TATA CONSULTANCY SERVICES

TATA

z/OS – VSAM Concepts

Session Outline

- Introduction to VSAM
- Structure of VSAM Datasets
- Access Method Services
- Demo/Exercise Discussion

Objectives

- Familiarization with
  - VSAM concepts
  - Space Allocation and types of data sets supported by VSAM
  - VSAM Characteristics
- Identify the general uses for Access Method Services

Introduction to VSAM
(Virtual Storage Access Method)

What is VSAM?

- VSAM is a high-performance access method used in OS/390, MVS/ESA and z/OS operating systems
- Used as a Dataset in conjunction with programming languages (COBOL or PL/I) to create Applications
- Used by CICS to store and retrieve data
- Not a database management system – does not provide a relationship among data (such as DB2 or IMS)

VSAM Characteristics

- VSAM supports three types of data access: Sequential, Random (also called Direct access) and Skip Sequential
- Protection of data against unauthorized access is an inherent part of VSAM
- Easily portable to AS/400, the PC or non-IBM computers (cross system compatibility)
- A format for storing data independently of the type of direct access storage device on which it is stored

VSAM Characteristics (Contd.)

- Options for optimizing performance
- A multifunction service program (Access Method Services - IDCAMS) for setting up catalog records and maintaining data sets
- Transactional VSAM - allows VSAM data set sharing in batch/online and batch/batch environments

VSAM Dataset Organizations

- ESDS Entry Sequenced Data Set
- KSDS Key Sequenced Data Set
- RRDS (Fixed Length) Relative Record Data Set
- VRRDS Variable Length Relative Record Data Set
- LDS Linear Data Set

VSAM Dataset Organization

VSAM Terminology

- Catalog Management
- Logical Record
- Key Field
- Physical Record
- Cluster
- Sphere
- Index component
- Record management
- Data component
- Master/User Catalog
- Data Space
- Control Interval and Control Area
- Alternate Indexes
- Spanned Records

Catalog Management

- VSAM maintains extensive information about data sets and direct access storage space in an integrated catalog facility (ICF) catalog.
- The catalog's collection of information about a data set defines that data set's characteristics. All VSAM files

  must be defined in an ICF catalog.

Logical record

- Unit of information used to store data in a VSAM data set.
- Set of bytes containing a logical description of an item processed by an application program.
- A logical record can be of a fixed size or a variable size depending on the business requirements.

Key field

- Identifies the item associated with the logical record.
- Important Field whose contents can be used to retrieve the specific logical record

Physical record

- The data is usually a set of logical records (packed together) transferred from or to memory by just one Read or Write CCW. The access method uses the BLKSIZE parameter to determine the length of the physical record. A large block size results in fewer gaps in the track but larger buffer in memory to keep the data.

- A physical record is device dependent, its size is calculated at the time the data set is defined.

- All physical records have the same length.

Cluster

- Collection of physical datasets that make up one logical data set.
- Consists of Data Component or Data Component & Index Component

Sphere

- A sphere is a base VSAM cluster and its associated clusters.
- These associated clusters are the alternate indexes (AIXs) of the base cluster.

Data and Index Component

- A component is an individual part of a VSAM data set. Each component has a name, an entry in the catalog and an entry in the VTOC.
- The KSDS and VRRDS organizations have data and index components. ESDS, RRDS and LDS organizations only have data components.
- The data component is the part of a VSAM data set, alternate index, or catalog that contains the data records.
- Using the Index component VSAM is able to randomly retrieve a record from the data component when a request is made for a record with a certain key. The key determines the record's position in the data set.

Control Interval and Control Area

- It is the fundamental building block of every VSAM file.A CI is a contiguous area of direct access storage that VSAM uses to store data records and control information that describes the records.
- A CI is the unit of information that VSAM transfers between the storage device and the process or during one I/O operation. Whenever a record is retrieved from direct access storage, the entire CI containing the record is read into a VSAM I/O buffer in virtual storage. The desired record is transferred from the VSAM buffer to a user-defined 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 one or more CAs. CAs are needed to implement the concept of splits. The size of a VSAM file is always a multiple of its CA. VSAM files are extended in units of CAs. A spanned record cannot be larger than a CA.

Alternate Indexes

- Alternate indexes (AIXs) allows logical records of a KSDS or of an ESDS to be accessed sequentially and directly by more than one key field.
- AIXs eliminate the need to store the same data in different sequences in multiple data sets for the purposes of various applications. Each alternate index is a KSDS cluster consisting of an index component and a data component.

Spanned Records

- Spanned records are needed when the application requires very long logical records. A spanned record may be the data component of an AIX cluster. If spanned records are used for KSDS, the primary key must be within the first control interval.
- Spanned records are logical records that are larger than the CI size. To have spanned records, the file must be defined with the SPANNED attribute at the time it is created. Spanned records are allowed to extend across or span control interval boundaries.
- A spanned record must always begin on a control interval boundary and fills one or more control intervals within a single control area. A spanned record cannot share the CI with any other records.

