IBM GLOBAL
SERVICES

E49

Design and Tuning of VSE/VSAM

Charles E. Olsen

zSeries Expo | **Nov. 1 - 5, 2004**

Miami, FL

RETURN TO INDEX

Design and Tuning of VSE/VSAM
Session E56 / E57

VSE/ESA
Technical Conference
Miami, Florida
Oct 7th – 10th, 2002

Charles E. Olsen

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 53.

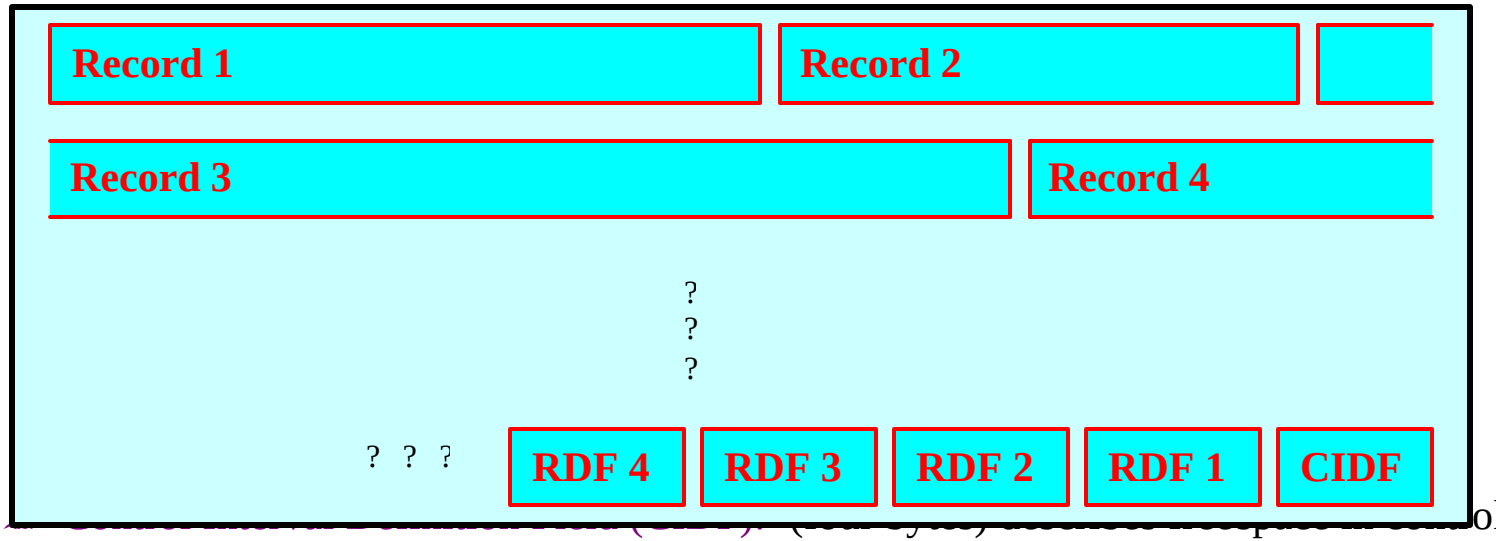
- Key Ranges
  - Archaic method to isolate I/O requests to different parts of a cluster (file).

- 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)



ol interval

- **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  *00100145270194702222436285217 2  *
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 40404040  *11431964979975                  *
000080  40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040  *                                *
0000A0  F0F0F1F0 F7F1F3F3 F3F8F9F6 F9F4F4F4  F6F2F5F7 F0F8F0F2 F0F7F6F3 F8F64040  *00107133389694446257080207638 6  *
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 40404040  *61490912290632                  *
000120  40404040 40404040 40404040 40404040  40404040 40404040 40404040 40404040  *                                *
000140  F0F0F1F0 F9F1F6F7 F7F7F8F1 F7F3F9F3  F2F3F9F1 F6F2F8F7 F1F0F0F6 F8F54040  *00109167778173932391628710068 5  *
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 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  *00100145270194702222436285217 2  *
000020  40404040 40404040 40404040 40404040  40404040 4040F0F0 F1F0F6F9 F1F4F4F6  *                      0010691446*
000040  F2F2F7F5 F2F0F1F1 F4F3F1F9 F6F4F9F7  F9F9F7F5 40404040 40404040 40404040  *2275201143196497997 5          *
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  *9694446257080207638 6          *
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:

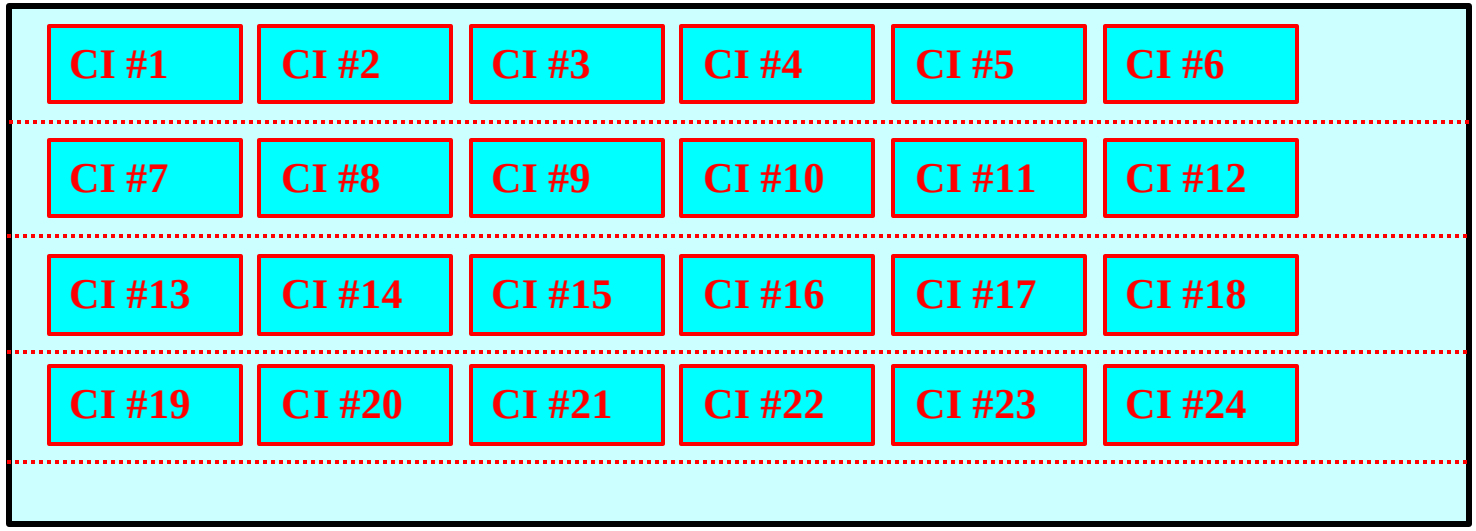
(See *"VSE/VSAM User's Guide and Application Programming"* (SC33-6632) under *"Chapter 5. Working with Compressed Files"*

- **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 date). 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: 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:

CI Size	CIs per Track	Track Capacity
2K	21	42K
4K	12	48K
8K	6	48K
18K	3	54K

VSE/VSAM Data Organization

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.

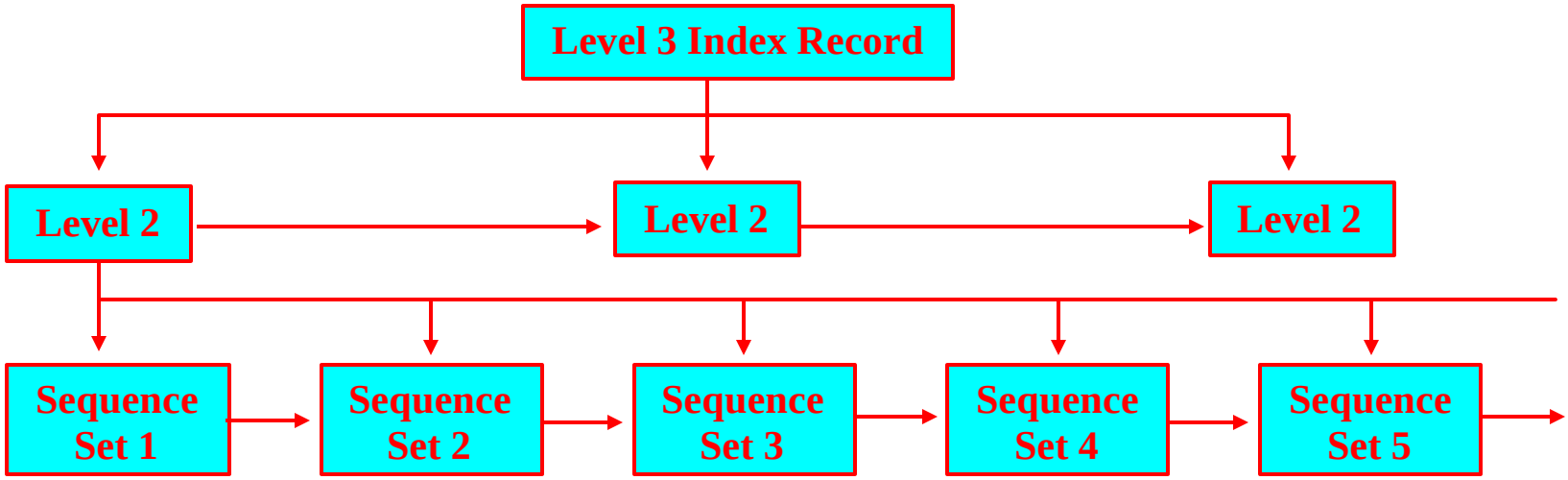
- **SAM ESDS**: non-CA format. CI can be split at end of track.

