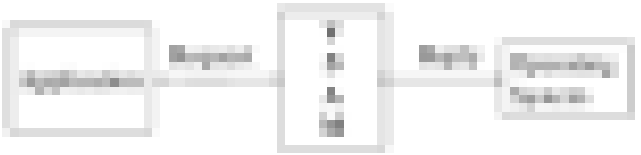
VSAM

VSAM Fundamentals

- Course Objectives
  - Introduction to VSAM
  - VSAM Data Set Organization
  - IDCAMS

What is VSAM?

- VSAM is Virtual Storage Access Method
- It is a method used to move data between DASD and Main Storage
- VSAM operates in Virtual Environment
- VSAM acts as interface between Operating System and Application Program



Types of VSAM data sets

- ESDS Entry Sequenced Data Set
- KSDS Key Sequenced Data Set
- RRDS Relative Record Data Set
- LDS Linear Data Set

Traditional access methods

- QSAM (Queued Sequential Access Method)
- BSAM (Basic Sequential Access Method)
  - for "flat" files
- BDAM (Basic Sequential Access Method)
  - for index files
- BDAM (Basic Direct Access Method)
  - for direct access files

VSAM History

- 1973 : VSAM was introduced with OS/VS & VS/1
- 1978 – Alternate Index, KSDS, Catalog Recovery Feature is introduced
- 1979 – VSAM re-introduced with ICF (Integrated Catalog facility)
- 1983 – DFP 2 VSAM ran under MVS / XA & (Data Facility Product)
- 1987 – DFP 3 VSAM 3.3 introduced with LSR

Advantages

- One method to support all types of data retrieval
- Can be a stand alone file or within a DBMS
- Supports fixed and variable records
- One utility (IDCAMS) to manage everything (DEFINE for any files, IMPORT for PDS, EXPORT for VSAM files)
- Supports alternate index
- Device independence (EXPORT and IMPORT)
- Portable across systems
- Wonderful catalog contains all the information
- Catalogs are protected by internal security

Disadvantages

- Requires lot of DASD space
- For KSDS, primary key cannot be changed
- Performance can be slow because of complexity of index
- Only available on DISK

IDCAMS

- Handles VSAM data sets (exclusively)
- Some of the functions
  - Creates
  - Copies
  - Prints
  - Deletes
  - Lists characteristics

VSAM data set organization

VSAM data set organization

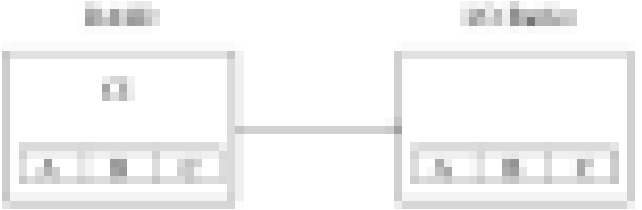
- VSAM Data Set can contain three major components
  - CLUSTER (Catalog entry)
  - INDEX
  - DATA (Actual data)
- Data Set is referred by cluster name in JCL

VSAM intervals

- CONTROL INTERVAL (CI)
- VSAM stores Data and Index in Control Intervals (CI)
- CI is similar to "Block"

Control Interval

- CI is the unit of retrieval between DASD and I/O Buffer (Virtual Storage)



Control Interval

- CI contains:
  - Records (or DASDs)
  - Free space (Optional)
  - Control Interval Definition Field (CIDF)
  - Record Definition Field (RDF)

RDF and CIDF

- RDF
  - 3 bytes long
  - Indicates length of records
  - And also how many adjacent records are of the same length
- CIDF
  - 4 bytes long
  - One per CI
  - Indicates free space

CONTROL AREA (CA)

- CIs are grouped into CAs
- Can have more than one CA in a VSAM data set
- CA is VSAMs internal unit for allocating space
- Smallest is a TRACK, and the largest is a CYLINDER
- VSAM determines CA size & number of CIs in CA based on
  - CI size specified

Index

- Separate entity
- Organized similar to Data component (CI & CA)
- Has different CI view
- VBuild builds index when data is loaded
- Index is organized as inverted Binary Tree
- VBuild compresses keys to conserve space
- Can have several levels of indexes
  - Lowest level of index is called "sequence set"
  - Next level is called index set

CI & CA Split

- CI split happens while
  - Adding new records or extending an old record
  - And not enough room in the CI to complete the operation
- CI split may trigger a CA split
- Splits generally degrades performance
- Specify freespace to reduce CI & CA splits

ESDS

- Similar to Sequential file
- Sequenced by the order in which data is entered/loaded
- New Records are added at the end only (chronological order)
- Supports both Fixed and Variable formats
- Contains only CLUSTER & DATA components
- Only sequential access in Batch Cobol Programs
- Random access is supported in on-line applications (CICS) using Relative Byte Address (RBA)
- Alternate index is supported in on-line applications (CICS)
- NO primary index

RRDS

- Has only CLUSTER and DATA components
- Records are stored in numbered, fixed length slots
- Each slot is given a number 'Relative Record Number (RRN)'
- VSAM determines the number of slots by
  - Size of CI
  - Length of Record
- Records can be deleted physically

KSDS

- Has all three components of VSAM (CLUSTER, INDEX and DATA)
- Key sequenced
- Primary key should be
  - Unique
  - Same position in every record
  - In non-split (Not to be overlapped)
- Records can be deleted physically
- Primary key cannot be changed
- Allows Alternate Index
- Has all the access methods
  - Sequential
  - Random
  - Dynamic (Both sequential)

