



VSAM

virtual storage access method

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

 <https://www.facebook.com/tutorialspointindia>

 <https://twitter.com/tutorialspoint>

About the Tutorial

VSAM stands for Virtual Storage Access Method. VSAM is a file storage access method used in MVS, ZOS and OS/390 operating systems. It was introduced by IBM in 1970's. It is a high performance access method used to organize data in form of files in Mainframes.

VSAM is used by COBOL and CICS in Mainframes to store and retrieve data. VSAM makes it easier for application programs to execute an input-output operation.

Audience

This tutorial is designed for software programmers with a need to understand the VSAM concepts starting from scratch. This tutorial will give you enough understanding on VSAM from where you can take yourself to higher level of expertise.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of JCL and COBOL. A basic understanding of any of the file handling method will help you in understanding the VSAM concepts and move fast on the learning track.

Disclaimer & Copyright

© Copyright 2018 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Disclaimer & Copyright	i
Table of Contents.....	ii
1. VSAM - OVERVIEW	1
Characteristics of VSAM.....	1
Limitations of VSAM	1
2. VSAM – COMPONENTS	2
VSAM Cluster.....	2
Control Interval	2
Analysis of Control Interval.....	3
Control Area	3
3. VSAM - CLUSTER	4
Defining a Cluster	4
Deleting a Cluster	7
4. VSAM – ENTRY SEQUENCED DATA SET	9
Defining ESDS Cluster	9
Deleting ESDS Cluster	11
5. VSAM – KEY SEQUENCED DATA SET	12
KSDS File Structure	12
Defining KSDS Cluster	13
Deleting KSDS Cluster	14

6. VSAM – RELATIVE RECORD DATA SET	16
RRDS File Structure	16
Defining RRDS Cluster	17
Deleting RRDS Cluster	18
7. VSAM – LINEAR DATA SET	19
Defining LDS Cluster	19
Deleting LDS Cluster.....	20
8. VSAM – COMMANDS	22
Alter	22
Repro.....	23
Listcat	24
Examine.....	25
Verify.....	25
9. VSAM – ALTERNATE INDEX.....	27
Creation of Alternate Index	27
Define Alternate Index.....	27
Define Path.....	29
Building Index.....	30
10. VSAM – CATALOG	31
Master Catalog	31
User Catalog	32
Data Space.....	32
Unique Clusters	32
Sub-allocated Clusters	32
Non-VSAM datasets.....	32

11. VSAM – FILE STATUS33

12. VSAM – INTERVIEW QUESTIONS35

1. VSAM - Overview

Virtual Storage Access Method (VSAM) is high performance access method and data set organization, which organizes and maintains data via a catalog structure. It utilizes virtual storage concept and can protect datasets at various levels by giving passwords. VSAM can be used in COBOL programs like physical sequential files. VSAM are the logical datasets for storing records. Files can be read sequentially and randomly in VSAM. It is an improved way of storing data which overcomes some of the limitations of conventional file systems like Sequential Files.

Characteristics of VSAM

Following are the characteristics of VSAM:

- VSAM protects data against unauthorized access by using passwords.
- VSAM provides fast access to data sets.
- VSAM has options for optimizing performance.
- VSAM allows data set sharing in both batch and online environment.
- VSAM are more structured and organized in storing data.
- Free space is reused automatically in VSAM files.

Limitations of VSAM

The only limitation of VSAM is that it cannot be stored on TAPE volume. It is always stored on DASD space. It requires a number of cylinders to store the data which is not cost-effective.

2. VSAM – Components

VSAM consists of following components:

- VSAM Cluster
- Control Area
- Control Interval

VSAM Cluster

VSAM are the logical datasets for storing records and are known as clusters. A cluster is an association of the index, sequence set and data portions of the dataset. The space occupied by a VSAM cluster is divided in contiguous areas called Control Intervals. We will discuss about control intervals later in this module.

There are two main components in a VSAM cluster:

- **Index Component** contains the index part. Index records are present in Index component. Using index component VSAM is able to retrieve records from the data component.
- **Data Component** contains the data part. Actual data records are present in Data component.

Control Interval

Control Intervals (CI) in VSAM are equivalent to blocks for non-VSAM data sets. In non-VSAM methods, the unit of data is defined by the block. VSAM works with logical data area which is known as Control Intervals.