- **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:

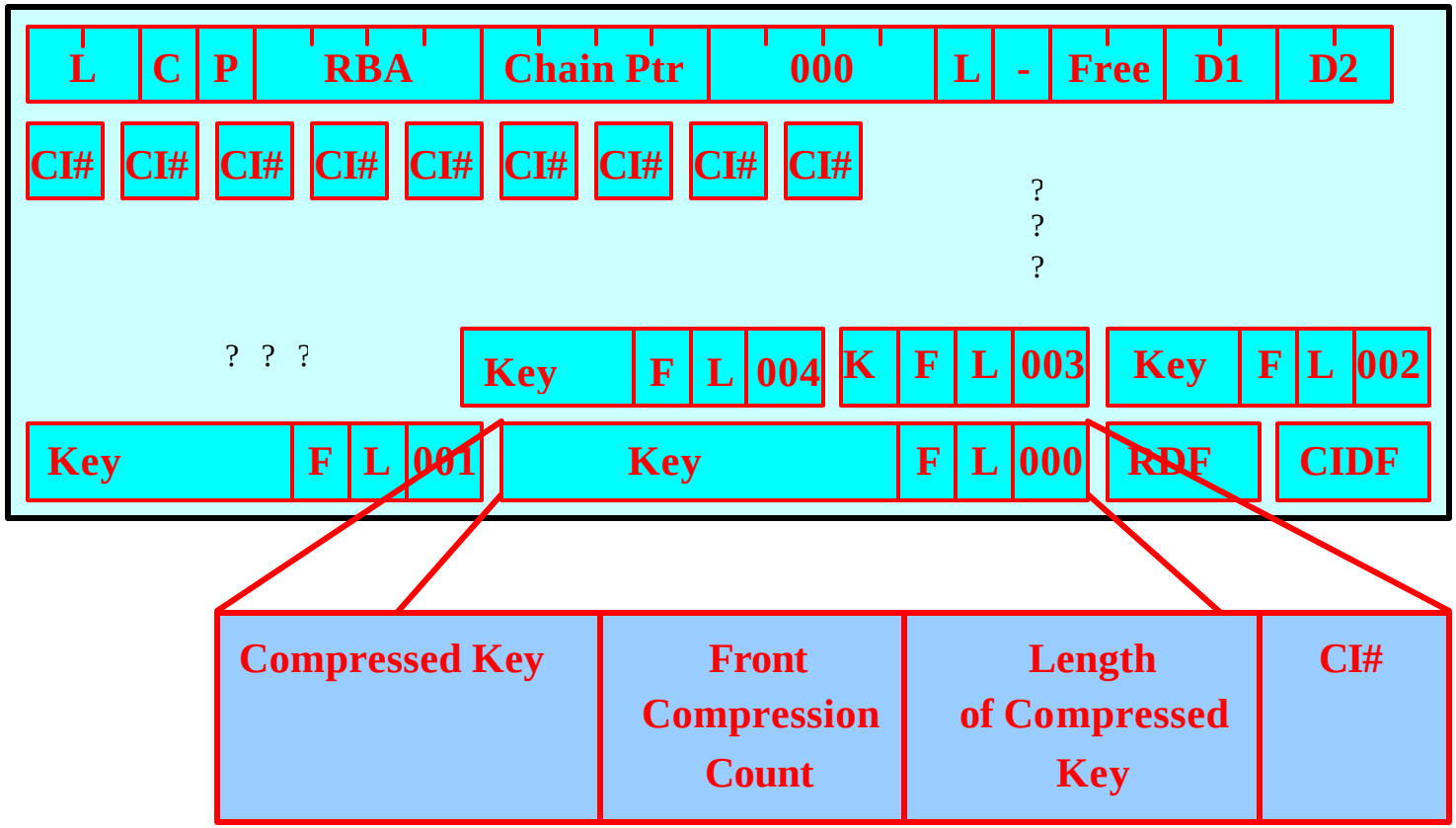


Each Index Sequence Set describes a data control interval.

VSE/VSAM Data Organization

Format of Sequence Set (Level 1 Index) CI

(See "*VSE/VSAM DRM*" (SC33-6321-00) under "*Chapter 4. Data Areas*" / "*Index*"



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.

	Highest Key in Control Interval		F	L	CI
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        rqponmlkj     *
00040  8B8A8988 87868584  83828180 7F7E7D7C  7B7A7978 77767574 73727170 6F6E6D6C  *  ihgfedcba "='@#:          ?>_%*
00060  6B6A6968 67666564  63626160 5F5E5D5C  5B5A0000 00000000 00000000 00000000  *,         /-ª;)*$!              *
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 **CISIZE** 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:

You define	IDCAMS defines	True-name range	Low-key range
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).

Backup / Recovery

Backup/Restore Options:

- IDCAMS Backup/Restore:
  Fairly quick. Allows restoration of individual clusters. No data reorganization

- IDCAMS REPRO:
  Slow. Use for compressed files. Reorganizes data.

- IDCAMS EXPORT / DISCONNECT:
  Slow. Compatible with MVS | OS/390 | z/OS.

- FASTCOPY
  Fast. Cannot restore individual clusters. No data reorganization. Must backup all volumes for catalog.

- IXFP / SnapShot or Flashcopy:
  Extremely Fast, Cannot restore individual clusters. No data reorganization. Must backup all volumes for catalog.

- IDCAMS SNAP:
  Extremely Fast, Allows restoration of individual clusters, by backing them up first, then restoring them. No data reorganization.

Backup / Recovery

IDCAMS BACKUP/RESTORE:

- Saves file contents, catalog definitions, and compression tokens
- Device-independent: Backup to either Tape or DASD.
- High-speed backup: faster than REPRO or EXPORT.
- Backup all files from catalog, or selectively via generic list
- Allows files to be selectively restored, or restored to a different catalog.
- Operates at CI-level
- Not compatible with MVS | OS/390 | Z/OS.

- **Compaction** ("**COMPACT**" option)
  - Software compaction of backup data via "**COMPACT**" option during Backup.
  - More efficient to use hardware
  - Do not use to backup compressed data

- **Performance**:
  - **"BLOCKSIZE(65535)"**
  - **"BUFFERS(4)" (**maximum 8)
  - "// **UPSI 1**" for PFIXing buffers sometimes helps.
  - Multiple concurrent backups very efficient.

Backup / Recovery

IDCAMS BACKUP/RESTORE: (cont)

- Return Code x'29': Warns of files open for update during backup.
  - Identifies potential integrity problems in backup copies.
  - Console message does not identify file. See SYSLST.
  - SHR(4) files use same lock for read and write open. Message says “might”.

- **Multiple Catalog Backup**
  - Added via “**NOREWIND**” parameter in VSE/ESA 2.3.
  - Valid only for Standard Labeled tapes.
  - One jobstep for each catalog backed up.
  - User Positioning Required
    - See next page for a circumvention

Backup / Recovery

Speaker Notes:

- Saves file contents, catalog definitions, and compression tokens. Allows restore without separate step to define cluster.
- Operates at CI-level. Cannot, therefore, be used for file re-organization. Will not detect file corruption. Use REPRO.

- **Compaction** ("**COMPACT**" option)
  - More efficient to use hardware (tape unit compaction = IDRC). Do not use "**COMPACT**" option on tapes with IDRC feature, or on compressed files.
  - Do not use to backup compressed data: Backup of critical compressed files should be done using IDCAMS REPRO, or another record-level backup program. If error in expansion, data cannot be used.