Control Intervals and Control Areas

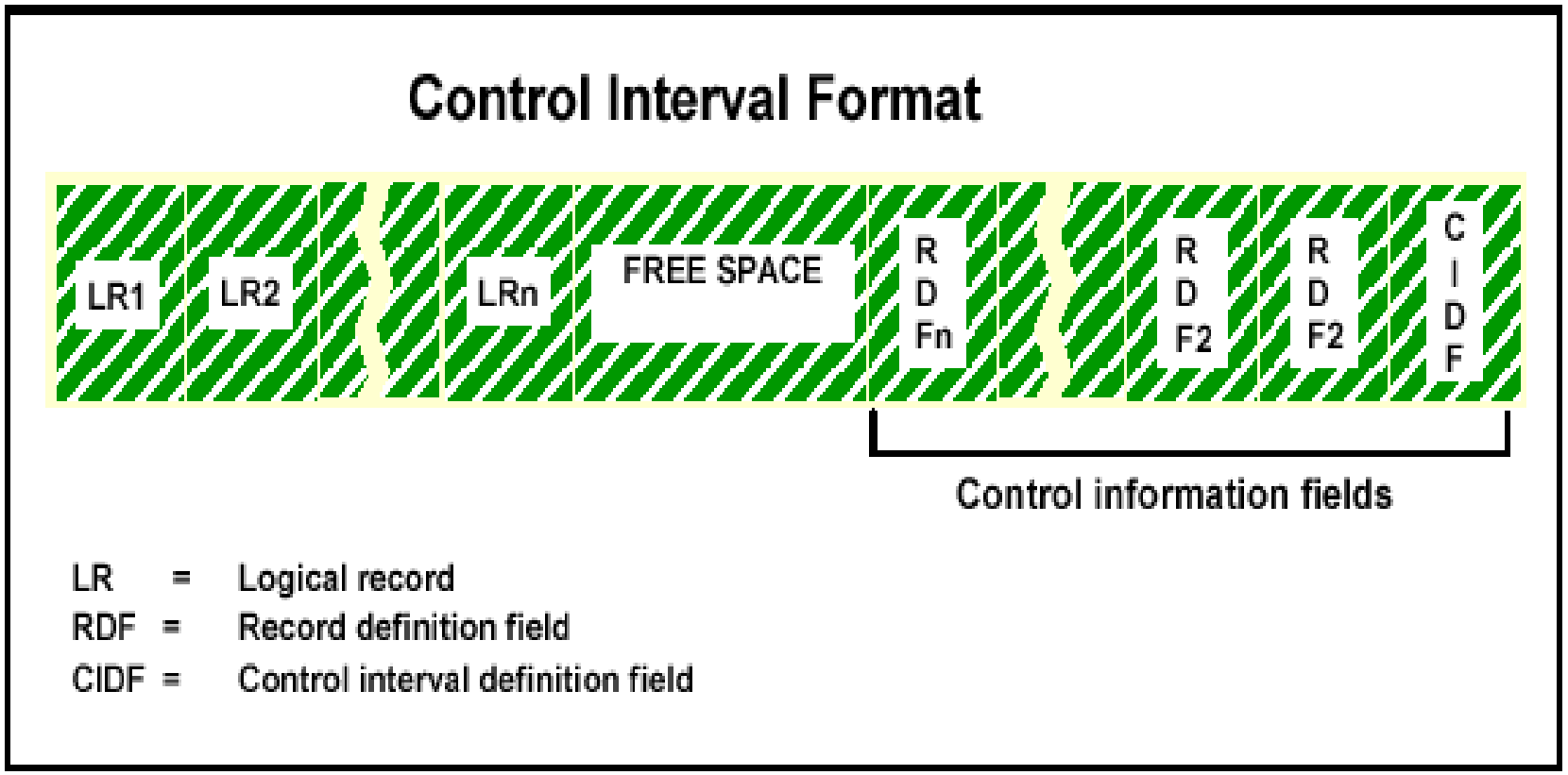
Control Interval

- VSAM Basic unit of work – the minimum amount of data that is transferred via buffers, between DASD and Main Memory
- VSAM determines the size of the physical Record (BLOCK) based on CI Size
- The CI size can be from 512 byes to 32 KB

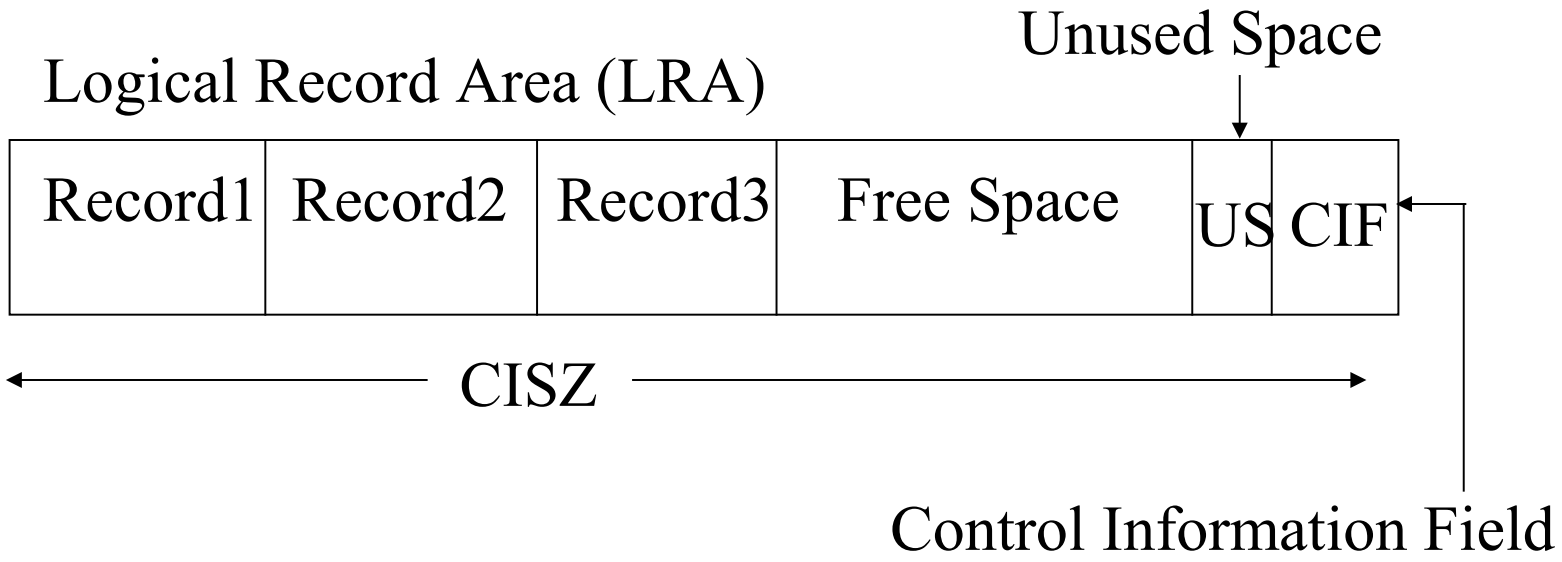
Control Interval & Control Area

- CA is a fixed length area of auxiliary storage space in which VSAM stores records
- Logical records of a VSAM dataset are grouped into a number of CIs.
- CIs of a dataset are grouped into one or more CAs
- The size of a VSAM file is always a multiple of its CA

General format of a Control Interval



Format of Data Control Interval



Control Information Field of a CI

CIF consists of two subfields:

- Control Information Definition Field (CIDF)
  - It is a 4 byte field
  - Contains the Offset and Amount of Free space

- Record Definition Field (RDF)
  - It is a 3-byte field.
  - Length of records (based on the spread of data)
  - Number of adjacent records that are of same length

