

CHAPTER

1

The Virtual Storage Access Method (VSAM)

<i>Introduction</i>	3
<i>What is VSAM?</i>	4
<i>Access Methods</i>	4
<i>Access Methods and File Organization</i>	4
<i>Types of VSAM Data Sets</i>	5
<i>VSAM Record Structure and Organization</i>	8
<i>VSAM Data Access Types</i>	11
<i>Sequential Access</i>	11
<i>Direct Access</i>	11
<i>Keyed direct access</i>	12
<i>Addressed direct access</i>	12
<i>Keyed direct access with an alternate index</i>	12
<i>Skip Sequential Access</i>	14

Introduction

This chapter contains an overview of the Virtual Storage Access Method (VSAM). It presents basic VSAM terminology and concepts that you will need to process VSAM data sets. In particular, you will learn what access methods are and how they relate to data set organization. Then you will be introduced to the three types of access that VSAM provides to records in VSAM data sets. The access types are

- sequential
- direct
- skip sequential.

Each of these access types is fully described in this chapter.

You will learn about the three types of VSAM data sets. The organization of your VSAM data set determines how you can process it. The three types of data set organization are

- ESDS
- KSDS
- RRDS.

These data set types are introduced and discussed briefly in this chapter. Each data set type is described in detail in Chapter 5, “Processing an ESDS in a SAS Job,” on page 47, Chapter 6, “Processing a KSDS in a SAS Job,” on page 55, and Chapter 7, “Processing an RRDS in a SAS Job,” on page 73, respectively. The VSAM data set types are summarized in Table 1.1 on page 7.

In addition, five figures are presented to illustrate the concepts presented in this chapter. Figure 1.1 on page 7 demonstrates the organization of VSAM data sets. Figure

1.2 on page 9, Figure 1.3 on page 10, and Figure 1.4 on page 11 show the differences between an ESDS, a KSDS, and an RRDS. Finally, Figure 1.5 on page 13 illustrates a KSDS alternate index.

This chapter addresses the needs of the new VSAM user. If you are already familiar with the information that is presented here, you might want to proceed to Chapter 2, “SAS Options and Variables for VSAM Processing,” on page 15, which explains how to access VSAM data sets with the SAS System.

What is VSAM?

VSAM stands for Virtual Storage Access Method and is available under the CMS, OS/390, and VSE operating environments. Beginning with Version 7, VSE is not supported by the SAS System; therefore, VSE concerns are not described in this book.

VSAM is an IBM data access method that provides three ways to *organize* records in a disk data set. The three kinds of data set organization are

- ESDS (Entry- Sequenced Data Set)
- KSDS (Key-Sequenced Data Set)
- RRDS (Relative-Record Data Set).

VSAM allows three types of *access* to records in VSAM data sets. They are

- sequential
- direct
- skip sequential.

In addition, VSAM provides

- two direct access modes (addressed or keyed)
- two access entities (logical records and control intervals)
- two access directions (forward and backward)
- retrieval options (such as generic key and key greater-than-or-equal).

The SAS System supports all of these VSAM features, although not necessarily in all possible combinations. By specifying options in the INFILE statement in your SAS program, you can read, update, create, and erase records from VSAM data sets. See Table 3.1 on page 28 for a summary of the operations that the SAS System supports.

Access Methods

Access methods are software routines that control the data transfer between primary storage (main memory) and secondary storage devices. Secondary, or auxiliary storage, is independent of the computer’s memory, for example, storage on tape or disk. VSAM is designed specifically for use with disks. Because VSAM data set structure permits the use of both direct and sequential access types, you can select either the type or the combination of access types that best suits your specific application requirements.

Direct access means that you have the ability to read any data record in a data set directly, without reading preceding records in the data set. The terms *direct* and *random* are sometimes used interchangeably when referring to data organization, access methods, and storage devices. In this document, only direct is used, but you may find that random is used in other literature.

Access Methods and File Organization

Data stored on IBM disks can be organized in a number of ways, which are referred to as data set types. IBM software supports the following data set types:

- Physical Sequential (PS)
- Partitioned Organization (PO: OS/390 operating system only)
- Indexed Sequential (IS)
- Direct Access Organization (DA)
- Virtual Storage Access Method (VSAM).