- **Performance**:
  - **"BUFFERS(4)" (**maximum 8). Additional buffers seldom improve performance, and may result in running off the end of the tape. All buffers are "strung together" in a single tape output CCW chain. When the reflective marker at physical end of tape is detected, an exception is posted to the control unit, but the write operation continues. Sufficient tape must be accessible AFTER the reflective marker, or the data is lost, and the I/O request fails.

- **Multiple Catalog Backup**

  See "*VSE/VSAM Commands*" (SC33-6731-00) under "Backup" or "Restore" / "Norewind"

  - User Positioning Required. Each backup jobstep creates multiple tape files. A cross-reference of backup files is only maintained within each jobstep. Therefore, restoring from a multiple catalog backup tape requires user-initiated positioning.

Restore Cluster from second catalog backed up on a tape

The first jobstep attempts to restore an unknown cluster, and fails, which leaves the tape positioned following the first catalog on the tape. The next jobstep works, because the tape is now positioned at the beginning of the second catalog. The first jobstep can be repeated as many times as required, once per catalog to be skipped.

```
// JOB RESTORE CATALOG
// ASSGN SYS004,181
// MTC REW,SYS004
// TLBL TAPE,'BACKUP.FILE'
// UPSI 1
// EXEC IDCAMS,SIZE=AUTO
    RESTORE OBJECTS(ASDF)    -
       BUFFERS(4)              -
       STDLABEL(TAPE)        -
       NOREWIND
/*
// TLBL TAPE,'BACKUP.FILE'
// DLBL IJSYSUC,'VSESP.USER.CATALOG',,VSAM
// UPSI 1
// EXEC IDCAMS,SIZE=AUTO
    RESTORE OBJECTS(DFHTEMP)  -
       BUFFERS(4)              -
       STDLABEL(TAPE)          -
       NOREWIND
/*
// MTC REW,SYS004
/&
```

Backup / Recovery

Error message on SYSLST is generated when file backup is complete. Buffering may result in the list on SYSLST being in a different order than the error message on the console.

Console:

```
F4 0004 4228I FILE VDWACB     OPEN  ERROR X'29'(041) CAT=IJSYSUC
   (OPNH1-14) WARNING: SHR2 FILE ALREADY OPENED FOR OUTPUT DURING BACKUP!

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'29'(041) CAT=IJSYSUC
   (OPNH1-14) WARNING: SHR2 FILE ALREADY OPENED FOR OUTPUT DURING BACKUP!

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'29'(041) CAT=IJSYSUC
   (OPNH1-14) WARNING: SHR2 FILE ALREADY OPENED FOR OUTPUT DURING BACKUP!

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'29'(041) CAT=IJSYSUC
   (OPNH1-25) WARNING: SHR4 FILE MIGHT BE OPENED FOR OUTPUT DURING BACKUP!

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'76'(118) CAT=IJSYSUC
   (OPND1-5 ) WARNING:FILE WAS NOT CLOSED ON A PREVIOUS OUTPUT-OPEN

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'29'(041) CAT=IJSYSUC
   (OPNH1-25) WARNING: SHR4 FILE MIGHT BE OPENED FOR OUTPUT DURING BACKUP!

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'76'(118) CAT=IJSYSUC
   (OPND1-5 ) WARNING:FILE WAS NOT CLOSED ON A PREVIOUS OUTPUT-OPEN

F4 0004 4228I FILE VDWACB     OPEN  ERROR X'29'(041) CAT=IJSYSUC
   (OPNH1-14) WARNING: SHR2 FILE ALREADY OPENED FOR OUTPUT DURING BACKUP!
```

SYSLST:

```
IDC11310I CICS.GCD  MIGHT BE INCONSISTENT IN BACKUP FILE
IDC11310I CICS.LCD  MIGHT BE INCONSISTENT IN BACKUP FILE
IDC11310I CICS.TD.INTRA  MIGHT BE INCONSISTENT IN BACKUP FILE
IDC11310I DFHTEMP  MIGHT BE INCONSISTENT IN BACKUP FILE
IDC11310I PC.HOST.TRANSFER.FILE  MIGHT BE INCONSISTENT IN BACKUP FILE
IDC11310I VSE.CONTROL.FILE  MIGHT BE INCONSISTENT IN BACKUP FILE
```

Backup / Recovery

IDCAMS REPRO:

- Saves only file contents in uncompressed (record) format.
- Restore requires separate cluster definition step.
- Output can be a sequential file (tape or disk), or another VSAM cluster
- Can be used to reorganize cluster
- All compressed files should be backed up in non-compressed format.
- Can also be used to reorganize catalog suffering from “gobi desert” problem. See description on page 21
- Not recommended for Catalog-only backup

IDCAMS EXPORT / IMPORT:

- Saves file contents in uncompressed (record) format or compressed CI format.
- Also saves catalog definition information
- “**SOURCEINHIBIT**”
- Can be used to migrate files to/from MVS | OS/390 | z/OS.
  - Use “**BLKSIZE(32767)**” for compatibility

Backup / Recovery

Speaker Notes:

IDCAMS REPRO:

- Can be used to reorganize cluster
  (places records, CIs and CAs in consecutive sequence, and reintroduces free space.)
- Not recommended to be used for Catalog-only backup
  Backing up catalogs without the application data is of limited usefulness, and may actually lead to corrupted data if restored in wrong sequence.

IDCAMS EXPORT / IMPORT:

- Also saves catalog definition information. Import automatically re-defines the target cluster.
- "**SOURCEINHIBIT**". Locks file after backup has been taken. (NOSOURCEINHIBIT is default).

Backup / Recovery

IXFP/SnapShot and Flashcopy Support by IDCAMS SNAP

- Adds VSE/VSAM access to IXFP/SnapShot and FlashCopy:
- Fast backup:
- Off-line backup:
- Significantly reduces the time when datasets may not be available to on-line processing.
- Synchronized backup:
- Duplicate volids:
- Steps:
  - **IDCAMS SNAP** calls SnapShot (VSE/ESA 2.5 or later) or FlashCopy (VSE/ESA 2.6 or later)
  - **IDCAMS IMPORT CONNECT**
  - **IDCAMS BACKUP SYNONYMLIST(..)**

Backup / Recovery

Speaker Notes:

See "VSE/ESA 2.5 Release Guide" (SC33-6718-01) and
http://www-1.ibm.com/servers/eserver/zseries/os/vse/pdf/vsenew20/vseesna.pdf

- **Adds VSE/VSAM access to IXFP/SnapShot and FlashCopy:** IXFP/SnapShot support for the IBM RAMAC Virtual Array (RVA) and FlashCopy for IBM Enterprise Storage Server (ESS), by themselves, only backup a complete volume, and cannot access individual VSE/VSAM clusters, or aggregate all volumes associated with a catalog.
- **Fast backup:** All volumes in a catalog backed up within seconds or minutes, instead of hours.
- **Off-line backup:** Once batch processing has resumed, a tape backup can be made from the "snapped" copy of the catalog
- **Synchronized backup:** Alleviates problem of related files in catalog being changed during backup.
- **Duplicate volids:** VSE/ESA has a restriction on duplicate volids. IDCAMS SNAP creates a synonym list, which is used together with IDCAMS BACKUP to access the catalog after it has been “snapped”.
- **Steps:**
  - **IDCAMS SNAP** calls SnapShot (VSE/ESA 2.5 or later) or FlashCopy (VSE/ESA 2.6 or later) to copy the catalog and creates a “synonym list” for the affected volumes.
  - **IDCAMS IMPORT CONNECT** creates a pointer to backed-up catalog under a pseudonym..
  - **IDCAMS BACKUP SYNONYMLIST(..)** uses the “synonym list” created earlier, to backup selected files, or the entire catalog.

Backup / Recovery

Migration:

- Backup files using IDCAMS BACKUP
- Make a list of all volumes (and extents) managed by old catalog.
- Remove old user catalog from Master Catalog using **IDCAMS EXPORT DISCONNECT**
- Remove catalog extents from VTOC on primary and managed volumes.
  Use IKQVDU or VSE/DITTO.
- Define new catalog and space on managed volumes.
- Restore files from backup tapes.
- Do not copy catalogs using FASTCOPY, unless old and new volume absolutely identical.