Control Information Fields in a CI

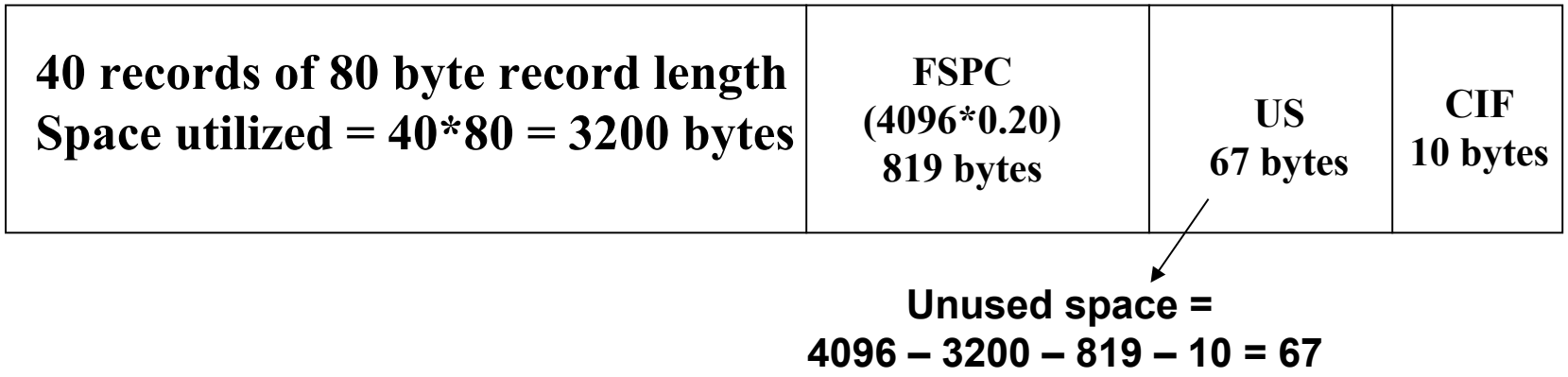
Record Type	Number of Records/CI	Control Information (bytes)
Fixed Length	Multiple	10
Fixed Length	1	7
Variable Length	Depends on the spread of records	4 (CIDF) and 3 (RDF)

CISZ (CI Size) – an example

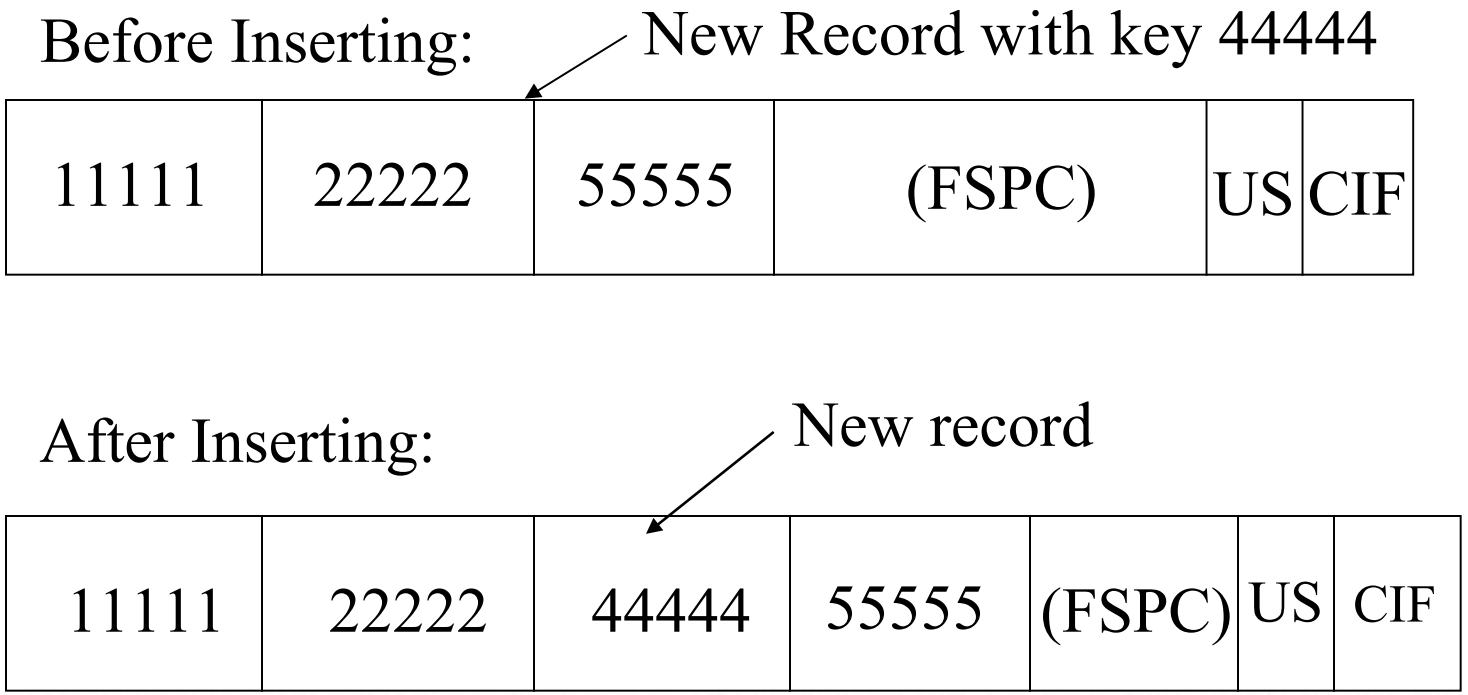
- How many 80 byte Fixed Length recs can be fit in a CISZ(4096) of Freespace(20,10) ?

Solution:

Logical Record Area (LRA)