VSAM data sets can be one of the following:

- Entry-Sequenced Data Set (ESDS)
- Key-Sequenced Data Set (KSDS)
- Relative Record Data Set (RRDS).

In each data set type except VSAM, the records are organized in a unique way, depending on their purpose. Each kind of data set organization has one or more special access methods. For example, a data set that is organized in a direct manner, that is, DA organization, is characterized by a predictable relationship between the key of a record and the address of that record on a DASD device. The programmer establishes this relationship and must supply most of the logic required to locate the individual records.

VSAM is different from the other data set types because it provides a functional equivalent for most of these data set organizations, and it is a multifunction, all-purpose access method. For instance,

- ESDS organization is the functional equivalent of physical sequential organization (PS).
- KSDS organization is the functional equivalent of indexed sequential organization (IS).
- RRDS organization is the functional equivalent of direct access organization (DA).

The kinds of data set organizations that you access with VSAM differ from some others in that they are device independent from the user's viewpoint and can be both sequentially and directly accessed. You access a record by addressing the record in terms of its displacement (in bytes) from the beginning of the data set, by its key, or by its record number.

The root of the VSAM access method is the VSAM catalog, which is a disk area for defining data sets and disk space and maintaining information about each VSAM data set. VSAM catalogs and data sets are created and managed with IBM Access Method Services (AMS), a multifunction service program.

Types of VSAM Data Sets

There are three types of VSAM data sets. The main difference between the three data set types is the logical order in which data records are arranged in the data set. The following is a description of each type of VSAM data set:

ESDS Entry-Sequenced Data Set — The record sequence is determined by the order in which the records are entered into the data set, without respect to the record contents. New records are stored at the end of the data set.

An ESDS is appropriate for applications that do not require any particular ordering of the data by the record contents or for those that require time-ordered data. Applications that use a log or journal are well-suited for an ESDS data set structure.

KSDS Key-Sequenced Data Set — The record sequence is determined by a key containing a unique value, such as an employee, invoice, or

transaction number. The key is a contiguous portion of the record and is defined when the data set is created. The record order is defined by the EBCDIC collating sequence of the key field contents.

A KSDS is always defined with a prime index that relates the record's key value to its relative location in the data set. VSAM uses the index to locate a record for retrieval and to locate a collating position for record insertion.

A KSDS is the most flexible approach for most applications since the record can be accessed directly, via the key field. Access is not dependent on the physical location of the record in the data set.

RRDS

Relative-Record Data Set — The data set is a string of fixed-length slots, each identified by a relative record number (RRN). Each slot can either contain a record or be empty. Records are stored and retrieved by the relative record number of the slot.

An RRDS is appropriate for many applications using fixed-length records or when the record number has a contextual meaning that can be used as a key.

Figure 1.1 on page 7 shows how the three types of VSAM data sets are organized.

When a VSAM data set is created, it is defined in a cluster. A cluster encompasses the components of a VSAM data set. ESDS and RRDS clusters have only a data component. A KSDS cluster has a data component and an index component. The index relates each record's key to its location in the data set. VSAM uses the index to sequence and locate the records of a KSDS.

Table 1.1 on page 7 summarizes the differences between the three VSAM data set types.