Backup / Recovery

This page was deliberately left blank

Alternate Indexes

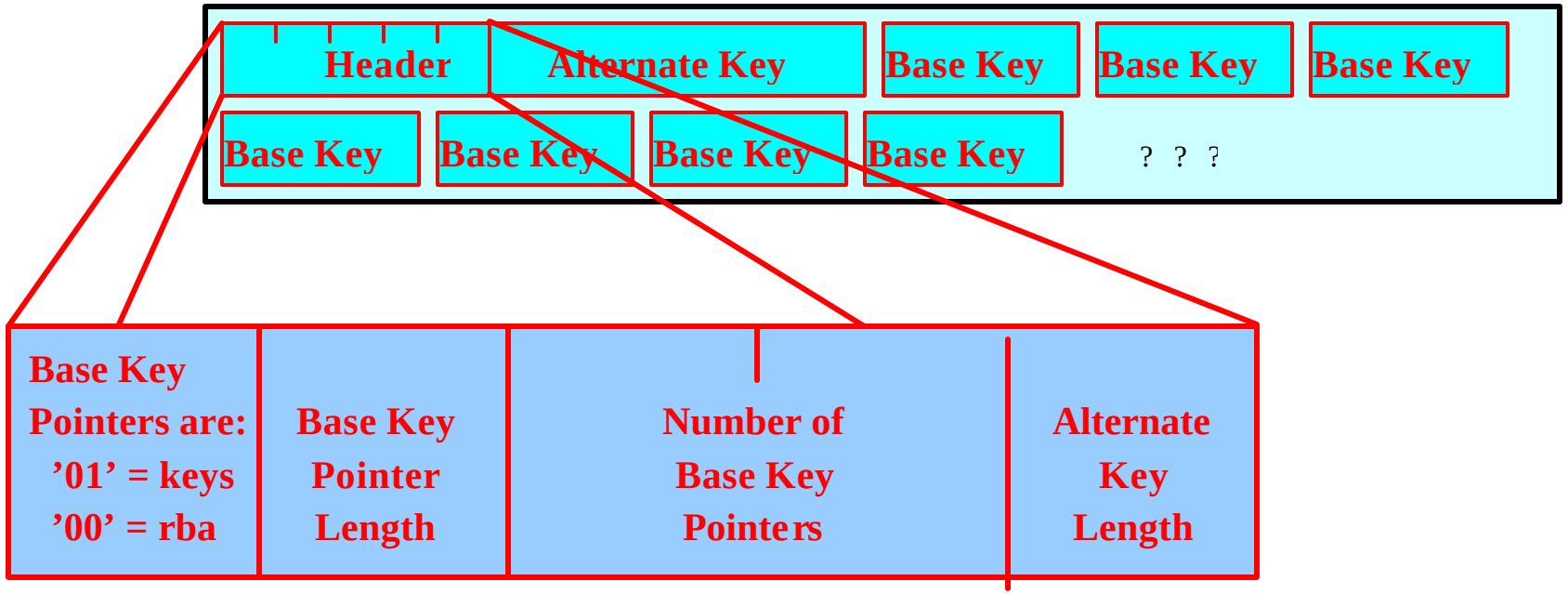
- Alternate way to access information in KSDS or ESDS file
- Path Name vs Alternate Index Name
  - Alternate Index is a KSDS file, Path is simply a logical connection between an alternate index and a base cluster.
- **UNIQUEKEY** vs **NONUNIQUEKEY**
  - Non-unique keys require definition of spanned records.
  - Logical Recordsize can be up to one CA in size (810K on 3390)
  - Major impact on 31-bit GETVIS.
- Feedback Code (Return Code):
  - x'08'(0) Duplicate aix key
  - x'28'(8) GETVIS error
  - x'94'(8) Max number of AIX pointers
  - x'64'(8) (OPEN) empty alternate index
  - x'6C'(8) Too many alternate index keys for record size
  - x'90'(8) Mismatch between AIX and Base

Alternate Indexes

“Dummy” aix keys:

- IDCAMS **BLDINDEX** will assign even “dummy” keys to alternate index (blanks, nulls, high values).
- This often causes unnecessarily large alternate index records. (See discussion of “**NONUNIQUE**” keys on previous page)
- Solution: Write your own build index routine.

Alternate Index Record Format:



SHAREOPTIONS and Security

SHR(1,x) One program opens file for output
-or-
Multiple programs open file for input

- VSE/VSAM guarantees read and write integrity

SHR(2,x) One program opens file for output
-and-
Multiple programs open file for input

- VSE/VSAM achieves fast response by keeping records in buffers and using read-ahead.
- VSE/VSAM guarantees write integrity, not read integrity

SHR(3,x) No VSE/VSAM control over access

- VSE/VSAM guarantees neither read nor write integrity
- Assumes program has own locking mechanism.
- Used for VSE/VSAM catalogs, VSE libraries
- Should not be used for "normal" VSAM files

SHAREOPTIONS and Security

SHR(4,x) Multiple programs can access cluster for either input or output, but only from a single VSE system.

SHR(4,4) Multiple programs from multiple VSE systems can access cluster for either input or output.

- VSAM guarantees write integrity, and some read integrity
- VSAM enhances read integrity by always reading a record from disk
- This additional operating system overhead can be costly
  SHR(4,4) is especially costly, because each lock must be reflected to dasd lock file.

In addition, VSE/VSAM controls access from different strings (programs) within the same VSE task (i.e. CICS) using The Hold Block (THB).

SHAREOPTIONS and Security

File Access Contention:

- SHR(1) or SHR(2): Second program receives x'A8' during OPEN
- SHR(4) Second program is allowed to open file. When a record update is requested:
  - VSE/VSAM will attempt to lock record
  - If not available, four possibilities:
    a. VSE sub-task will wait until record is available.

    Online:
    b. Return to CICS via Exit List
    c. Return code x'08..D0'
    d. Return code x'08..34'
  - Return code x'08..14'

SHAREOPTIONS and Security

Speaker Notes:

- SHR(1) or SHR(2): Second program receives x'A8' during OPEN

```
F2 057 4228I FILE IESPRB     OPEN  ERROR X'A8'(168) CAT=VSESPUC
(OCSHR--5) File already open in another partition
```

- **VSE/VSAM will attempt to lock record:** Data Control Area (for keyed access) or Index Control Interval (for RBA access).

Online: If application (usually CICS) passed an "Exit List" in ACB, VSE/VSAM will ask that VSE/ESA Lock Manager return with an ECB pointer, which will be posted when resource is available. On-line requests also flag RPL with "ASYNCHRONOUS" option.

b. **Return to CICS via Exit List:** If lock held by another VSE sub-task:, VSE/VSAM returns to caller (CICS), via the Exit List, to allow asynchronous processing.

c. **Return code x'08..D0':** If lock held by same VSE sub-task, VSE/VSAM returns to caller (CICS) with x'08..D0'. CICS is then responsible for queuing and retrying the request after the ECB is posted. If CICS TS returns to VSE/VSAM with an "RPL Message Area", VSE/VSAM will return a pointer to the string which is currently holding the record.

d. **Return code x'08..34':** If other string was not initiated by CICS, VSE/VSAM returns with x'08..34' (Logic Error).

- **Return code x'08..14':** If two transactions request a record in the same CI, the second transaction is refused with x'08..14'. CICS handles this just like '08..D0'. This can also occur with read access.

On-line Considerations

Local Shared Resource (LSR) Pools

- **Shared Pool Concept**
- **Automatic Definition**
- **Advantages:**
  - **Better Storage Management**
  - **Data In Memory**
- **Monitor LSR Pools**
- **Index Sub-pools**
- **Initial Allocation:**
  - **“Deal the files out”.**
  - **Keep aix and base clusters in same pool:** (CICS will warn you during start-up).
  - **Monitor statistics, and make required adjustments.**
- **Non-Shared Resources (NSR):**
  - **Read Ahead, Write Behind**
  - **Prevents “flooding” LSR pool**

On-line Considerations

Speaker Notes:

- **Shared Pool Concept:** Under CICS, if cluster is assigned an "**LSRPOOLID**" (CEDA) or "**LSRPOOL**" (**DFHFCT TYPE=FILE**) it will share a buffer pool with all other clusters assigned to the same pool. Sub-pools of various size are defined for each pool to match the CI Size of the clusters assigned to the pool. Buffers are reused on a "least recently used" algorithm.
- **Automatic Definition:** If Clusters are assigned to the pool, but no pool is defined, CICS will, at startup, scan the cluster definitions and automatically allocate a pool. CICS will define a percentage of resources defined for all the files in this LSR pool (strings and buffers per sub-pool), based on a percentage specified as "**SHARELIMIT**" (CEDA) or "**RSCLMT**" (**DFHFCT**). This can be a good way to attain a first approximation of the size of the pool, but costs time and machine resources during startup, so you are better off defining each LSR Pool explicitly ("`DFHFCT TYPE=SHRCTL`", or "`CEDA DEFINE LSRPOOL`").
- **Advantages:**
  - **Better Storage Management:** Sharing resources between multiple clusters provides more resources during periods of intense activity for a specific cluster, yet fewer total resources.
  - **Data In Memory:** With a large buffer pool, CIs are retained in memory once referenced, and save I/O if they are referenced again.
- **Monitor LSR Pools:** Migrating from D/T3380 to D/T3390 increases index CI size from 2K to 2.5K. This will move the Index CIs into the same LSR pool as 4K data CIs, resulting in reduced performance.
- **Index Sub-pools:** Starting with VSE/ESA 2.5, index buffers can be specified separate from data buffers in LSR pool.