VSAM dataset choice

	KSDS	ESDS	RRDS
Low RAID			
Variable Record Access			
Batch Sequential			
			

Working with VSAM Datasets

Access Method Services (AMS)
IDCAMS

IDCAMS

- Normally executed in Batch
- Always has the following JCL structure
  //STEP01 EXEC PGM=IDCAMS
  //SYSPRINT DD SYSOUT=*
  //SYSIN DD *

Basic IDCAMS Commands

- DEFINE (Cluster, Alternate Index etc.)
- BUILDINDEX (Alternate Index)
- REPRO (Cluster)
- LISTCAT (Catalog Entries)
- IMPORT / EXPORT (Cluster)
- VERIFY (Cluster)

Basic define command syntax for cluster

```
DEFINE CLUSTER                      -
(NAME(CONDL.NET.CLUSTER)             -
CYLINDERS (1 1)                      -
    VOLUMES (WORK1)                  -
    <data set type>                  -
)
```

Define Command (Contd..)

- NAME is a positional keyword parameter and must be coded first.
- Other keywords can be placed anywhere
- '-' sign is used for continuation within a field

- e.g.

  DEFINE CLUSTER -

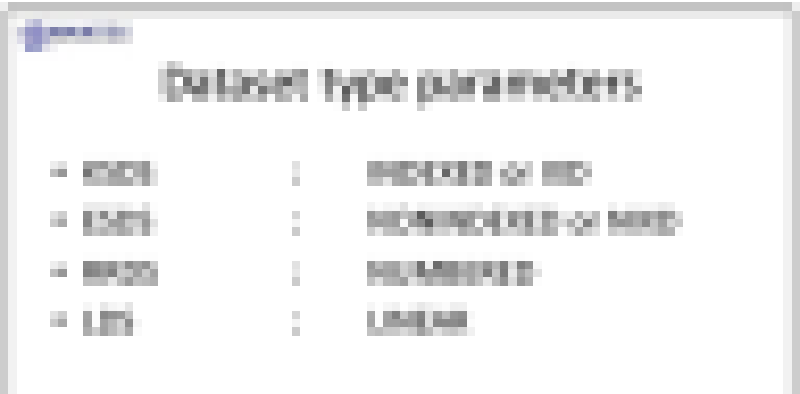
  (NAME([illegible]) -

  [illegible])

Record size parameter

- Syntax : RECORDSIZE (Average Maximum)
- Optional: Default : 4089
- Average & Maximum are same for fixed length records
  - DEFINE CLUSTER -
  - (NAME(KSDS.NCI.CLUSTER)) -
  - [illegible] -


KEYS Parameter
Optional : Default Value : Length=64 & Offset=0


Dataset type parameters

Data & Index components

Required when installation defaults names are to be overridden

DEFINE CLUSTER ...

(NAME) (VOLUMES) ...

CYLINDERS (1 1) VOLUMES (WORK01) ...

RECORDSIZE (120 4089) SHAREOPTIONS 3 ...

DATA ...

(NAME) (VOLUMES) ...

INDEX ...

(NAME) (VOLUMES) ...

VOLUMES

- Can specify different volumes for
  - Data component
  - Index component

MORE AMS COMMANDS

REPRO

- All purpose load and backup utility command
- Can be used against empty / loaded VSAM file with another VSAM file or sequential file
- Much easier to use than IDCAMS
- Can be used against all four types of VSAM datasets

REPRO (Contd..)

Repro (Contd..)

- INFILE or INDATASET parameter is mandatory, similarly OUTFILE or OUTDATASET is mandatory
- All other parameters are optional
- SKIP specifies number of input records to skip before beginning to copy

Repro (Contd..)

- REUSE parameter
  - Can be used only if the VSAM dataset was originally defined with REUSE option
  - Has the effect of logically deleting records before loading
- REPLACE parameter
  - Replaces the records for which primary keys are matching between input and output records
  - If not specified, the matching key records are unloaded
  - If the target is ESDS the records are appended and REPLACE is inappropriate.

Repro - Example

//REPRO EXEC PGM=IDCAMS

//SYSPRINT DD SYSOUT=*

//IDS1 DD DSN=IBMUSER.KSDS.CLUSTER,DISP=SHR

//IDS1 DD DSN=IBMUSER.KSDS.DATA, DISP=SHR

//SYSIN DD*

EXPORT/IMPORT

- Used for backup and recovery
- Catalog information also is exported along with the data, unlike REPRO
- DFSMS classes are preserved
- Cluster deletion and redefinition are not necessary during the import
- Can be easily ported to other systems

EXPORT/IMPORT (Cont'd...)

- Disadvantages
  - The ESTIMATE file must exist before it is imported
  - Slower than REPRO

VERIFY

- Syntax : VERIFY FILE [<dataname>]
  VERIFY [ALTINDEX] <dataentryname>]
  - Can be issued from TDP or from a CL.
  - Verifies the catalog HURBA (High Used Relative Byte Address) field and stores the true values from the control block HURBA field.
  - Should be used against cluster name only and not against data or index components.
  - Used to rectify some of the problems due to data corruption.

LISTCAT

Used to list the contents of a master or user catalog

LISTCAT CATALOG (name)
ENTRIES (name : ref : entries)
LEVEL (generic-level-name(s))

Other parms:

ALL NAME

DELETE

- DELETE <object name> [parameters]
- Example : DELETE [illegible]
  - All the parameters are optional
  - Deletes all subordinate objects such as dd, Path

Some Common DELETE Parameters

- ERASE / NO ERASE : ERASE writes binary zeroes after deletion
- PURGE / NO PURGE : PURGE allows deletion even though expiration date is still due
- ALTERNATE INDEX Or AIX : Deletes only Alternate Index of the cluster


Thank you !