Figure 1.1 VSAM Data Set Organization: Data Components and Index Components

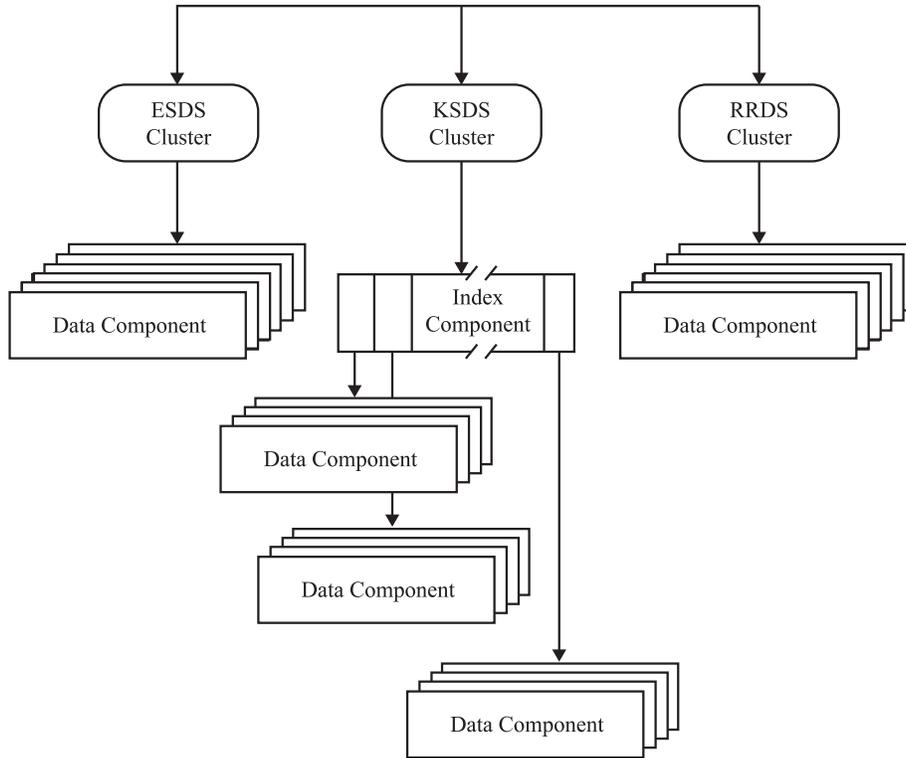


Table 1.1 Comparison of VSAM Data Set Types

	ESDS	KSDS	RRDS
Sequential access is by	entry order	primary key order	RRN**
Direct access is by	RBA**	key RBA	RRN
Record format can be	fixed variable spanned	fixed variable spanned	fixed
Changeable record length	no	yes	no
New records added to	end of file	anywhere	RRN slot (if empty)
Embedded free space defined*	no	yes	no
Delete records and reuse space	no***	yes	yes
Access through alternate index	yes	yes	no

	ESDS	KSDS	RRDS
REUSE the file	yes (if no AIX**)	yes (if no AIX)	yes
Can RBA or RRN change	no	yes	no

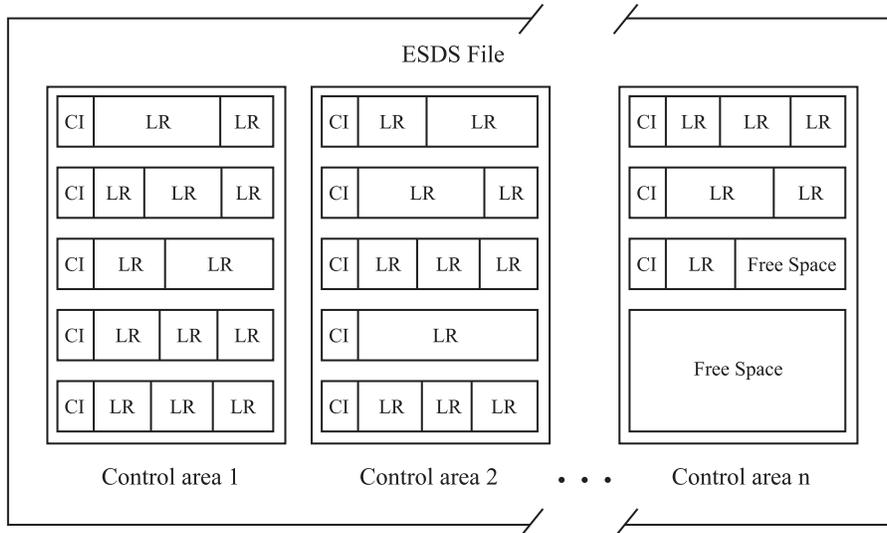
- * You can insert records and change their lengths.
- ** RRN= relative record number, RBA= relative byte address, and AIX= alternative index
- *** You can, however, overlay a record if the length does not change.