On-line Considerations

```
CEDA  ALter Lsrpool( TESTLSR  )
   Lsrpool         : TESTLSR
   Group           : MINE
   DEscription  ==> POOL WITH SEPARATE INDEX BUFFERS
   Lsrpoolid    ==> 02                    1-15
   Maxkeylength ==> 255                   0-255
   SHarelimit   ==> 001                   1-100
   STrings      ==> 100                   1-255
  DATA BUFFERS
   DATA512      ==>                       3-32767
   DATA1K       ==>                       3-32767
   DATA2K       ==> 00010                 3-32767
   DATA4K       ==> 00100                 3-32767
      ?
      ?
      ?
   DATA32k      ==> 00010                 3-32767
  INDEX BUFFERS
   INDEX512     ==> 00010                 3-32767
   INDEX1K      ==> 00010                 3-32767
   INDEX2K      ==> 00010                 3-32767
      ?
      ?
      ?
   INDEX32k     ==>                       3-32767
```

If "**STRINGS**" and buffer sizes are specified, "**SHARELIMIT**" is ignored.

Don't be stingy with "**STRINGS**" and buffer sizes.

On-line Considerations

Read Integrity Problems:

- **Symptom:** No record found condition when the application should expect to find the record.
- **"Illogic Error". EIBRCODE = '020890'**
- **Resolution:**
  - **Ensure the record really does exist in the cluster:**

If record does not exist:

When VSE/VSAM Record Management receives an update request for an alternate index:

1. Lock the base cluster record: Prevents another request from locking the base record while we are updating the alternate index.
2. Update the alternate index record:
3. Update the base cluster record. If there is an error, VSE/VSAM attempts to backout the alternate index update.
4. If unable to backout the aix update, the update must be manually backed out, or the alternate index must be re-built.

On-line Considerations

If record does exist:

- **VSE/VSAM only guarantees read integrity with SHR(1)**
- **SHR(2) is the biggest problem**
- **SHR(4) provides "pretty good" read integrity protection.**
  - Direct retrieval of records results in unique I/O
  - Records written to dasd immediately
  - Quick method to resolve most read integrity problems
  - Performance penalty
- **Update application can issue an "End Request" (CICS FILE UNLOCK command.)**
- **Or … close and reopen the file after updates are complete.**

On-line Considerations

Speaker Notes:

- **Ensure the record really does exist in the cluster:** Close all open files in on-line partitions. Ensure no batch programs are updating this file, and do an IDCAMS **REPRO PRINT FROMKEY**

If record does not exist:
There was an error during alternate index updating, and alternate index will need to be re-built. This can happen if you are short on LSR buffers.

When VSE/VSAM Record Management receives an update request for an alternate index:

1. Lock the base cluster record: Prevents another request from locking the base record while we are updating the alternate index.
3. Update the base cluster record. If there is an error, VSE/VSAM attempts to backout the alternate index update. The most common error is "no available LSR buffers" (rc x'98')
4. If unable to backout the aix update, the update must be manually backed out, or the alternate index must be re-built. VSE/VSAM sets byte two of RPLFDBK to x'01', and returns to application. It is CICS's responsibility to alert the application appropriately, and prevent the request from being simply retried. A retry may fail with "Duplicate Record", since the aix key already exists, which may result in the transaction simply throwing the update away.

If record does exist:
Read integrity problems occur when a record has already been processed for insertion into a VSE/VSAM cluster, but another application cannot find the record. Either the record is still in the buffer of the update application, or the read application is using old buffers.

- **VSE/VSAM only guarantees read integrity with SHR(1):** If an application has the cluster open in update mode, no one can open it for read.
- **SHR(2) is the biggest problem:** Updated records are retained in buffers until the buffer is required for another request.

On-line Considerations

Dataset Name Sharing

- Multiple output OPENs for SHR(2) file
- Shares more than just buffers
- **CEDA DEFINE FILE DSNSHARING(ALLREQS)**
  -or-
  DFHFCT TYPE=FILE,DSNSHR=ALL,BASE=NAME
- Advantages:
  - Multiple Output OPENs (see above)
  - More efficient usage of storage
  - Buffer sharing, reduce chance of read integrity problems
- Restrictions:
  - First FCT entry opened determines access mode
  - First FCT entry opened must specify sufficient resources (strings, buffers). Recommend: Use LSR.
  - Reuseable files are not supported.
- Recommendations:
  - Only use if absolutely necessary
  - Use both Dataset Name Sharing and LSR

On-line Considerations

Speaker Notes:

- **Multiple output OPENs for SHR(2) file:** Some on-line applications wish to update a cluster via multiple FCT entries, often a base and alternate index. Unfortunately, SHR(2) only allows a single output open, so if the base is open for output, the alternate index can only be opened for read. Defining the files to CICS as "dataset name sharing" bypasses this restriction.
- **Shares more than just buffers:** When a group of FCT entries are marked as "**DATASET NAMESHARING**", they share all internal VSE/VSAM control blocks. As additional FCT entries are opened, they are added to the sharing list. As FCT entries are closed, they are moved from the list. The cluster stays open until the last entry is removed from the list, at which time VSE/VSAM closes the file and frees the internal control blocks.
- **CEDA DEFINE FILE DSNSHARING(ALLREQS) or DFHFCT TYPE=FILE,DSNSHR=ALL,BASE=NAME:** for CICS/VSE, "**BASE**" must also be specified, specifying a unique name for this set of shared files. This is not required for CICS TS (CEDA).
  - **Buffer sharing, reduce chance of read integrity problems:** between base and alternate index.
  - **First FCT entry opened determines access mode:** Input or Output. If an FCT entry opens the file in output mode, followed by an entry for input mode, the file remains opened for output. Even if the first file is closed, the file remains opened for output, making it unavailable for batch update. If the first open is for input, a later open of a different FCT for output is rejected.
  - **First FCT entry opened must specify sufficient resources (strings, buffers).**
    Recommend: Use LSR. CICS and VSE/VSAM make reasonably accommodations, but if more than three files are opened, the default values might not prove to be sufficient.
  - **Only use if absolutely necessary.** Complicates VSE/VSAM control block and buffer mgmt.
  - **Use both Dataset Name Sharing and LSR:** Eliminates resource problem if more than three files are opened, as well as other LSR advantages.

Default Models

- NOALLOCATION files (only a catalog entry)
- Reserved name:
  DEFAULT.MODEL.ESDS
  DEFAULT.MODEL.ESDS.SAM
  DEFAULT.MODEL.KSDS
  DEFAULT.MODEL.RRDS
  DEFAULT.MODEL.VRDS
  DEFAULT.MODEL.AIX
- Over-rides system defaults
- Useful for defining default volumes
- Any file can be used as model for a subsequent IDCAMS DEFINE by coding the MODEL parameter
- See write-up in "*VSE/VSAM User's Guide*"

Default Models

Missing DEFAULT.MODEL

The following series of messages appears rather intimidating, but in actuality only mean that you tried to open a SAM ESDS file (new file for output) without a DEFAULT.MODEL defined in the catalog (or perhaps defined a volid on the // EXTENT card which did not match the candidate volumes in the default model.

```
4A37I FILE SAMFILE  CATALOG ERROR DURING IMPLICIT DEFINE - 248,BX,000
```

(248 indicates "Volume Record not found", "BX" is IGG0CLBX, the VSAM Catalog Mgmt routine that caught the error)

```
4228I FILE SAMFILE    OPEN  ERROR X'4F'(079) CAT=UCAT    (  8,CG,  6)
      (IKQOPN-3) RC X'0000004F' ON CALL TO  IKQOPNHC
```