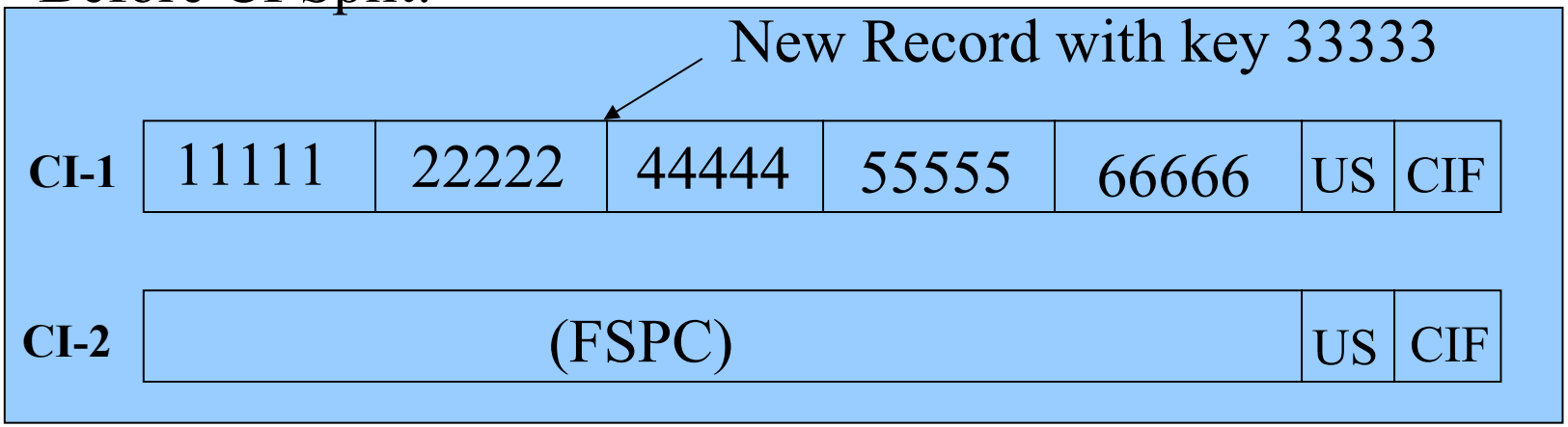


Inserting a record in a CI

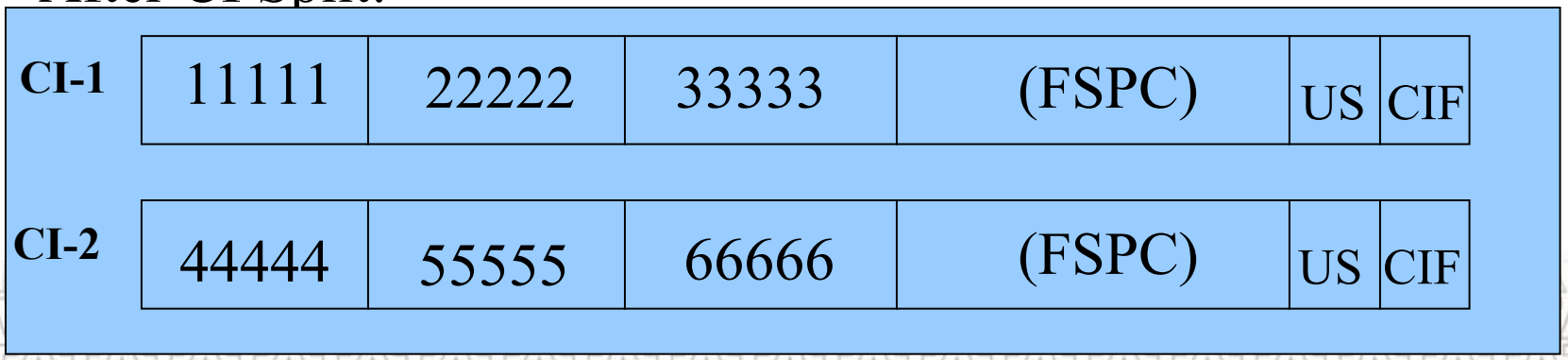


Control Interval Split

Before CI Split:



After CI Split:

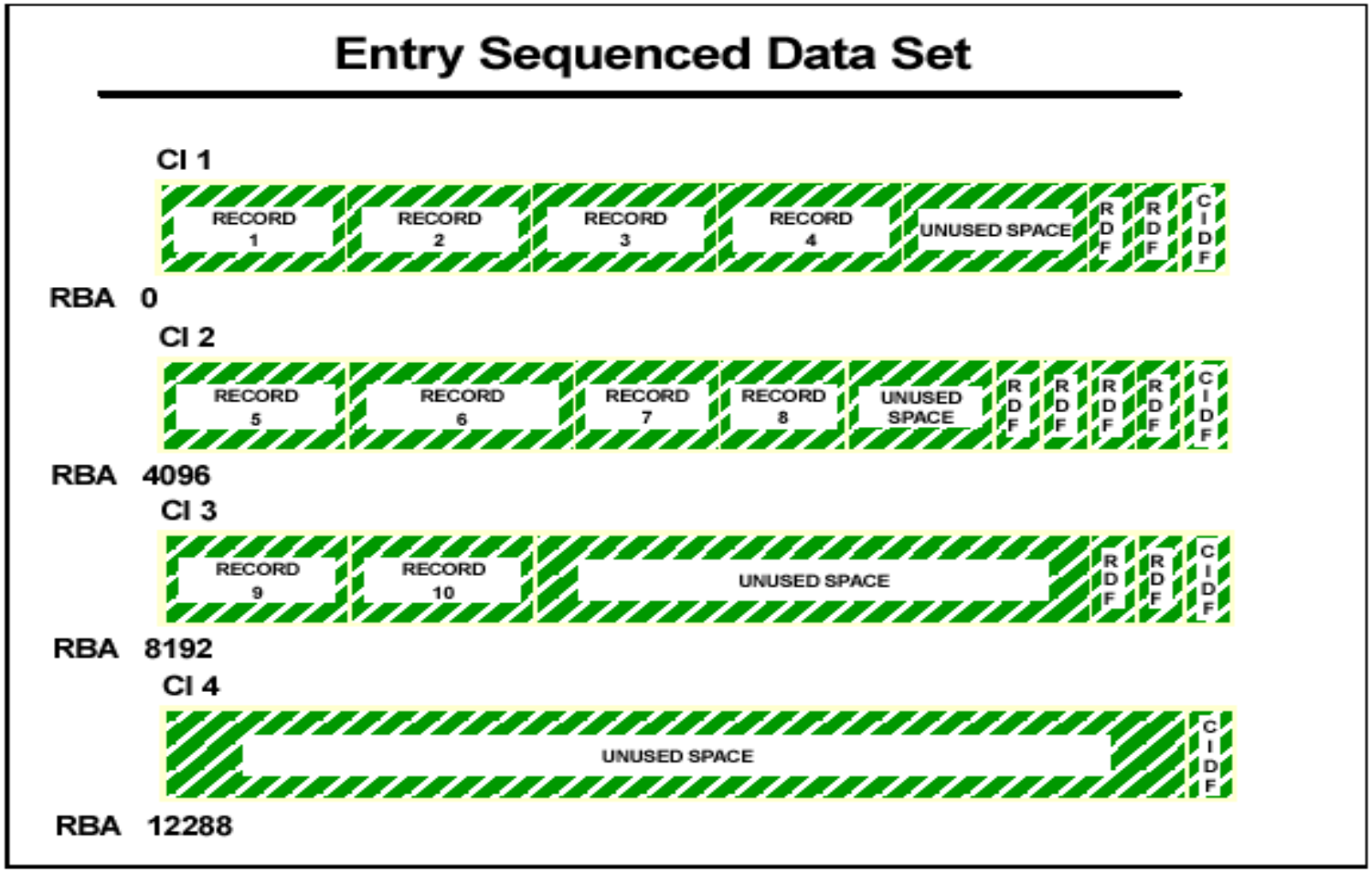


Types of VSAM Datasets

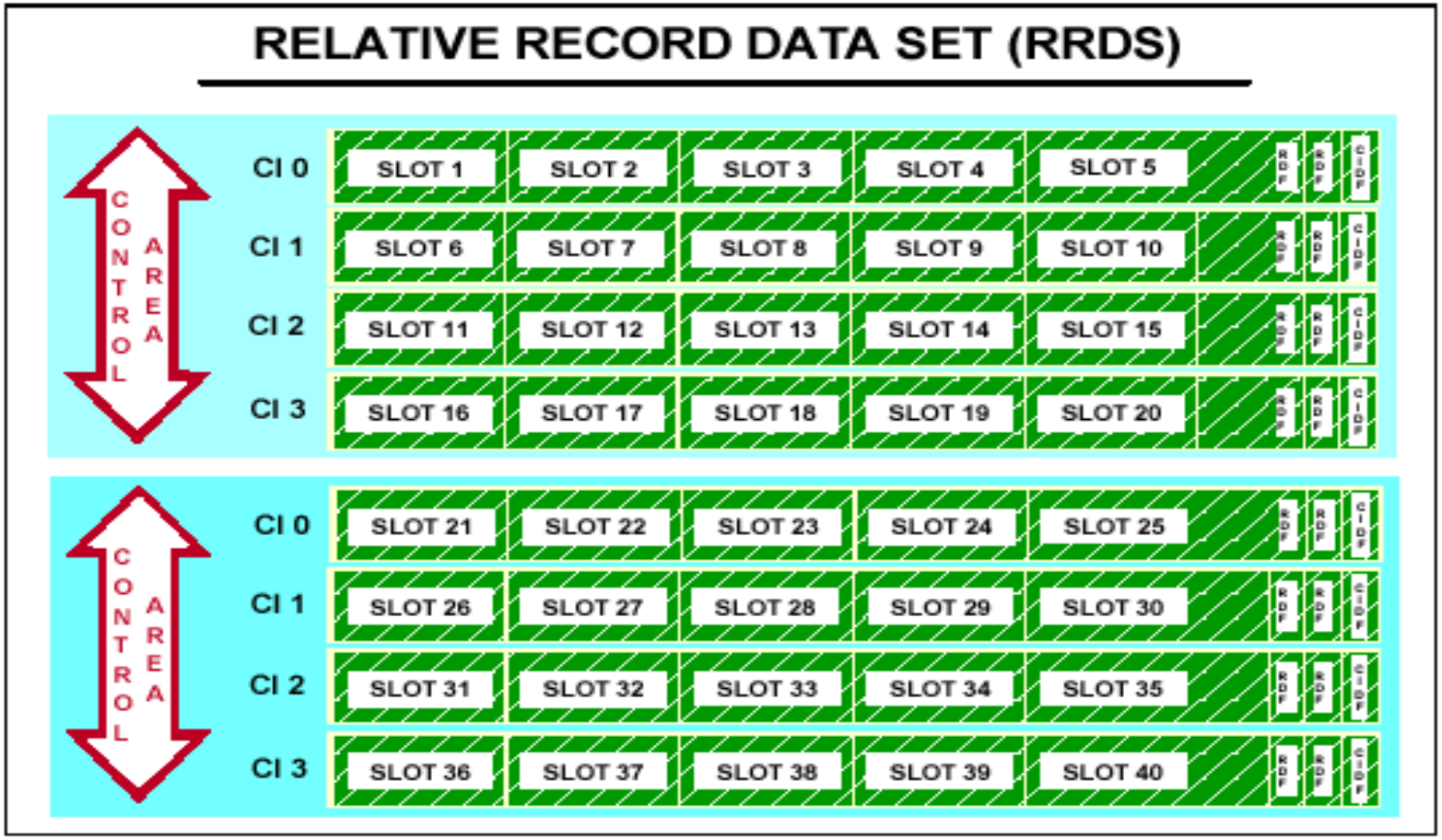
VSAM Datasets Types

- Entry Sequenced Data Set (ESDS)
- Relative Record Data Set (RRDS)
- Key Sequenced Data Set (KSDS)
- Variable Length Relative Record Data Set (VRRDS)
- Linear Data Set (LDS)

