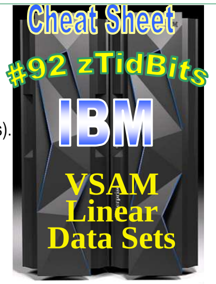
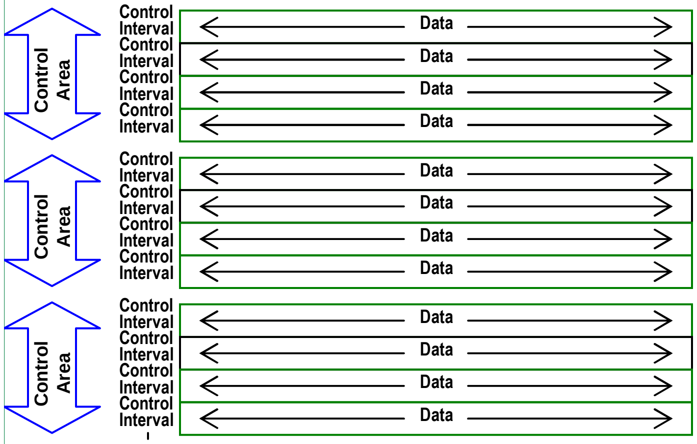


- VSAM is used to organize records into four types of data sets: Key-sequenced (KSDS), entry-sequenced (ESDS), linear (**LDS**), and relative record (RRDS and VRRDS). This issue focuses on LINEAR.
  - The difference between the types of VSAM data sets is the way that their records are stored and accessed.
  - The 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[1] and CIDFs[2].
  - All LDS bytes are data bytes where logical records must be blocked and deblocked by the program.
  - Logical records are not apparent from VSAM's point of view.

    NOTE: In a sense, an LDS is a non-VSAM data set with some of the VSAM facilities, such as the use of IDCAMS and VSAM specific information in the catalog.
- A linear data set is processed as an entry-sequenced data set, with certain restrictions.
  - All types of VSAM data sets, including linear, can be accessed by 'control interval' access, but this is used only for very specific applications.

    NOTE: CI mode processing is not permitted when accessing a "compressed" data set.



Like the ESDS and RRDS, LDS contains a data component only.

VSAM meta data descriptions

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.

A CA 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 CAs. In most cases, a CA is the size of a 3390 cylinder (15 tracks). The minimum size of a CA is one track. The maximum size of a CA 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 are PRINT*ed* in physical sequential order.

- Linear data sets being extended format are eligible for striped data.
- Linear data sets are considered 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.

  NOTE: The system chooses the best physical record size to use for the track size geometry. For an example, if you specify CISIZE(16384), the block size is set to 16,384. If the specified BUFFERSPACE is greater than 8192 bytes, it is decremented to a multiple of 4096. If the BUFFERSPACE is less than 8192, access method services issues a message and fails the command.
- Linear data sets cannot be processed at the record level and you must use the DIV macro to process this data set type under native programming practices.

```
//IDCAMS JOB (ACCT),'CREATE LINEAR VSAM',
//   MSGCLASS=A,CLASS=A,TIME=30,
//   NOTIFY=&SYSUID,MSGLEVEL=(1,1)
//VSMLNR01 EXEC PGM=IDCAMS
//SYSIN DD *
  DEFINE CLUSTER (NAME(KETTNER.EXAMPLE1.LINEAR) -
      TRACKS(5) -
      CISZ(8192) -
      SHAREOPTIONS(1,3) -
      LINEAR)
//SYSPRINT DD SYSOUT=*
```

Illustration VSAM Linear Data Set Definition

[1] Record Descriptor Field describes record length and number of records contained in a control interval.

[2] Control Interval Descriptor Field describes the displacement and amount of free space available for record insertions.

An **access method is** an optional function to simplify the logic of the application when it requests I/O operations to z/OS. An application in EXCP mode does not require an access method to help run I/O operations. An access method is implemented through a set of programs that belong to DFSMS that are in PLPA or in the SMSVSAM private address space. The access method runs under the application task that starts it through a branch instruction (no PSW change status).

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 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.

The CYLINDERS, TRACKS, MEGABYTES, KILOBYTES, and RECORDS parameters are permitted for linear data sets.

ALTER can be used to change an entry-sequenced data set, with the proper attributes, to a linear data set

The REPRO command can be used to load data from one linear data set into another linear data set.

The RECORDSIZE parameter is not used for a linear data set.