('4F' indicates an error during implicit define. "08..06' indicates a no-record-found condition, in this case, the search for the default model. "CG" is, of course, IGG0CLCG)

```
0V15I REQUEST FROM SYSTEM SERVICE ROUTINE
0S08I LOG. TRANS. AREA CANCELED, PHASE = $$BOSMXT
```

($$BOSMXT is the SAM interface to VSAM, so he is the one that catches it when something goes bump in the night)

```
0S00I JOB RUN      CANCELED.
1I51I DUMP COMPLETE
1S78I  JOB TERMINATED DUE TO  PROGRAM ABEND
EOJ RUN
```

SAM-Managed Files

- Expedites migration from dasd sequential files (DTFSD) to native VSAM.
- Allows programs to access VSAM files without re-write.
- Allows dynamic allocation of files (i.e. compiler / sort work files)
- No comparable function under MVS | OS/390 | z/OS.
- VSAM SAM ESDS files are slightly different (non-CA) from ESDS files.

```
// DLBL SAMFILE,'THIS.IS.A.SAM.FILE',,VSAM,CAT=<catalog fname>,     X
        RECORDS=(<primary>,<secondary>),RECSIZE=nn                  X
        DISP=(<open disp>,<close disp>,<error disp>)                X
```

OPEN:

- Space Management intercepts OPEN request: Points DTF at location of VSAM cluster. If file does not already exist, it will be dynamically defined. This requires a default model (for volume information), and **RECORDS** and **RECSIZE** parameters on "// **DLBL**".
- Some programs require extent to be contiguous
  File-id = '`DOS.WORKFILE.SYS...`'
  Special handling for **'DOS.WORKFILE.SYSLNK'**
- Disposition Processing: "**DISP=(OLD | NEW,…)**"
  - If file already exists, "**DISP=(NEW,…)**" will only reset file, not delete and re-define.
- Place pointer to SSR Service Routine ($IJGXSRV) into DTF Extension (DTFX)

SAM-Managed Files

Record Processing:

- I/O requests branch-and-link (BAL) directly to special BAM module ($IJGXSRV and its relatives).
- BAM handles I/O: executes EXCP for I/O, reads, deblocks, returns –or- blocks and writes user records into a VSE/VSAM-format Control Interval
- If this extent fills up: SAM sets "Extend mode" into DTF, and issues a new OPEN. VSE/VSAM adds an additional extent (if possible), updates DTF, and returns to BAM.
- If no extension is possible, job is cancelled with: "`4250I NO MORE AVAILABLE EXTENTS`"
- Does not work with OEM dasd managers (DYNAM/D, EPIC)

CLOSE:

- Space Management intercepts CLOSE request and resets DTF fields.
- Disposition Processing: DISP=(…,DELETE | KEEP | <date>,…)

  -or, if job cancelled: DISP=(…, …, DELETE | KEEP | <date>)
- No record statistics kept in catalog.

Librarian / DB2

- VSE Libraries are non-CI-Format VSE/VSAM clusters.
  - VSAM only provides Catalog Management and OPEN/CLOSE support
  - Libraries are defined SHR(3), access is controlled via VSE/ESA locks.
  - After OPEN processing, VSE/VSAM closes the Library in catalog.
    - Possible to delete library using IDCAMS while still open.
  - See next page for sample jcl.
- SQL | DB2 tables are in VSE/VSAM CI format.
  - DB2 accesses data using Control Interval mode
- Don't use VSAM file utilities (REPRO, Backup/Restore)

Librarian / DB2

Sample JCL:

Here are the steps you may follow to define and initialize a new VSEDUMP library:

Step 1: Define SYSDUMP library in VSAM space:

```
// JOB DEFINE NEW LIBRARY
// EXEC IDCAMS,SIZE=AUTO
 DELETE VSE.DUMP.LIBRARY PURGE                -
        CATALOG (VSAM.MASTER.CATALOG)
 DEFINE CLUSTER (                             -
        NAME (VSE.VSAM610.LIBRARY)            -
        CYL (000200 00000)                    -
        SHAREOPTIONS (3)                      -
        RECORDFORMAT (NOCIFORMAT)             -
        VOLUMES (CTS220,CTS240)               -
        NOREUSE                               -
        NONINDEXED                            -
        TO (99366))                           -
     DATA (NAME (VSE.DUMP.LIBRARY.@D@))       -
        CATALOG (VSAM.MASTER.CATALOG)
  IF LASTCC NE 0 THEN CANCEL JOB
/*
/&
```

Step 2: Define the SYSDUMP library to VSE Librarian:

```
// JOB SYSDUMP DEFINE A NEW LIBRARY
// OPTION NODUMP,NOSYSDUMP
// DLBL SYSDUMP,'VSE.DUMP.LIBRARY',99/365
// EXTENT SYS018,PACCC3,1,0,1500,750
// ASSGN SYS018,DISK,VOL=PACCC3,SHR
// EXEC LIBR,PARM='MSHP'
   DEFINE L=SYSDUMP
   DEFINE SUBLIB=SYSDUMP.BG
   DEFINE SUBLIB=SYSDUMP.F1
   DEFINE SUBLIB=SYSDUMP.F2
   DEFINE SUBLIB=SYSDUMP.F3
   DEFINE SUBLIB=SYSDUMP.F4
   DEFINE SUBLIB=SYSDUMP.F5
   DEFINE SUBLIB=SYSDUMP.DYNAMIC
/*
/&
```

Librarian / DB2

Step 3: Define BLNXTRN and BLNDMF files. BLNXTRN contains the external routines (dump analysis routines). BLNDMF contains the dump management files.Remove the VTOC label for both BLNXTRN and BLNDMF prior to running this job. You can use the DITTO PVT function if you wish

```
// JOB INIT SYSDUMP MANAGEMENT FILE
// OPTION NODUMP,NOSYSDUMP
// DLBL SYSDUMP,'VSE.DUMP.FILE',99/365,,DSF
// EXTENT SYS018,PACCC3,1,0,1500,750
// ASSGN SYS018,DISK,VOL=PACCC3,SHR
// DLBL BLNXTRN,'EXT.RTNS.FILE',1999/365,SD
// EXTENT SYS017,PACC19,1,0,1495,5
// ASSGN SYS017,DISK,VOL=PACC19,SHR
// DLBL BLNDMF,'DUMP.MGMT.FILE',1999/365,SD
// EXTENT SYS017,PACC19,1,0,1485,10
// EXEC INFOANA,SIZE=300K
   SELECT DUMP MANAGEMENT
   UTILITY
   RETURN
   SELECT END
/*
/&
```

```
// JOB INIT EXTERNAL ROUTINES FILE
// OPTION NODUMP,NOSYSDUMP
// DLBL BLNXTRN,'EXT.RTNS.FILE',1999/365,SD
// EXTENT SYS017,PACC19,1,0,1495,5
// ASSGN SYS017,DISK,VOL=PACC19,SHR
// UPSI 1
// EXEC DITTO
   $$DITTO CSQ FILEOUT=BLNXTRN,BLKFACTOR=1
   ANEXIT DFHDAP     (CICS 2.3 Dump Analyzer 2.3)
   ANEXIT DFHPD410   (CICS TS Dump Analyzer)
   ANEXIT IJBXCSMG   (Format Console Buffer)
   ANEXIT IJBXDBUG   (Format Standalone Dump)
   ANEXIT IJBXSDA    (Format SDAID Buffer)
/*
$$DITTO EOJ
/*
/&
```

Librarian / DB2

This page
Was Left
Intentionally
Blank

Utilities

```
// EXEC IKQVCHK,SIZE=AUTO,PARM='<catalog.name>'
```

- Checks catalog for internal consistency, primarily between volume, cluster and true-name descriptor records.
- Customer's should run as part of normal monthly maintenance

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

- Internal trace points (SNAP TRACE) within VSAM
- Output to SYSLST (except SNAP001 and SNAP013)
- Only use under advisement from Level2

```
// ASSGN SYS000,DISK,VOL=<volid>,SHR
// UPSI nn
// EXEC IKQVDU,SIZE=AUTO
```

- Manipulate VTOC (delete, define, reset ...)
- Only use under advisement from Level2

Utilities

```
// EXEC IKQPRED,PARM='<catalog.name> / <cluster.name>'
```

See "*VSE/VSAM User's Guide and Application Programming*" (SC33-6732-00)

- Compression prediction
- Checks an entire catalog or a series of files (generic specification supported).

```
// EXEC PROC=IDCONS
```

- Interactive batch interface to IDCAMS
- Can be used to enter any IDCAMS command interactively from console
- Extremely useful in system recovery situations.

```
VSAMIO
```

See "*REXX/VSE 2.6 Reference*" (SC33-6642-07)

- Allows access of VSE/VSAM data from REXX/VSE.
- Belongs to the host command environment ADDRESS VSE.
- Supports KSDS, ESDS, and RRDS data sets.
- Supported operations are READ, WRITE, DELETE, and UPDATE.
- Can read or write data from/to REXX stem variables.