Control Intervals are the smallest unit of transfer between a disk and the operating system. Whenever a record is retrieved directly from the storage, the entire CI containing the record is read into VSAM Input-Output buffer. The desired record is then transferred to work area from VSAM buffer.

Control Interval consists of:

- Logical Records
- Control information fields
- Free Space

When a VSAM dataset is loaded, control intervals are created. The default Control Interval size is 4K bytes and it can extend up to 32K bytes.

Analysis of Control Interval

R1	R2	R3	R4	R5	FS	R D F	R D F	C I D F
----	----	----	----	----	----	-------------	-------------	------------------

Following is the description of terms used in the above program:

- **R1..R5:** Records which are stored in Control Interval.
- **FS:** FS is free space, which can be used for further expansion of dataset.
- **RDF:** RDF is known as Record Definition Fields. RDF are 3 bytes long. It describes the length of records and tells how many adjacent records are of the same length.
- **CIDF:** CIDF is known as Control Interval Definition Fields. CIDF are 4 bytes long and contain information about the Control Interval.

Control Area

A Control Area (CA) is formed by putting together two or more Control Intervals. A VSAM dataset is composed of one or more Control Areas. The size of VSAM is always a multiple of its Control Area. VSAM files are extended in units of Control Areas.

Following is the example of Control Area:

R1	R2	R3	R4	R5	FS	R D F	R D F	C I D F
R1	R2	R3	R4	R5	FS	R D F	R D F	C I D F
R1	R2	R3	R4	R5	FS	R D F	R D F	C I D F

Control Area

3. VSAM - Cluster

VSAM cluster is defined in **JCL**. JCL uses **IDCAMS** utility to create a cluster. IDCAMS is a utility, developed by IBM, for access method services. It is used to primarily define VSAM datasets.

Defining a Cluster

The following syntax shows the main parameters which are grouped under **Define Cluster**, **Data** and **Index**.

```
DEFINE CLUSTER (NAME(vsam-file-name)      -
BLOCKS(number)                            -
VOLUMES(volume-serial)                    -
[INDEXED / NONINDEXED / NUMBERED / LINEAR] -
RECSZ(average maximum)                   -
[FREESPACE(CI-Percentage,CA-Percentage)]  -
CISZ(number)                              -
[KEYS(length offset)]                     -
[READPW(password)]                        -
[FOR(days)|TO(date)]                      -
[UPDATEPW(password)]                      -
[REUSE / NOREUSE] )                      -
DATA                                       -
  (NAME(vsam-file-name.data))             -
INDEX                                       -
  (NAME(vsam-file-name.index))            -
CATALOG(catalog-name[/password]))
```

Parameters at the CLUSTER level apply to the entire cluster. Parameters at the DATA or INDEX level apply only to the data or index component.

We will discuss each parameter in detail in the following table:

Sr. No.	Parameters with Description
1	<p>DEFINE CLUSTER Define Cluster command is used to define a cluster and specify parameter attributes for the cluster and its components.</p>
2	<p>NAME NAME specifies the name of VSAM file for which we are defining the cluster.</p>
3	<p>BLOCKS Blocks specifies the number of blocks assigned for the cluster.</p>
4	<p>VOLUMES Volumes specifies one or more volumes that will contain the cluster or component.</p>
5	<p>INDEXED / NONINDEXED / NUMBERED / LINEAR This parameter can take three values INDEXED, NONINDEXED or NUMBERED depending upon the type of dataset we are creating. For key-sequenced(KSDS) files INDEXED option is used. For entry-sequenced(ESDS) files the NONINDEXED option is used. For relative-record(RRDS) files the NUMBERED option is required. For Linear(LDS) files the LINEAR option is required. The default value of this parameter is INDEXED. We will discuss more about KSDS, ESDS, RRDS and LDS in coming modules.</p>
6	<p>RECSZ Record Size parameter has two values which are Average and Maximum record size. The Average specifies the average length of the logical records in the file and the Maximum denotes the length of the records.</p>
7	<p>FREESPACE Freespace specifies the percentage of free space to reserve for the control intervals (CI) and control areas (CA) of the data component. The default value of this parameter is zero percentage.</p>
8	<p>CISZ CISZ is known as Control interval size. It specifies the size of control intervals.</p>
9	<p>KEYS Keys parameter is defined only in key-sequenced (KSDS) files. It specifies the length and offset of primary key from first column. The range of value of this parameter are from 1 to 255 bytes.</p>
10	<p>READPW Value in READPW parameter specifies the password of read level.</p>
11	<p>FOR/TO The value of this parameter specifies the amount of time in terms of date and days for retaining the file. The default value for this parameter is zero days.</p>