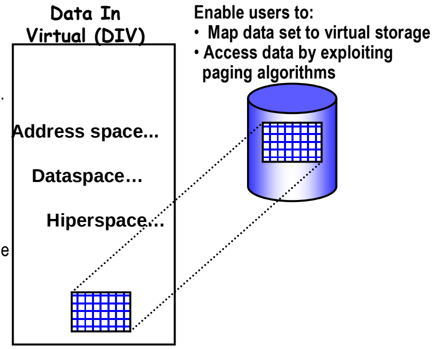
You can access a native linear data set with VSAM, the DIV macro, or window services.

To update a native linear data set using VSAM, you must use control interval access, and must have 'control' authority. To read a native linear data set with VSAM, you must use control interval access, and have read authority.

- 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.

    NOTE: An LDS data set is some times called a DIV object.
  - Data is read into central storage through the paging mechanism only when a DIV 4-KB data block is referenced.
  - During Real Storage Management (a z/OS component) page steal 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.

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



Enable users to:
- **Map data set to virtual storage**
- **Access data by exploiting paging algorithms**

- Data-in-virtual enables you to map data into virtual storage but deal only with the portion of it that you need.
- The DIV macro provides the system services that manage the data object enabling you to map the object into virtual storage, create a window, and "view" through that window only the portion of the data object that is needed, (The system brings into central storage only the data that you actually reference).
- You can map a data-in-virtual object in either an address space, a data space, or a hiperspace.
  - Mapping the object into a data space or hiperspace 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.
    - It requires that the source of the object be a VSAM linear data set on DASD (a permanent object) or a hiperspace, (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 - 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. .
- zFS always reads and writes an 8K (two CIs) at a time.
  - zFS can write mulltiple small files into an 8K block by writing fragments (fragments are always 1K bytes).
  - Two 1K files could be contained in a single 8K block.
- A z/OS Unix Systems Services zFS aggregate is a data set that contains zFS files containing a VSAM Linear Data Set and is a container that houses one or more zFS file systems.
  - An aggregate can only have one VSAM LDS, but contain an unlimited number of file systems.

  NOTE: The name of the aggregate is the same as the VSAM LDS.
  - A zFS aggregate can contain up to 4GB 1K blocks for a maximum of 4 TBs.
  - Sufficient space must be available on the volume or volumes, as multiple volumes may be specified on the DEFINE of the VSAM LDS.

    NOTE: DFSMS decides when to allocate on the volumes during any extension of a primary allocation.
  - VSAM LDS greater than 4GBs may be specified by using the extended format (EF) and the extended addressability (EA) capability in the data class of the data set.
  - You cannot assign more than one VSAM LDS per aggregate.
- **LDS's are used with z/OS' DB2, (the most popular implementation of LDSs)**.
- DB2 uses linear data sets (LDS) for its table spaces without implementing Data-in-Virtual.
  - All the control (including buffer pool) is done by DB2. For example, DB2 implements data striping in LDS data sets.
- Page Sets are used to store DB2 data and a file page set contains data records which is the physical representation of a table space.
- A file page set can be *nonpartitioned*, *partitioned*, or universal.
- A linear addressing range is a page set which is a collection of one or more data sets logically concatenated.
- DB2 data sets are defined as VSAM linear data sets (LDSs).
  - Because of the format incompatibilities between DB2 and VSAM records, they cannot be read or written by VSAM record processing.
  - The data in LDSs can be accessed by services which use VSAM control interval (CI) processing such as the access method services commands IMPORT and EXPORT.
- The access method services (IDCAMS) commands PRINT and REPRO DB2 data within LDSs can also be used through services which support the LDS type.
- DB2 data sets support VSAM CI mode processing therefore, z/OS' Data Facility Hierarchical Storage Manager (DFHSM) can be used.
- A nonpartitioned page set, excluding a nonpartitioning index on a partitioned table space that's defined as LARGE or defined with the DSSIZE parameter consists of between 1 and 32 VSAM LDSs.
- A nonpartitioning index on a partitioned table space defined as LARGE or defined with the DSSIZE parameter can have up to *maxnumpart* VSAM LDSs, where maxnumpart is the maximum number of partitions allowed for the table space.
- The VSAM LDSs are concatenated to form a single DASD space, or addressing range.

DB2 striping is unrelated to VSAM striping.

DB2 uses linear data sets for its table spaces without implementing Data-In-Virtual (DIV).