VSAM is used to organize records into four types of data sets: Key-sequenced (KSDS), entry-sequenced (ESDS), linear (LDS), and relative record (RRQS and VRDQS). This issue focuses on LDS.

- Linear data set (LDS) organization is specified with the IDCAMS DEFINE command by using the LINEAR parameter (see illustration below).
- A linear data set contains data that can be accessed as byte-addressable strings in virtual storage.
- It is a VSAM data set with a control interval size multiple of 4096 bytes (to 32768 bytes in increments of 4096 bytes).
- An LDS has no embedded control information in its CI, that is, no RDFS and CIRDF.
- All LDS are data bytes where logical records may be blocked and deblocked by the program.
- Logical records are not apparent from VSAM's point of view.

### LDS Overview

**Control Intervals (CIs)**

- A CI is a contiguous area of DASD volume track that VSAM uses to store data logical records and control information that describes the records in the CI. A CI is the unit of information that VSAM transfers between the DASD device and the central storage during one I/O operation. If the CI is formed by several physical blocks, these blocks are read or written in a single I/O operation (with several Read or Write CCWs). Whenever a logical record is retrieved from a DASD device, the entire CI containing the record is read into a VSAM I/O buffer in virtual storage. The logical record is then transferred from the VSAM buffer to a user-defined logical record buffer or work area.

### LDS Parameters

- **LINEAR**
- **DIV**
- **Data In Virtual (DIV)**

#### Data Virtualization

- VSAM data sets may be accessed by 'control interval' access, but this is used only for very specific applications.

### Control Interval Data Descriptions

- A CI is formed by two or more CIs put together into fixed-length contiguous areas of direct-access storage. A VSAM data set is composed of an integer number of CIs. In most cases, a CI is the size of a 3390 cylinder (15 tracks).
- The minimum size of a CI is one track. The maximum size of a CI is 16 tracks when the data set is stripped. The CA size is implicitly defined when you specify the size of a data set at definition time. There is no keyword to set the CA size.

### Linear Data Sets

- Linear data sets are considered to be clusters without index components.
- If a linear data set is not an integer multiple of 4096, the control interval size is rounded up to the next 4096 increment.

### Additional Examples - Implementations of VSAM Linear Data Sets:

- For z/OS UNIX use, the VSAM data set must be linear.
- When a linear data set is defined, the catalog forces the block size to 4096 bytes unless specified differently on VSAM DEFINE.

### VSAM Linear Data Sets

#### Linear Data Set Parameters

- You can access a linear data set using these techniques:
  - VSAM - DIV, if the control interval size is 4096 bytes. The data-in-virtual (DIV) macro provides access to VSAM linear data sets.
  - Window services, if the control interval size is 4096 bytes.

#### Alternate Indexes

- Alternate indexes are not supported for linear data sets.

- Linear data sets are considered to be clusters without index components.
- To be consistent with other VSAM data sets, cluster names are used for processing.

### LDS Data In Virtual (DIV)

- LDS is the VSAM data set organization that is used by Data-in-Virtual (DIV) facility.
- DIV is an optional and unique buffering technique that is used for LDS data sets only.
- Data is read into central storage through the paging mechanism only when a DIV 4-KB data block is referenced.

### VSAM Linear Data Sets

- During Real Storage Management (a z/OS component) page stealing processing, only changed pages are written to auxiliary storage.
- Unchanged pages are discarded because they can be retrieved again from the permanent linear data set.
- DIV is designed to improve application performance that process large data sets non-sequentially in an unpredictable pattern.

### LDS Parameters

- NOTE: It reduces the number of I/O operations that are traditionally associated with data retrieval where logically candidates are large arrays and table data sets.

#### Additional Parameters

- LDS-in-virtual is most useful for applications, such as graphics, that require large amounts of data but normally reference only small portions of that data at any given time.
- It requires that the source of the object be a VSAM linear data set on DASD (a permanent object) or a hyperspace, (a temporary object).
- Data-in-virtual is also useful for applications that require small amounts of data; data-in-virtual simplifies the way you access data by avoiding the complexities of access methods.

#### Additional Examples

- You can map a data-in-virtual object in either an address space, a data space, or a hyperspace.
- Mapping the object into a data space or hyperspace provides additional storage for the data; the size of the window is no longer restricted to the space available in an address space.
- It provides additional isolation and integrity for the data, as well as more direct methods of sharing access to that data.
- Data-in-virtual is most useful for applications, such as graphics, that require large amounts of data but normally reference only small portions of that data at any given time.

### LDS Parameters

- NOTE: CI mode processing is not permitted when accessing a ‘compressed’ data set.