VSAM Record Structure and Organization

Records in VSAM data sets are grouped into control intervals, the unit of data transfer between main storage and secondary disk storage. *Control intervals* are continuous areas of direct access storage that VSAM uses for storing records and control information describing them. Although the size of control intervals varies from one data set to another, the size within a data set is fixed, either by VSAM or by the user (within VSAM-imposed restrictions). If VSAM chooses the size, it does so based on the DASD type, record size, and smallest amount of virtual storage space that user applications make available for I/O buffers. A spanned record is one that exceeds the established control interval size by spanning one or more control interval boundaries. Spanned records are permitted in an ESDS and a KSDS but not in an RRDS.

Control intervals are grouped into control areas. Control areas are the units of a data set that VSAM preformats as records are added to the data set. VSAM fixes the number of control intervals for each control area. (Refer to Figure 1.2 on page 9, Figure 1.3 on page 10, and Figure 1.4 on page 11 for a depiction of the control interval format used by each of the data set types.) In addition, KSDS control areas are used for distributing free space throughout the data set, as a percent of control intervals per control area.

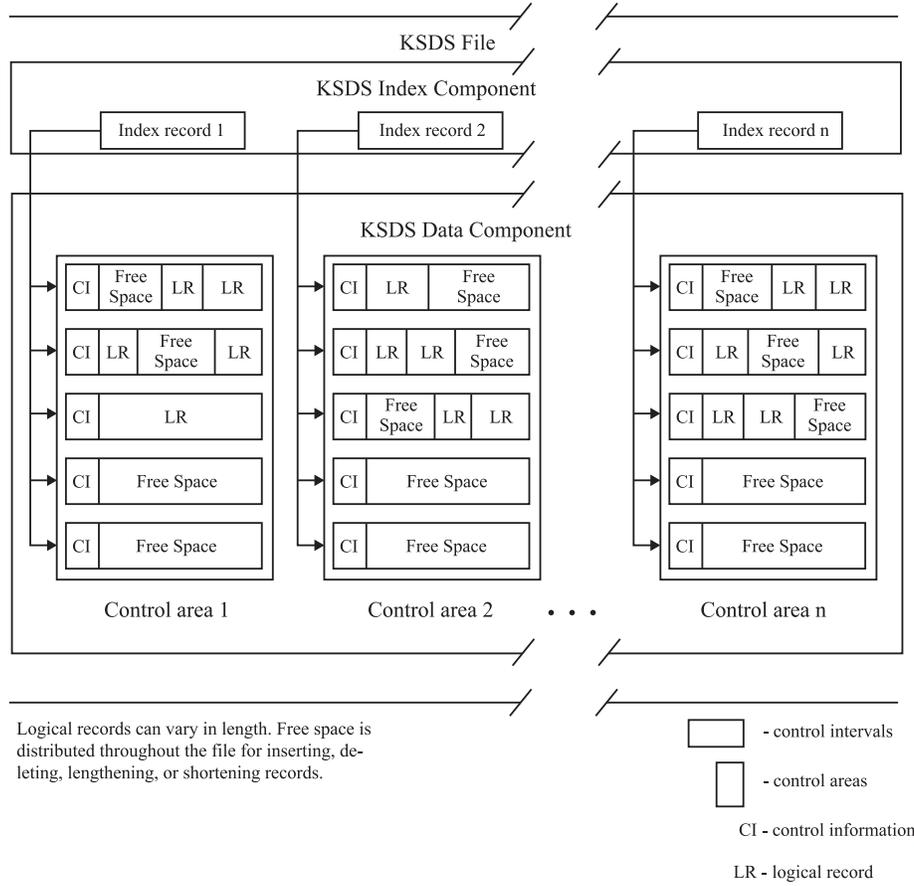
Figure 1.2 ESDS Control Intervals and Control Areas



Logical records can vary in length, but the length cannot change once the record is written. All free space is at the end of the file.

