VSAM for PL/I Programmers

VSAM For PL/I Programmers - Course Objectives

On successful completion of this course, the student, with the aid of the appropriate reference materials, should be able to:

1. Create and delete VSAM data sets, using Access Method Services (AMS)

2. Load VSAM data sets, using AMS REPRO or PL/I programs

3. Use the AMS PRINT command to list all or parts of a VSAM data set, and the AMS LISTCAT command to list all or part of a VSAM catalog

4. Use PL/I programs to process VSAM data sets

5. Create and use VSAM alternate indexes and paths.

VSAM For PL/I Programmers - Topical Outline

VSAM For PL/I Programmers - Topical Outline, p.2.

Day Two

This page intentionally left almost blank.

Section Preview

❒ Introduction to VSAM

CI's and CA's

VSAM Data Set Organizations

Clusters, Components, and Catalogs

Access Method Services (AMS)

ESDS's

DEFINE CLUSTER, REPRO, PRINT, DELETE

Working with ESDS's using AMS (Machine Exercise)

Virtual Storage Access Method

❒ Design Philosophy

Easy To Use

Take decisions away from the programmer

The system should manage data

Let the programmer concentrate on the application

High Performance

Sequential retrieval / update

Random retrieval / update

Insertions / deletions

Alternate indexes or keys

Data Security / Integrity

Prevent unauthorized access

Internal consistency checks

Simple backup / recovery tools

VSAM - Actual Implementation

- ❒ Four data set organizations

  ESDS - Entry Sequenced Data Set

  KSDS - Key Sequenced Data Set

  RRDS - Relative Record Data Set

  LSDS - Linear Space Data Set (also called simply "LDS" for "Linear Data Set")

- ❒ Alternate Indexes

- ❒ Two-level Catalog structure

  Master Catalog

  User Catalogs

- ❒ VSAM utility program: IDCAMS

- ❒ Programming language support

- ❒ Database product support

- ❒ Data security

  Data encryption

  Passwords

  RACF interface

VSAM - Units Of Space

❐ Control Interval (CI)

Unit of transmission between DASD and storage

User selects, or may allow VSAM to select

If you select, VSAM may override you (and not tell you about it!)

❐ CYLINDERS / TRACKS / RECORDS / MEGABYTES / KILOBYTES

Unit of space allocation request

You specify

❐ Control Area (CA)

Unit of space allocation fulfillment

1 CA = 1 track, or 2 tracks, or ... 1 cylinder

VSAM selects

Control Interval Layout

LR_1	LR_2	LR_3	...FS...	RDF_3	RDF_2	RDF_1	CIDF

LR_n — Logical Record *n*

RDF_n — Record Descriptor Field n
- **3 bytes (1 for flag bits; 2 for length of corresponding LR)**

FS — Free Space; neither data nor control information

CIDF — Control Interval Descriptor Field
- **4 bytes; only 1 in each CI**

+ two bytes: space in use
+ two bytes: length of free space

- **When two or more consecutive records in a CI have the same length, need only 2 RDFs for that string of records:**

RDF-count / RDF-length

So:

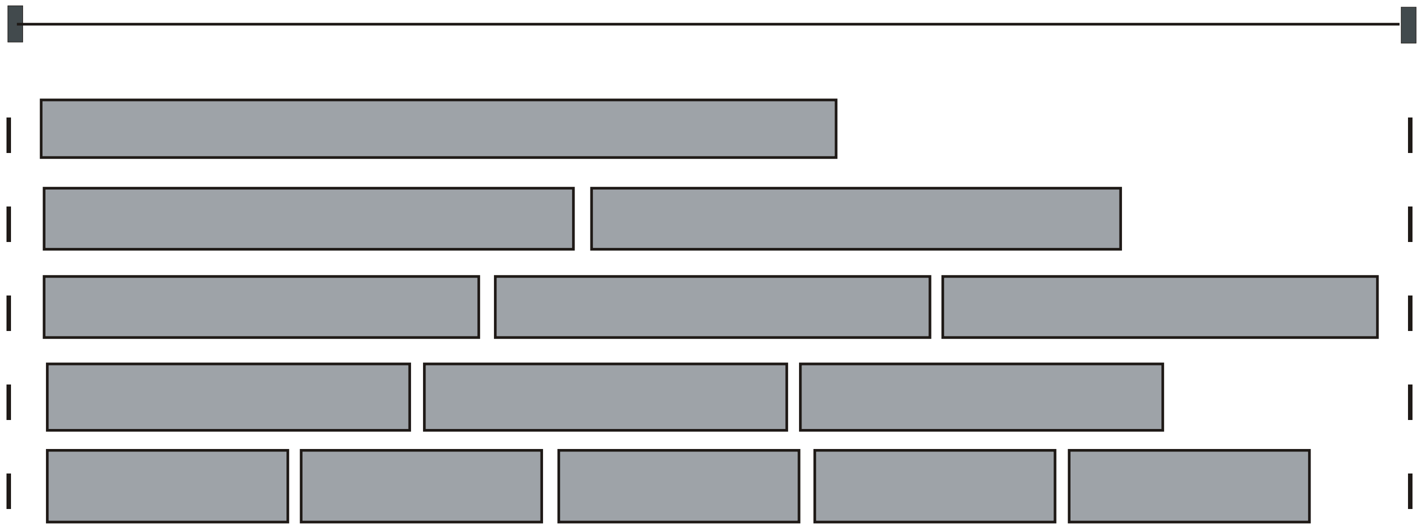
(80)	(80)	(80)	(80)	(100)	(120)	...FS...	L120	L100	C4	L80	CIDF