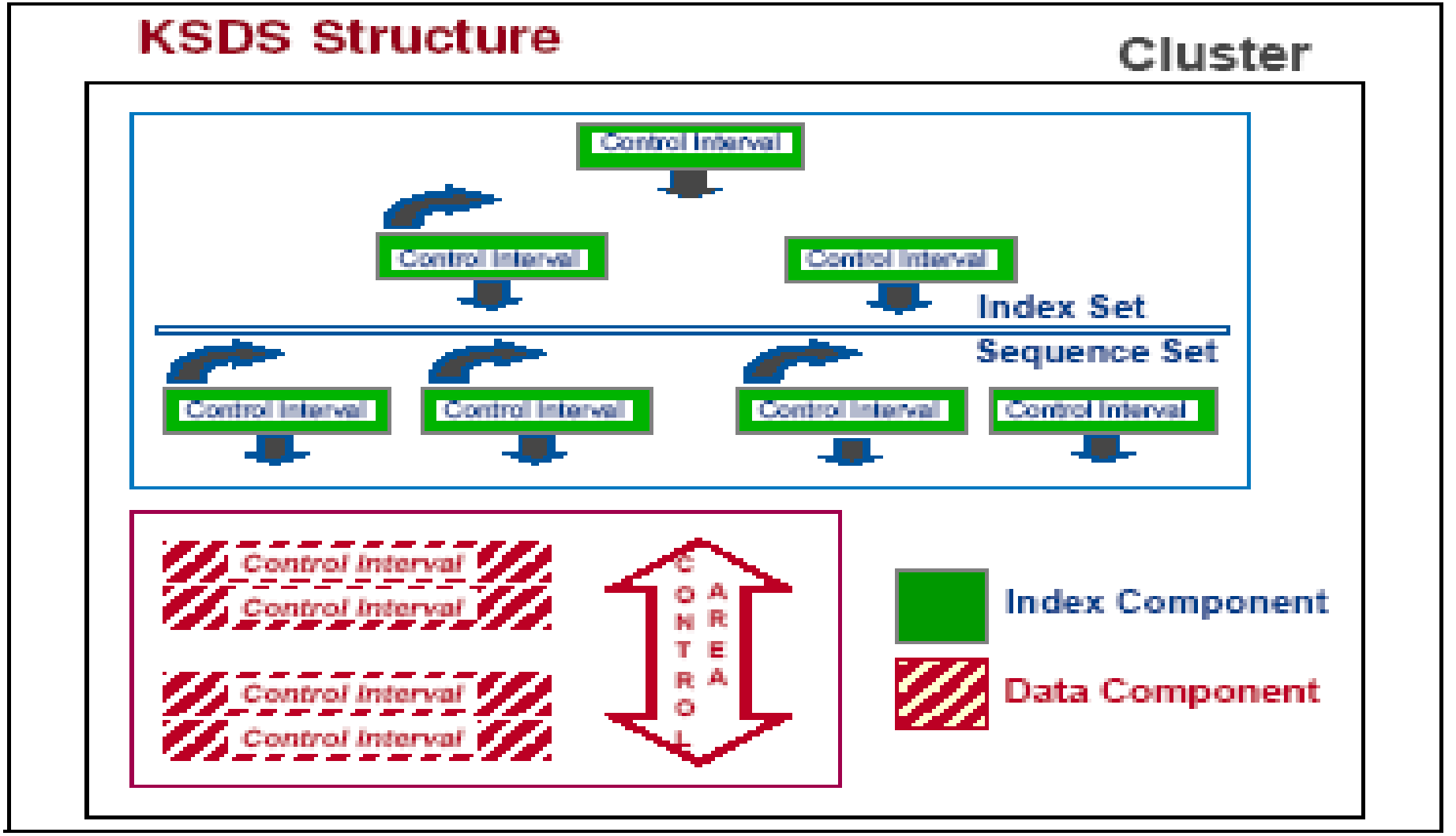
ESDS CA Structure



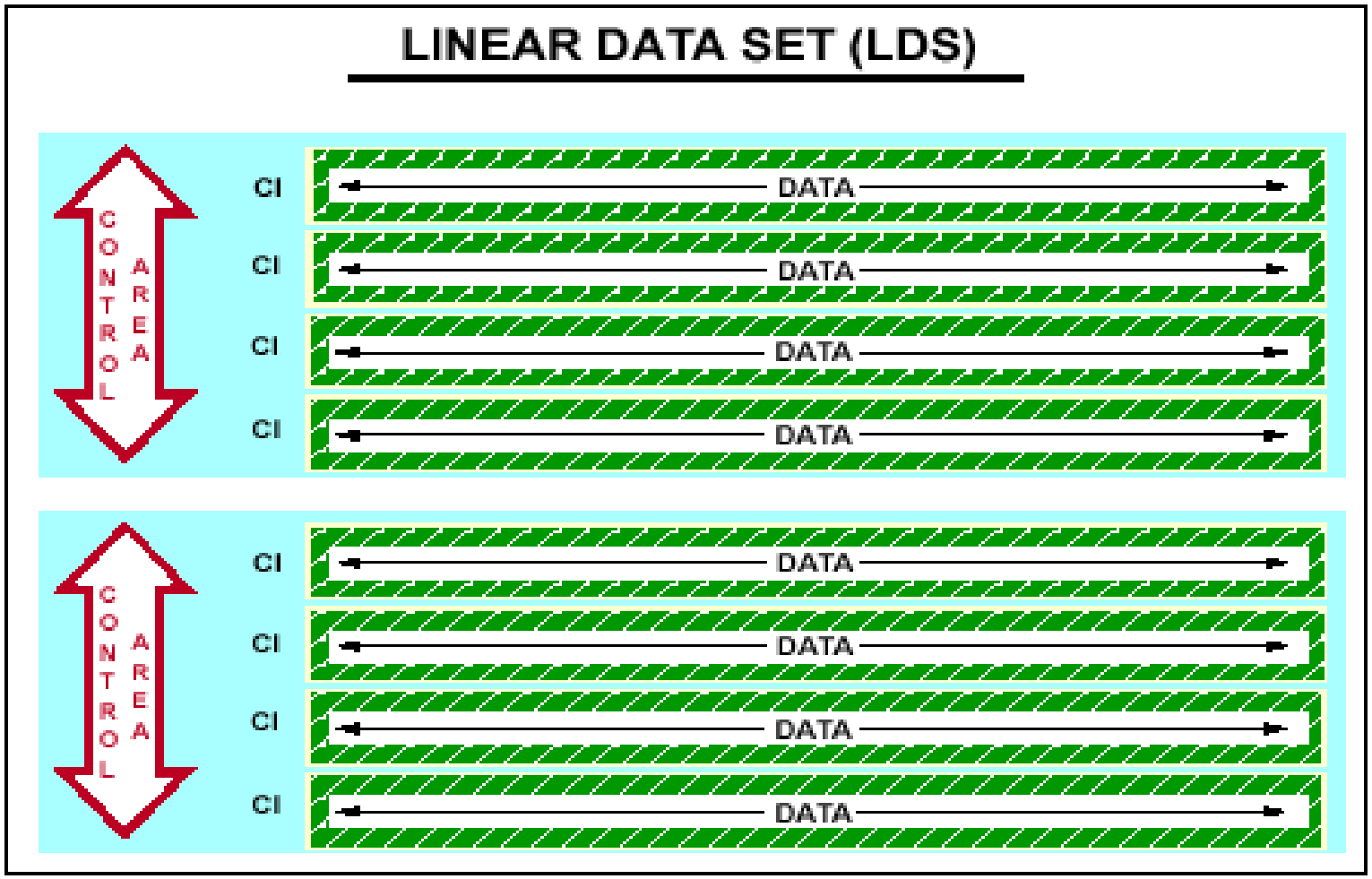
RRDS Structure



KSDS Structure



LDS Structure



Comparison of VSAM DS

ESDS	KSDS	Fixed-length RRDS	Variable-length RRDS	Linear data sets
Records are in the same order as they are entered	Records are in collating sequence by key field after load	Records are in relative record number order	Records are in relative record number order	No processing at record level
Records can be fixed or variable length	Records can be fixed or variable length	Records have fixed length	Records have variable length	No processing at record level
Direct access by RBA	Direct access by key or by RBA	Direct access by relative record number	Direct access by relative record number	Access with Data-In-Virtual (DIV) optionally
Consist of data component only	Consist of data and index components	Consist of data component only	Consist of data and index components	Consist of data component only
Alternate index allowed	Alternate indexes allowed	No alternate index allowed	No alternate index allowed	No alternate index allowed

Comparison of VSAM DS (Contd.)