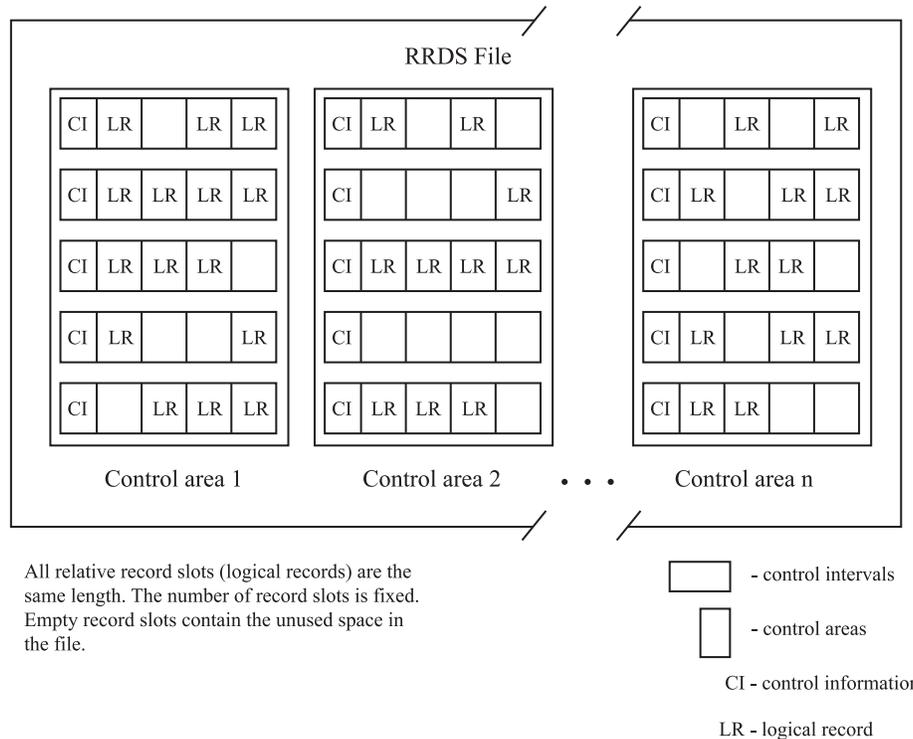
-  - control intervals
-  - control areas
- CI - control information
- LR - logical record

Figure 1.3 KSDS Control Intervals and Control Areas



Logical records can vary in length. Free space is distributed throughout the file for inserting, deleting, lengthening, or shortening records.

Figure 1.4 RRDS Control Intervals and Control Areas



VSAM Data Access Types

Access to VSAM data sets can be either sequential, direct (random), or a combination of both. Records can be retrieved in either a forward or backward direction.

Sequential Access

In sequential access, a series of records is retrieved in sequence. Sequence has a different meaning for each of the three VSAM data set organizations.

- In an ESDS, sequential access means that a record is retrieved by its entry sequence.
- In a KSDS, sequential access means that a record is retrieved by its key sequence.
- In an RRDS, sequential access means that a record is retrieved by its relative record sequence.

In all three cases, a record is located by its position relative to the last record accessed.

Direct Access

With direct access, data storage or retrieval depends only on the location of the record and not on a reference to records previously accessed. Each record is stored or retrieved *directly*, according to its logical address (its key or relative record number, RRN) or its address relative to the beginning of the data set (relative byte address, or

RBA). Thus, there are two direct access modes: *keyed* by key or relative record number, and *addressed* by relative byte address.

Keyed direct access

In keyed direct access, records are retrieved or stored by either

- an index that relates the record's key to its relative location in the data set
- a relative record number (RRN) that identifies the record wanted. The RRN is relative to the first record in the data set.

The SAS System supports keyed access to logical records in both KSDS and RRDS data sets. Keyed access to data records in KSDS data sets is by key; in RRDS data sets, keyed access is by the relative record number.