CI's and Common DASD Devices

CI SIZE	CI's / trk - 3380	CI's / trk - 3390
512	46	49
1024	31	33
1536	23	26
2048	18	21
2560	15	17
3072	13	15
3584	11	13
4096	10	12
4608	9	10
5120	8	9
5632	7	9
6144	7	8
6656	6	7
7168	6	7
7680	5	6
8192	5	6
10240	4	5
12288	3	4
14336	3	3
16384	2	3
18432	2	3
20480	2	2
22528	2	2
24576	1	2
26624	1	2
28672	1	1
30720	1	1
32768	1	1

3380:	15 Tracks/Cylinder	885 | 1770 | 2655 Cylinders/Volume
3390:	15 Tracks/Cylinder	1113 | 2226 | 3339 Cylinders/Volume

Some CI Sizes Work Better Than Others



VSAM CISZ Worksheet

Device Type: 3380 TRKS/CYL: 15 CYLS/VOL:_____

For record size (RECSZ): 200

---------- from table --------		A	B
CISZ	**CIs/Trk**	**Records/CI**	**Records/Trk**
512	46	2	92
1024	31	5	155
1536	23	7	161
2048	18	10	180
2560	15	12	180
3072	13	15	195
3584	11	17	187
4096	10	20	200
4608	9	22	198
5120	8	25	200
5632	7	28	196
6144	7	30	210
6656	6	33	198
7168	6	35	210
7680	5	38	190
8192	5	40	200
. . .	. . .	. . .	. . .

A = INT((CISZ - 10) / RECSZ)
for RRDS, A = INT((CISZ - 4) / (RECSZ + 3))

B = A * (CIs/Trk)

Rules of thumb:
- **For disk utilization, use CISZ w/ most records/trk**
- **For primarily sequential processing, use large CIs**
- **For primarily random processing, use small CIs**
- **Don't exceed any installation standard maximum**

Relative Byte Address - RBA

❒ **Fullword binary integer**

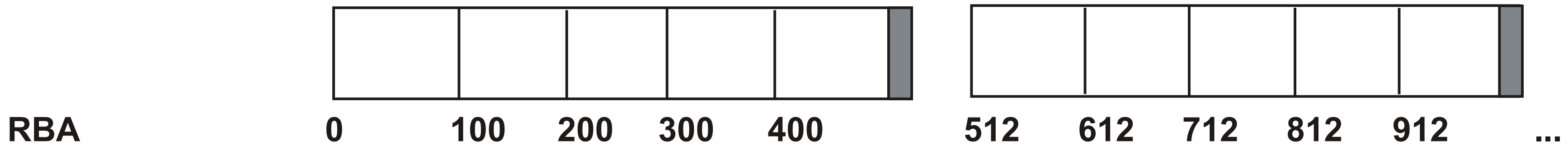
32 bits unsigned

0 to 4,294,967,296

❒ **Displacement into file**

Including free space, RDF's, and CIDF's

❒ **For example, assume 100 byte records and 512 byte CI size:**



❒ **RBA may be used to address / locate records in a VSAM data set (Assembler and PL/I)**

VSAM Data Set Organizations

ESDS

- Physical sequential
- Fixed length, variable length, or spanned records
- Add only at end (no inserts)
- Sequential retrieval: all languages
- Random retrieval possible in Assembler and PL/I
- Update in place allowed
- Alternate index support only in Assembler and PL/I
- No deleting records
- No changing record length

KSDS

- Sequenced by unique key field (ascending sequence only)
- Fixed length, variable length, or spanned records
- Add records at end
- Insert records in middle
- Sequential retrieval: all languages
- Random retrieval: all languages
- Dynamic retrieval: all languages
- Free space may be used for inserts
- May delete records (space freed up added to free space)
- Alternate index support in all languages