12	UPDATEPW Value in UPDATEPW parameter specifies the password of update level.
13	REUSE / NOREUSE REUSE parameter allows clusters to be defined that may be reset to empty status without deleting and re-defining them.
14	DATA - NAME The DATA part of the cluster contains the dataset name which contains the actual data of the file.
15	INDEX-NAME The INDEX part of the cluster contains the primary key and the memory pointer for the corresponding record in the data part. It is defined when a Key Sequenced cluster is used.
16	CATALOG Catalog parameter denotes the catalog under which the file will be defined. We will discuss about catalog separately in upcoming modules.

Example

Following is a basic example to show how to define a cluster in JCL:

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER (NAME(MY.VSAM.KSDSFILE) -
INDEXED -
RECSZ(80 80) -
TRACKS(1,1) -
KEYS(5 0) -
CISZ(4096) -
FREESPACE(3 3) )
        DATA (NAME(MY.VSAM.KSDSFILE.DATA)) -
        INDEX (NAME(MY.VSAM.KSDSFILE.INDEX))
/*
```

If you will execute the above JCL on Mainframes server. It should execute with MAXCC=0 and it will create MY.VSAM.KSDSFILE VSAM file.

Deleting a Cluster

To delete a VSAM file, the VSAM cluster needs to be deleted using IDCAMS utility. DELETE command removes the entry of the VSAM cluster from the catalog and optionally removes the file, thereby freeing up the space occupied by the object. If the VSAM data set has not expired, then it will not be deleted. To delete such types of datasets, use PURGE option.

```
DELETE data-set-name CLUSTER
[ERASE / NOERASE]
[FORCE / NOFORCE]
[PURGE / NOPURGE]
[SCRATCH / NOSCRATCH]
```

Above syntax shows the parameters which we can use with Delete statement. We will discuss each of them in detail in the following table:

Sr. No.	Parameters with Description
1	ERASE / NOERASE ERASE option is specified to override the ERASE attribute specified for the object in the catalog. NOERASE option is taken by default.
2	FORCE / NOFORCE FORCE option is specified to delete the SPACE and USERCATALOG even if they are not empty. NOFORCE option is taken by default.
3	PURGE / NOPURGE PURGE option is used to delete the VSAM dataset if dataset has not expired. NOPURGE option is taken by default.
4	SCRATCH / NOSCRATCH SCRATCH option is specified to remove the associated entry for the object from the Volume Table of Contents. It is mainly used for non-vsam datasets like GDGs. NOSCRATCH option is taken by default.

Example

Following is a basic example to show how to delete a cluster in JCL:

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEPNAME EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DELETE MY.VSAM.KSDSFILE CLUSTER
    PURGE
```

```
/*
```

If you will execute the above JCL on Mainframes server. It should execute with MAXCC=0 and it will delete MY.VSAM.KSDSFILE VSAM file.

4. VSAM – Entry Sequenced Data Set

ESDS is known as Entry Sequenced Data Set. An entry-sequenced data set behaves like sequential file organization with some more features included. We can access the records directly and for safety purpose we can use passwords also. We must code **NONINDEXED** inside the DEFINE CLUSTER command for ESDS datasets. Following are the key features of ESDS:

- Records in ESDS cluster are stored in the order in which they were inserted into the dataset.
- Records are referenced by physical address which is known as **Relative Byte Address (RBA)**. Suppose if in an ESDS dataset, we have 80 byte records, the RBA of first record will be 0, RBA for second record will be 80, for third record it will be 160 and so on.
- Records can be accessed sequentially by RBA which is known as **addressed access**.
- Records are held in the order in which they were inserted. New records are inserted at the end.
- Deletion of records is not possible in ESDS dataset. But they can be marked as inactive.
- Records in ESDS dataset can be of fixed length or variable length.
- ESDS is non-indexed. Keys are not present in ESDS dataset, so it may contain duplicate records.
- ESDS can be used in COBOL programs like any other file. We will specify the file name in JCL and we can use the ESDS file for processing inside program. In COBOL program specify file organization as **Sequential** and access mode as **Sequential** with ESDS dataset.

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>