Utilities

IDCONS (Sample Invoction) :

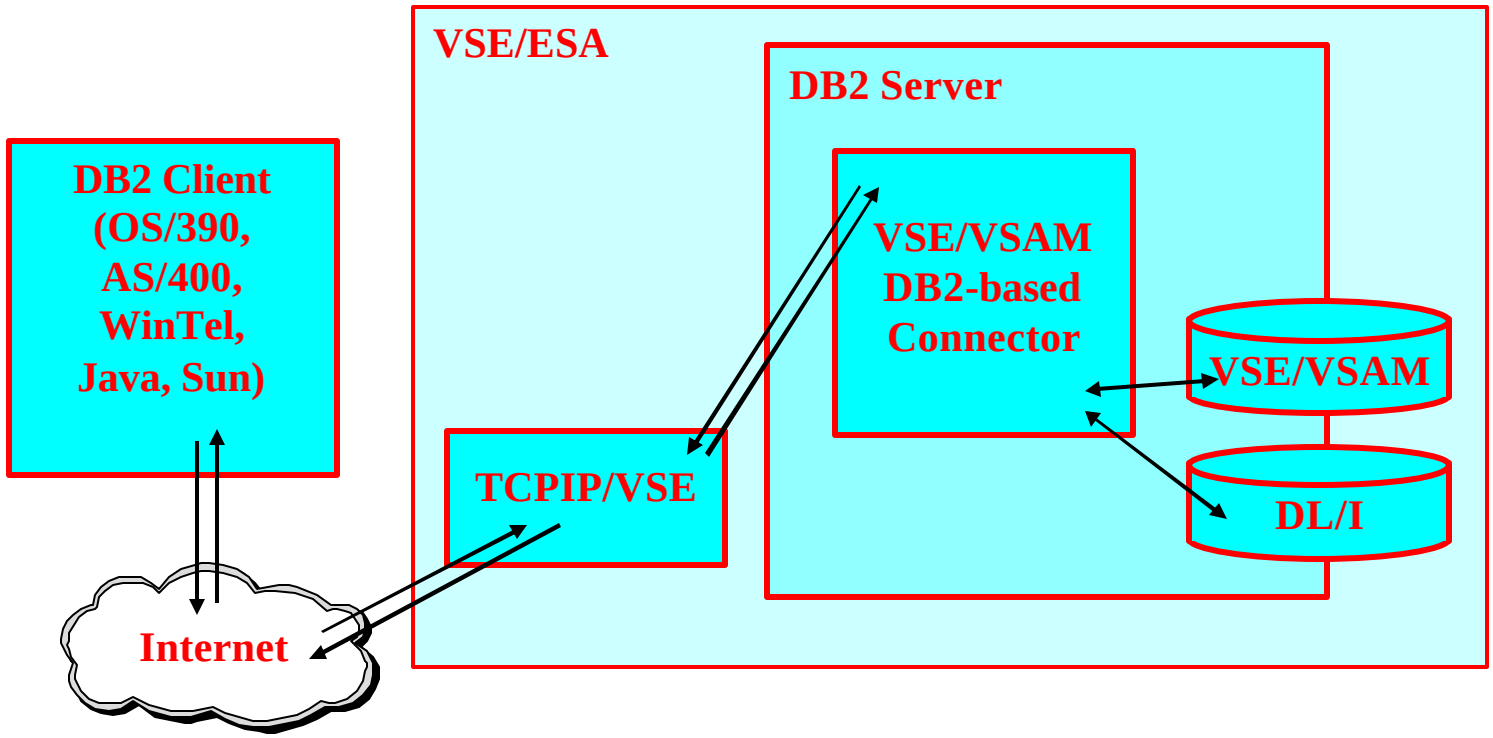
```
BG 0001 1Q47I   BG PAUSEBG 00757 FROM (SYSA) , TIME=16:08:46
BG 0000 // JOB PAUSEBG
         DATE 08/29/2002, CLOCK 16/08/46
BG-0000 // PAUSE
0 // EXEC IDCONS
BG 0000 IDCONS400I IDCONS 1.0  Console Interface to Access Method
Services
BG 0000 IDCONS402I Select output device:
BG 0000 IDCONS403I 1= SYSLOG (compressed); 2= SYSLOG (normal); 3=SYSLST
BG 0000 IDCONS405D Reply 1, 2 or 3

BG-0000
0 1
BG 0000 IDCONS406D Enter IDCAMS command or QUIT to exit

BG-0000
0 LISTCAT ENTRIES(VSE.CONTROL.FILE) CATALOG(VSESP.USER.CATALOG)
BG 0000
BG 0000  LISTCAT ENTRIES(VSE.CONTROL.FILE) CATALOG(VSESP.USER.CATALOG)
BG 0000 IDCAMS SYSTEM SERVICES TIME: 16:10:55 08/29/2002 PAGE 2
BG 0000  LISTING FROM CATALOG - VSESP.USER.CATALOG
BG 0000 CLUSTER - VSE.CONTROL.FILE
BG 0000  DATA - VSE.CONTROL.FILE.@D@
     .
     .
     .
```

Utilities

This page was deliberately left blank

e-Business Connectors

DB2 Server for VSE and VM, release 7.1 or 7.2 (2.6)



e-Business Connectors

DB2 Server for VSE and VM, release 7.1 or 7.2 (2.6) (continued)

- DB2 Server is a separately charged product.
- Distributed on VSE/ESA extended base tape with “trial” key.
- Exploits DB2 connection infrastructure (JDBC / ODBC, DB2 Connect, etc)
- Exploits DB2 Stored Procedure facility
- Exploits Distributed Relational Database Connectivity (DRDA)
  - Standard defining interactions with relational databases over a network
- VSE/VSAM DB2-based Connector
  - Maps SQL requests to/from VSE/VSAM or DL/I datasets
  - Supports VSE/VSAM data mapping

e-Business Connectors

DB2 VSAM Transparency for VSE/ESA

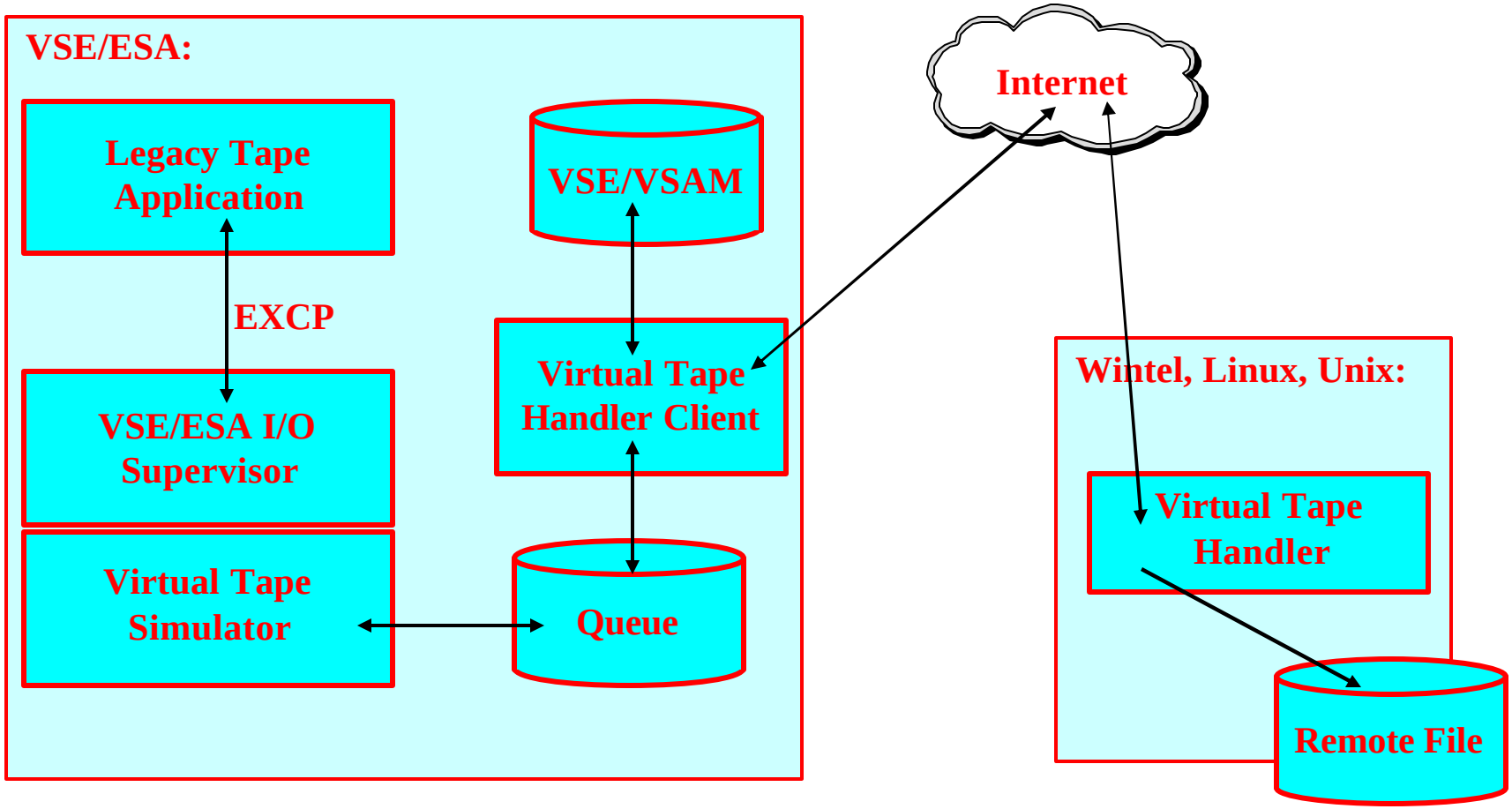
See also "*DB2 for VSE System Administration*" (GC09-2406), and
"*DB2 for VSE Database Administration*" (GC09-2398).

