interpreting LISTCAT ... Output*”
 - When restoring an entire catalog, do not set the catalog volume as first volume in list.
 - If catalog fills up, re-define with additional allocation at “**DATA**” component level.

- ❖ Allocating space to true name range
 - More space will be allocated to the true-name range than the low key range.
 - If the true-name range fills up, and a new allocation is acquired, the entire new allocation is assigned to the true-name range.
 - Control Area will always be two tracks
 - IDCAMS always adjusts your allocation as it sees fit.
 - Index Allocation

VSE/VSAM Catalogs

Speaker Notes:

- ❖ Catalogs are limited to 16 extents, and can only expand on original volume.
 - When restoring an entire catalog, do not set the catalog volume as first volume in list. If you are restoring a series of files, all with the same volume list, and if the first volume is the catalog volume, this volume will fill up first, not leaving room for catalog expansion. Remember, the catalog cannot expand to a second volume.
- ❖ Allocating space to true name range: Normally, it is the true-name (high-key) range of the catalog which fills up first (Check for “Gobi Desert” problem, on page 21). You cannot explicitly define the initial space allocation for a specific “Key Range”. IDCAMS will allocate it according to a secret, hidden, arcane algorithm:
 - Control Area will always be two tracks, regardless of the allocation, If the catalog is defined as “IMBED”, the Control Area is three tracks, but one track per CA is reserved for the sequence set index record.
 - IDCAMS always adjusts your allocation as it sees fit. For example:

| <u>You define</u> | <u>IDCAMS defines</u> | <u>True-name range</u> | <u>Low-key range</u> |
|-------------------|-----------------------|------------------------|----------------------|
| Default | TRACKS(4,2) | 2 | 2 |
| TRACKS(75,15) | TRACKS(76,4) | 70 | 6 |
| CYL(100,20) | TRACKS(1444,300) | 1300 | 144 |

- Index Allocation: Normally, you don’t have to worry about the index allocation. IDCAMS will increase it to match the data allocation. For instance, 100 cylinders of catalog data, at 2 tracks per CA requires about 800 index records. Catalog index records are 1024 bytes long, and 33 fit per track, so 800 records would require about 25 tracks. IDCAMS defines 60.

VSE/VSAM Catalogs

Recoverable catalogs:

- ❖ Creates a copy of catalog data in a unique file (Catalog Recovery Area = CRA).
- ❖ Each volume has a CRA
- ❖ All data is written twice
- ❖ Does not protect against VSE/VSAM internal logic errors
- ❖ IDCAMS RESETCAT does not ensure multi-volume file data consistency
- ❖ RESETCAT will recover the latest version of the catalog from the CRA
- ❖ If catalog volume is lost
- ❖ If a non-catalog volume is lost:
- ❖ Not recommended

VSE/VSAM Catalogs

Speaker Notes:

- ❖ Each volume has a CRA: Reflects catalog data for clusters residing on that volume.
- ❖ All data is written twice: Once to normal catalog, and once to CRA. Doubles vulnerability window.
- ❖ Does not protect against VSE/VSAM internal logic errors: Since the same data is written to catalog and CRA.
- ❖ IDCAMS RESETCAT does not ensure multi-volume file data consistency: It copies catalog data from a specific CRA to catalog. No file data is reset, user is responsible to ensure that multi-volume file data is consistent.
- ❖ RESETCAT will recover the latest version of the catalog from the CRA: This is handy, if catalog data is overlaid (corrupted). Unfortunately, if the Catalog Recovery Area (CRA) is overlaid, there is no way to recover it, and you can't open a perfectly good catalog.
- ❖ If catalog volume is lost: If the volume containing the catalog is lost, define a new catalog (or restore an old version), and recover (using RESETCAT) the other volumes from the lost catalog. User is responsible to ensure compatibility of data on other volumes.
- ❖ If a non-catalog volume is lost: restore a backup copy and use RESETCAT to resynchronize catalog data. User is responsible to ensure compatibility of multi-volume files.
- ❖ **Not recommended.** Much more efficient and reliable to simply backup all volumes for a user catalog (using FASTCOPY) or all clusters in a catalog (using IDCAMS BACKUP).