ESDS	KSDS	Fixed-length RRDS	Variable-length RRDS	Linear data sets
A record's RBA cannot change	A record's RBA can change	A record's relative record number cannot change	A record's relative record number cannot change	No processing at record level
Space at the end of the data set is used for adding records	Free space is used for inserting and lengthening records	Empty slots is in the data set are used for adding records	Free space is used for inserting and lengthening records	No processing at record level
A record cannot be deleted, but you can reuse its space for a record of the same length	Space given up by a deleted or shortened record becomes free space	A slot given up by a deleted record can be reused	Space given up by a deleted or shortened record becomes free space	No processing at record level
Spanned records allowed	Spanned records allowed	No spanned records	No spanned records	No spanned records

Introduction to IDCAMS

IDCAMS Utility

- Multifunction utility program supplied with VSAM (IDCAMS) to manage and maintain datasets

IDCAMS Features

- IDCAMS can be used to
  - Define, Alter and Delete datasets and allocate space for them
  - List and Maintain VSAM Catalogs
  - Load, Print and Copy datasets
  - Reorganize datasets
  - Define and build alternate indexes
  - Facilitate backup and recovery
  - Provide security and integrity functions

Invoking IDCAMS

1. As a Job or Job step

- Example

```
//YOURJOB  JOB (ACCT),'IDCAMS JOB',
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
AMS commands and their parameters
/*
```

Invoking IDCAMS (Contd.)

2. TSO can be used with VSAM and access method services to
   - Execute access method services commands
   - Execute a program to call access method services

3. From an application Program

IDCAMS Commands

COMMANDS

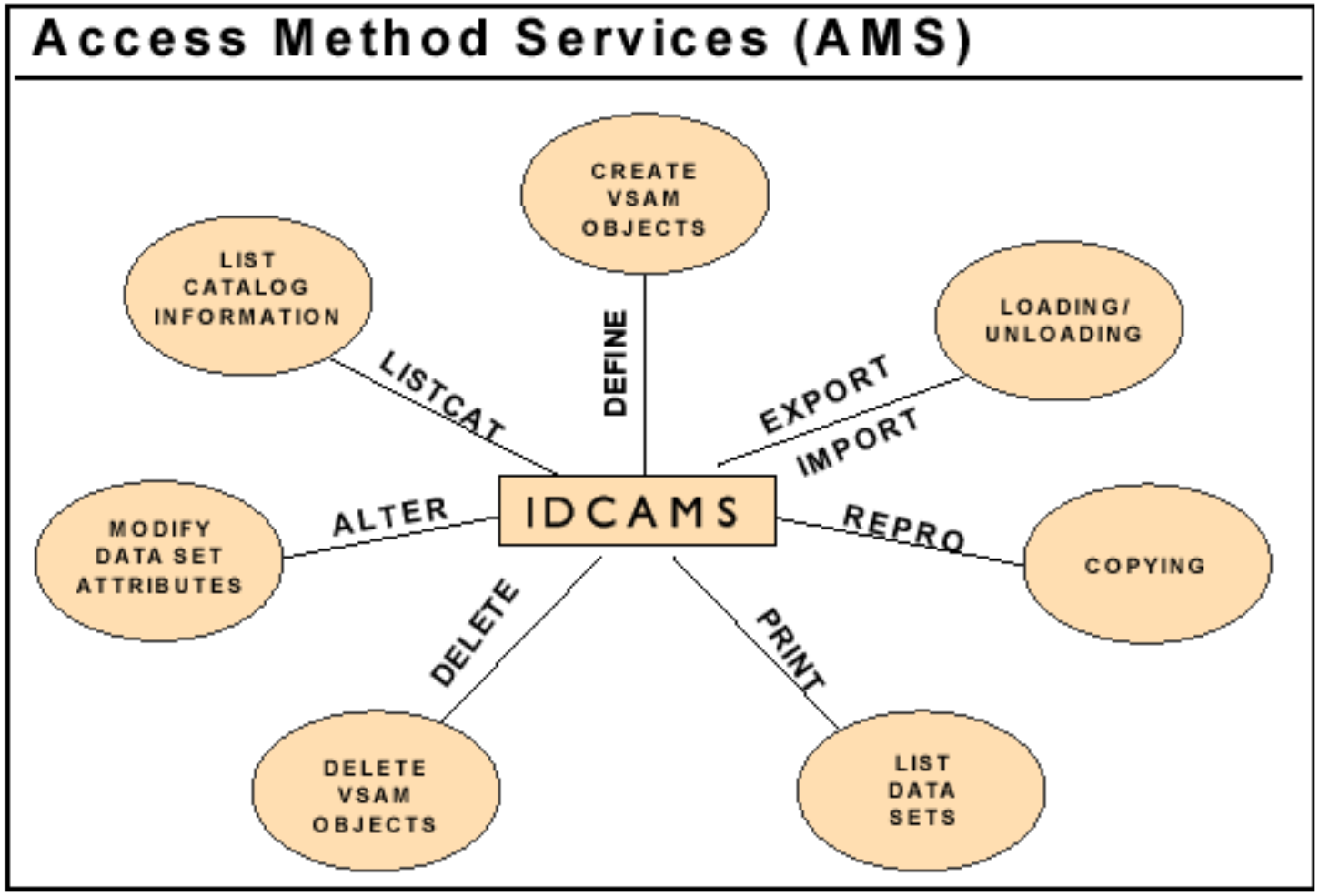
- Modal Commands
  - Used to control execution and establish options
  - Modal commands allow the conditional execution of functional commands.

- Functional Commands
  - Used to invoke access method services

COMMANDS (Contd..1)

- COMMAND specifies the type of service requested
- Default statement margins are positions 2 through 72.
- Command stmt can be continued to the next line with '-' as the last parameter
- Comment may be embedded in Command statements between /* and */

COMMANDS (Contd..2)



Condition Codes

- IDCAMS returns a condition code following the execution of each Command

- Can be used to selectively execute or skip subsequent commands

- Initialized to zero at the start of the execution

Condition Codes (Contd..1)

LASTCC - specifies the condition code value that resulted from the immediately preceding function command

MAXCC - specifies the maximum condition code value that has been established by any previous function command

Condition Codes (Contd..2)

Code	Explanation
0	Function executed successfully
4	Some problem was met in executing the function but it was possible to continue. The continuation might not provide exact results, but no permanent harm will have been done

Condition Codes (Contd..3)

Code	Explanation
8	A requested function was completed but major specifications were unavoidably bypassed
12	The requested function could not be performed, as a result of a logical error
16	A severe error occurred that caused the Job Step to terminate

Functional Commands (Contd..2)

- IMPORT Connects user catalogs and imports VSAM datasets
- EXPORT Disconnects user catalogs and exports VSAM datasets.
- PRINT Used to print VSAM and non-VSAM datasets and catalogs

Functional Commands (Contd..3)

- ALTER - Used to alter attributes of datasets and catalog that have already been defined

- Alter
  - Freespace
  - Volume
  - Prevent temporary updates to a VSAM dataset

Functional Commands (Contd..4)

- BLDINDEX - Builds alternate indexes for existing datasets

- Keys will be actually built from the base cluster with the BLDINDEX command

- VERIFY : Used to cause a catalog to correctly reflect the end of a dataset after an error occurred in closing a VSAM dataset. The error may have caused the catalog to be incorrect.