- **Benefit:**
  - **Relational database manager access to legacy data:**
    Without the need to immediately rewrite your VSE/VSAM applications.
  - **Migration Path to DB2:**
    With DB2 VSAM Transparency for VSE/ESA, migration to a relational database no longer means you have to convert all your VSE/VSAM applications and replace VSE/VSAM calls with SQL/DS at once. DB2 VSAM Transparency for VSE/ESA lets you move some or all of your data to DB2 Server for VSE and allows you to continue using your existing applications.
  - **SHR(2):** VSE/VSAM files controlled from a single batch partition. Allows multi-partition update access to SHR(2) files.
- **Implementation:**
  Data is migrated to DB2. VSE/VSAM calls from legacy applications are routed to DB2 Transparency Manager, who satisfies the request from the DB2 database and returns the data in a manner and form expected by a VSE/VSAM application.

e-Business Connectors

This page was deliberately left blank

Virtual Tape (2.6)



Virtual Tape (2.6)

Output Redirection for applications which write to "real" tapes:

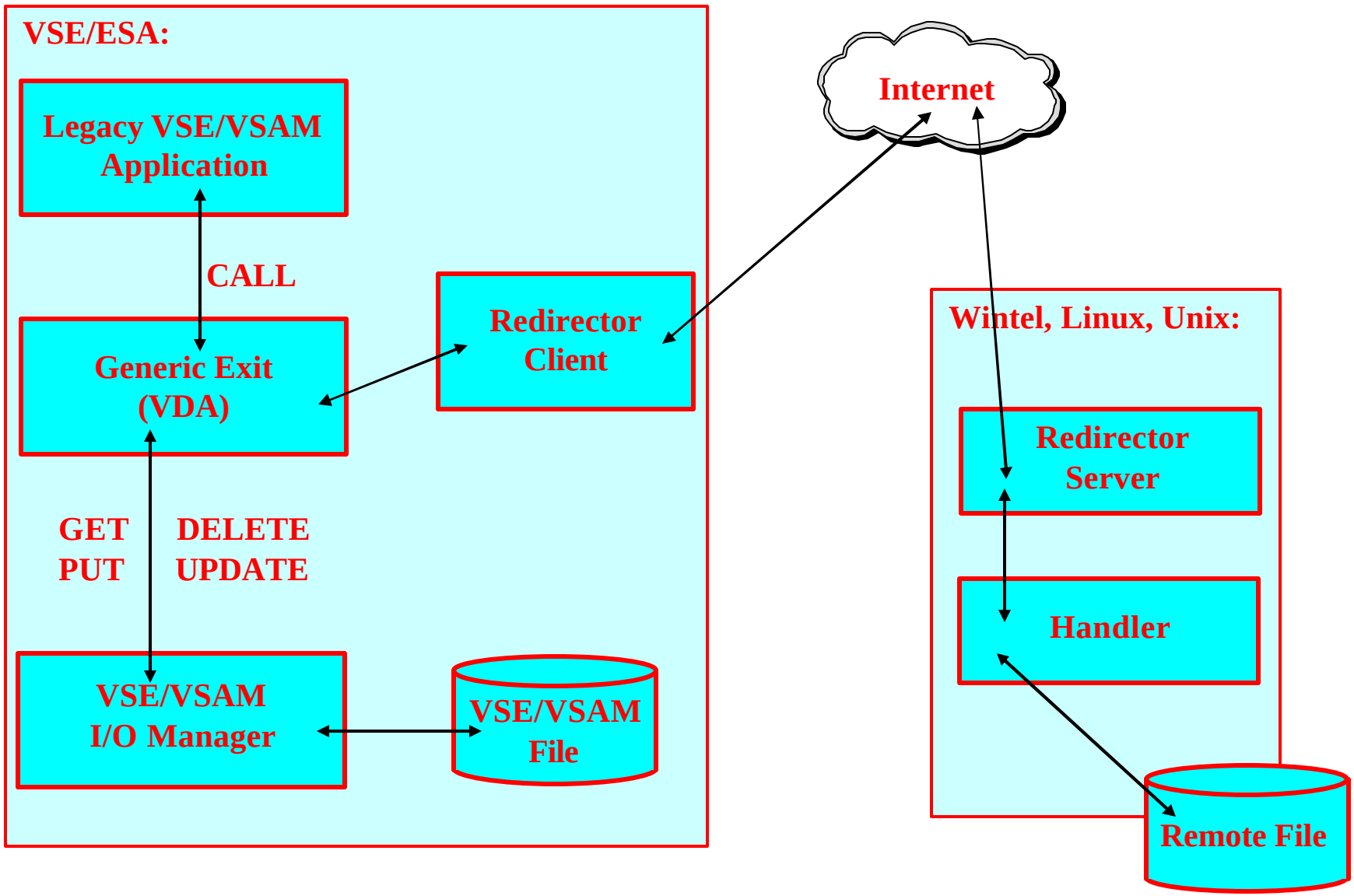
1. Target can be a VSAM ESDS file, or a remote file (on remote host).
2. No changes required for application.
3. Local File (VSAM ESDS):

```
DVCDN 480
VTAPE START,UNIT=480,LOC=VSAM,FILE='TEST.TAPE.FILE'
DVCUP 480
. . . APPLICATION RUNS . . .
DVCDN 480
VTAPE STOP,UNIT=480
DVCUP 480
```

4. Remote File:

```
DVCDN 480
VTAPE START,UNIT=480,LOC=9.64.128.5,                X
         ,FILE='E:\BACKUPS\CHUCK''S BACKUP FILE.TXT'
DVCUP 480
. . . APPLICATION RUNS . . .
DVCDN 480
VTAPE STOP,UNIT=480
DVCUP 480
```

VSE/VSAM Re-director



VSE/VSAM Re-director

Output Redirection for applications which read / write to VSE/VSAM files:

1. Output can be to a remote file or split with local VSE/VSAM file. This allows synchronizing remote files.

2. No changes required for application.

3. VSE/VSAM request intercepted by “VDA” exit (IKQEX01).

   - Processes I/O according to Config File.
   - Skeleton in ICCF Library 59.
   - VSE/ESA 2.7 implemented optional parameters when specifying the names of clusters that are to be redirected, for more exact filtering

4. Redirected requests passed to VSE/VSAM Redirector Client.

   - File requests redirected to I/P address specified in Config File

5. Remote requests intercepted by VSE/VSAM Redirector Server on remote WinTel or Linux/Unix machine.

   - Calls user-written “handler” routine.

6. **Handler routine** must handle data translation, field mapping, and writing to remote file.

   - Written in Java, and resides in directory of Redirector Server.
   - Two samples (DB2 and HTML) distributed with VSE/ESA 2.6