Addressed direct access

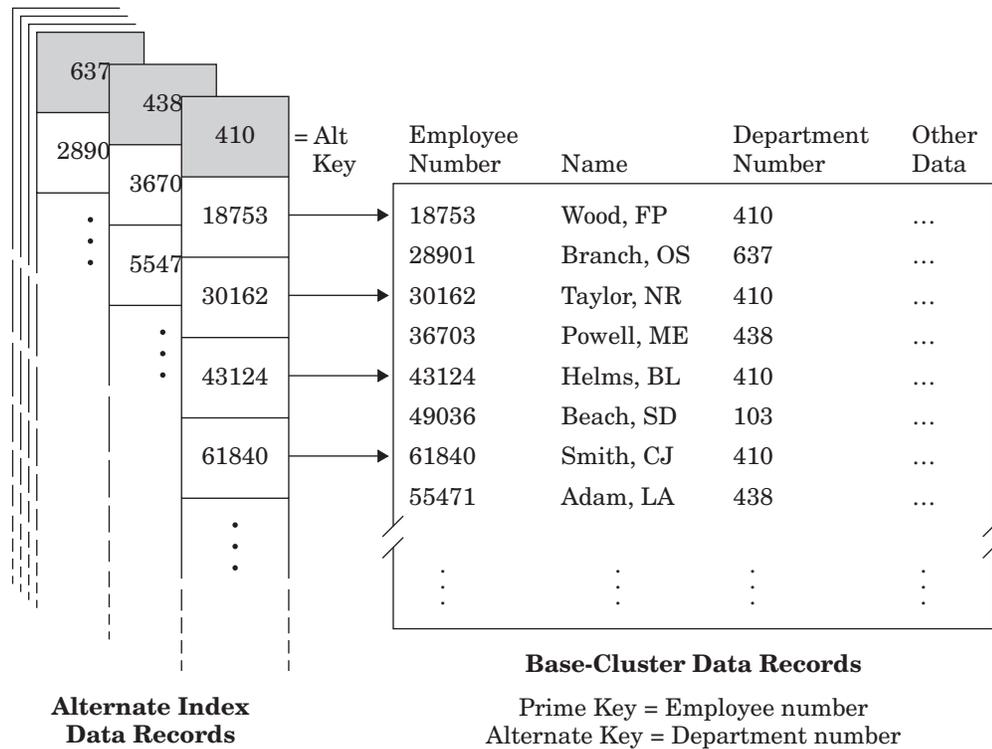
In addressed direct access, the entire data set is treated as a continuous stream of bytes. A record is retrieved and stored directly by its address relative to the beginning of the data set (relative byte address, or RBA), which is dependent on the record's location relative to records previously accessed. The SAS System supports addressed access to logical records in ESDS and KSDS data sets. It also supports addressed access (read only) to control intervals in all three data set types.

Keyed direct access with an alternate index

An alternate key index, commonly called an alternate index (AIX), provides another way to access a VSAM data set. The advantage of an alternate index is that you effectively reorganize the data set instead of keeping separate copies organized in different ways for different applications. For example, suppose you have a KSDS with the employee number as the prime key. By building alternate indexes using employee names and department numbers, you can access the same data set in three ways: by employee name, by employee number, or by department number. The alternate key does not have to be unique; that is, there can be more than one record with the same alternate key.

Figure 1.5 on page 13 illustrates an alternate index with nonunique keys over a KSDS. The base cluster records are sequenced by employee number, which is the prime key. The alternate index records are sequenced by department number, which is the alternate key. Each alternate index data record points to the prime key (employee number) in the base cluster. Note that because the alternate keys are nonunique there can be multiple base records with the same department number.

Figure 1.5 KSDS with Alternate Index (Nonunique Alternate Keys)



Alternate indexes can be built over a KSDS or an ESDS. You define and build an alternate index using the IBM utility program Access Method Services (AMS). The data set over which an alternate index is built is called the base cluster. The alternate key can be any field having a fixed length and a fixed position within each record. The alternate index itself is a KSDS. The data component of an alternate index contains the alternate key, followed by a pointer to the appropriate record(s) in the base cluster. In a KSDS, the pointer is the prime key; in an ESDS, the pointer is the RBA of the base record(s).

A path logically relates a base cluster and one of its alternate indexes. You define and name a path to access the base cluster records through a specific alternate index with AMS. Refer to Chapter 8, "Using Alternate Indexes for VSAM Data Sets," on page 87 and the appropriate IBM documentation for more information on defining alternate indexes.

Skip Sequential Access

A combination of both direct and sequential access can be used in a two-step process called skip sequential access. Use keyed direct access to find a starting point. Once the initial record is obtained, additional records are retrieved sequentially. Skip sequential processing can be used with a KSDS, RRDS, and, if it has an alternate index, an ESDS.

Skip sequential processing is useful when

- performance is important. Processing skip sequentially can improve performance and reduce overhead since a simple sequential retrieval is faster than direct retrieval.
- you know the key, RBA, or RRN of the first record you want but do not know the key, RBA, or RRN of the subsequent records.
- you want to process the data set sequentially, starting at a record other than the first.