Example: Modal & Functional Commands

```
//JOBNAME JOB ...
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DEFINE CLUSTER -
IF LASTCC = 0 THEN -
REPRO -
ELSE -
DO
SET LASTCC = 0
DELETE CLUSTER ...
DEFINE CLUSTER ...
END
/*
```

Define and Load a VSAM Dataset

DEFINE CLUSTER

- The DEFINE CLUSTER command can be used to define a cluster and specify attributes for the cluster as a whole and for the components of the cluster. The general format is :

DEFINE CLUSTER example

```
//JOB1  JOB   CLASS=A
//DEFINE   EXEC  PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DEFINE CLUSTER -
(NAME(TRGXXX.MAST.CLUSTER) -
TRACKS(51) -
CONTROLINTERVALSIZE(4096) -
INDEXED -
VOLUMES(TRG001) -
KEYS(6,0) -
FREESPACE(10,10) -
DATA -
(NAME(TRGXXX.MASTER.DATA) -
TRACKS(21) -
RECORDSIZE(200 200)) -
INDEX -
(NAME(TRGXXX.MASTER.INDEX) -
IMBED).
/*
```

Functional Commands (Contd..5)

REPRO

- Loads an empty VSAM cluster with records
- Creates backup of a VSAM dataset on a sequential dataset and restores the dataset
- Merges data from two VSAM datasets

LISTCAT

- The LISTCAT command lists catalog entries.
  - Freespace, CI/CA split, Data Attributes, Statistics. Allocation information

- All parameters of the LISTCAT command are optional

- LISTCAT can be used under TSO

LISTCAT Syntax

LISTCAT ALIAS | ALTERNATEINDEX |

CLUSTER | NONVSAM | PATH | ENTRIES(entryname/password entryname/password..) |

LEVEL(level)

EXPIRATION(days)

NAME | HISTORY | VOLUME|

ALLOCATION | ALL

OUTFILE(ddname)

LISTCAT example

```
//JOBTRGXX   ...
//LISTCAT1   EXEC PGM=IDCAMS
//SYSPRINT   DD  SYSOUT=A
//SYSIN      DD *
LISTCAT -
ENTRIES(TRGXXX.MAST.CLUSTER) -
CLUSTER -
ALL
/*
```

REPRO

- The REPRO command copies VSAM and non-VSAM data sets, copies catalogs.

REPRO Syntax

REPRO {INFILE(ddname)/password
INDATASET(entryname)/password
{OUTFILLE(ddname/password
OUTDATASET(entryname/password}
(CHARACTER | DUMP | HEX)
(FROMKEY(key) |
(FROMADDRESS(address) |
(FROMNUMBER(number))
SKIP(number)
(TOKEY(key) |
TOADDRESS(address) |
TONUMBER(number))
COUNT(Number)

REPRO example

Example: Load VSAM KSDS from Non-VSAM dataset and copy selected records to an ESDS

```
//JOBTRGXX   ...
//STEP1     EXEC PGM=IDCAMS
//INPUTDD    DD  DSN=TRGXXX.DATA.RECORDS,DISP=OLD
//SYSPRINT   DD  SYSOUT=A
//SYSIN      DD *
REPRO -
INFILE(INPUTDD) -
OUTDATASET(TRGXXX.KSDS.CLUSTER)
/*
```

DELETE

- The DELETE command deletes catalogs, VSAM datasets, non-VSAM datasets and objects.

DELETE Command Syntax

DELETE (entryname/password entryname/password..)

ALIAS | ALTERNATEINDEX | CLUSTER | NONVSAM |

PATH | USERCATALOG

ERASE | NOERASE

PURGE| NOPURGE

FORCE | NOFORCE

DELETE example

```
//DELMAST EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD *
DELETE -
(TRGXXX.MAST.CLUSTER  -
TRGXXX.TEST.DATA) -
FILE(DD1) -
PURGE -
ERASE
/*
```

PRINT

- PRINT: The PRINT command prints VSAM datasets, non-VSAM datasets, and catalogs. The syntax is:

PRINT Syntax

PRINT {INFILE(ddname /password)
INDATASET(entryname /password) }
CHARACTER| DUMP | HEX
FROMKEY(key) |
FROMADDRESS(address) |
FROMNUMBER(number) }
SKIP(number)
OUTFILE(ddname)
TOKEY(key) |
TOADDRESS(address) |
TONUMBER(number)|
COUNT(Number)

PRINT example

```
//JOBTRGXX   ...
//DEFAIX    EXEC PGM=IDCAMS
//SYSPRINT   DD   SYSOUT=A
//SYSIN      DD *
PRINT -
INDATASET(TRGXXX.MAST.CLUSTER) -
FROMKEY(E00001) -
COUNT(100)
/*
```

VERIFY

- The VERIFY command causes a catalog to correctly reflect the end of a VSAM data set after an error occurs while closing a VSAM data set

- The syntax is:

  VERIFY {FILE(ddname/password) |

  DATASET(entryname/password)}

VERIFY Example

Example of a typical VERIFY usage

- Job TRGJJJJ has terminated abnormally and VSAM dataset TRGXXX.MAST.CLUSTER is not closed correctly. This job is followed by another job that uses this file for an IDCAMS copy. Following are the set of return codes that will appear in the Job Spool

- In JESMSGLG the error will be

  IEC161I 056-084,TRGGXXXX,JS010,IFILE01,,,
- In SYSPRINT of the step failed the error will be

  IDC3300I ERROR OPENING TRGXX.MAST.CLUSTER
  IDC3351I ** VSAM OPEN RETURN CODE IS 118

- Solution: A VERIFY issued on VSAM file TRGXXX.MAST.CLUSTER will correct this error

VERIFY Example (Contd..)

```
//JOBTRGXX   ...
//STEP1     EXEC PGM=IDCAMS
//SYSPRINT   DD   SYSOUT=A
//SYSIN      DD *
VERIFY -
DATASET(TRGXXX.MAST.CLUSTER)
/*
```

References

- MVS/VSAM for the Application Programmer – Gary D.Brown & S.A.M. Smith

- VSAM Demystified – Mary Lovelace et al. IBM Redbooks 2003 (Document Number SG24-6105-01) in IBM's Website

- DFSMS/MVS V1R2 Access Method Services for VSAM (Document Number SC26-4905-01) in IBM's Website

THANK YOU