VSAM Data Set Organizations, 2

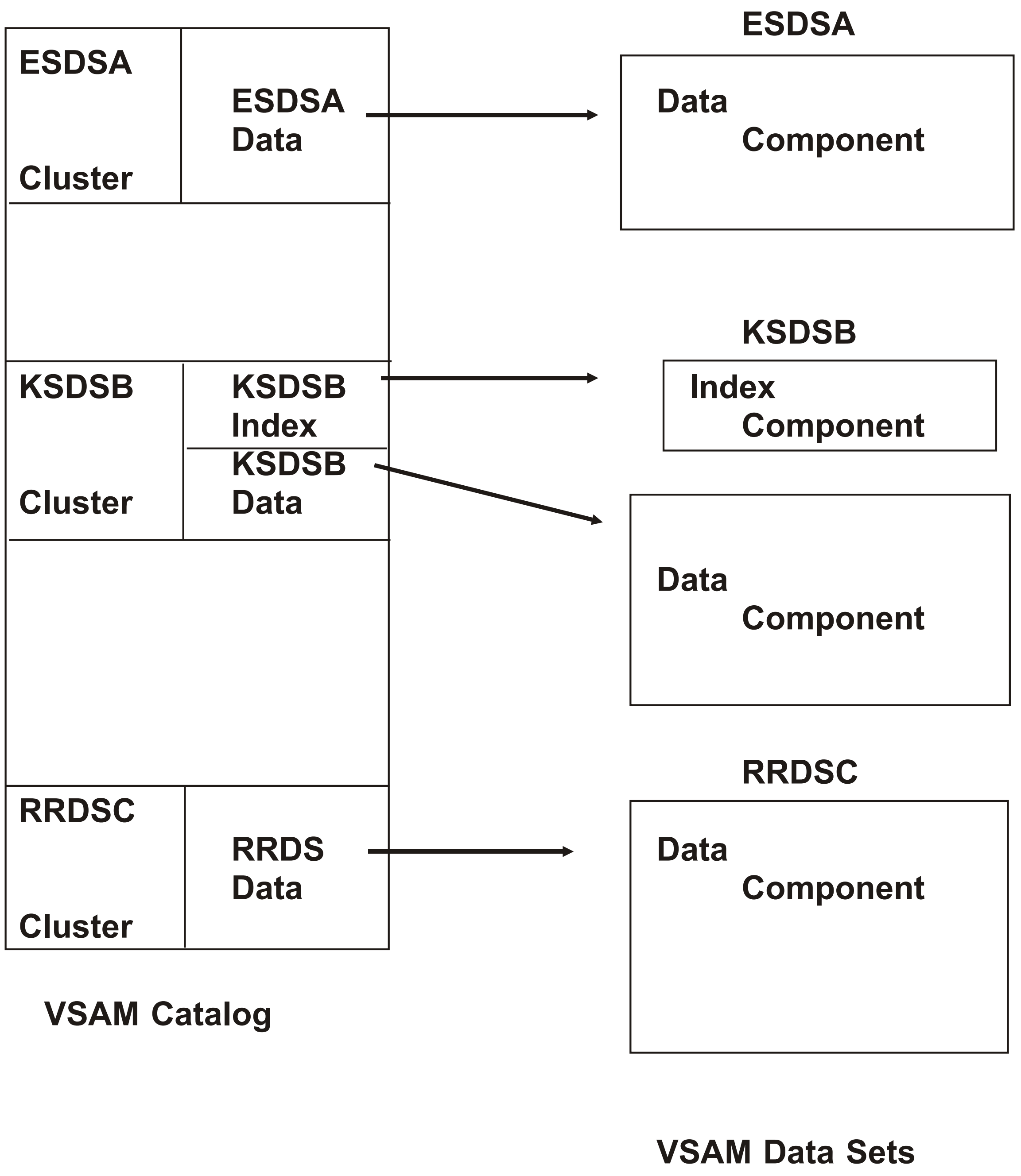
<u>RRDS</u>

- Fixed length or variable length records
- File preformatted into <u>slots</u>: areas large enough to hold 1 record
- Slots are numbered from 1 - n
- Add records by slot number
- Sequential retrieval: all languages
- Random retrieval (by slot number): all languages
- Dynamic retrieval: all languages
- May delete records (slot available for future insert)
- Update in place allowed
- No alternate indexes

<u>LSDS (or LDS)</u>

- Control intervals with no RDF's, CIDF's, or embedded free space
- CI size always 4096 bytes
- Appears as large string of memory rather than discrete records
- Access through calls to 'Window services' routines
  (CSRxxxx routines provided by IBM)
- May be permanent or temporary
- Permanent LSDS's may have temporary changes made, or
  permanent updates
- Used extensively behind the scenes by DB2 and other products

Clusters and Components



VSAM Data Sets - JCL

```
//ddname      DD        DSN=clustername,DISP=SHR
```

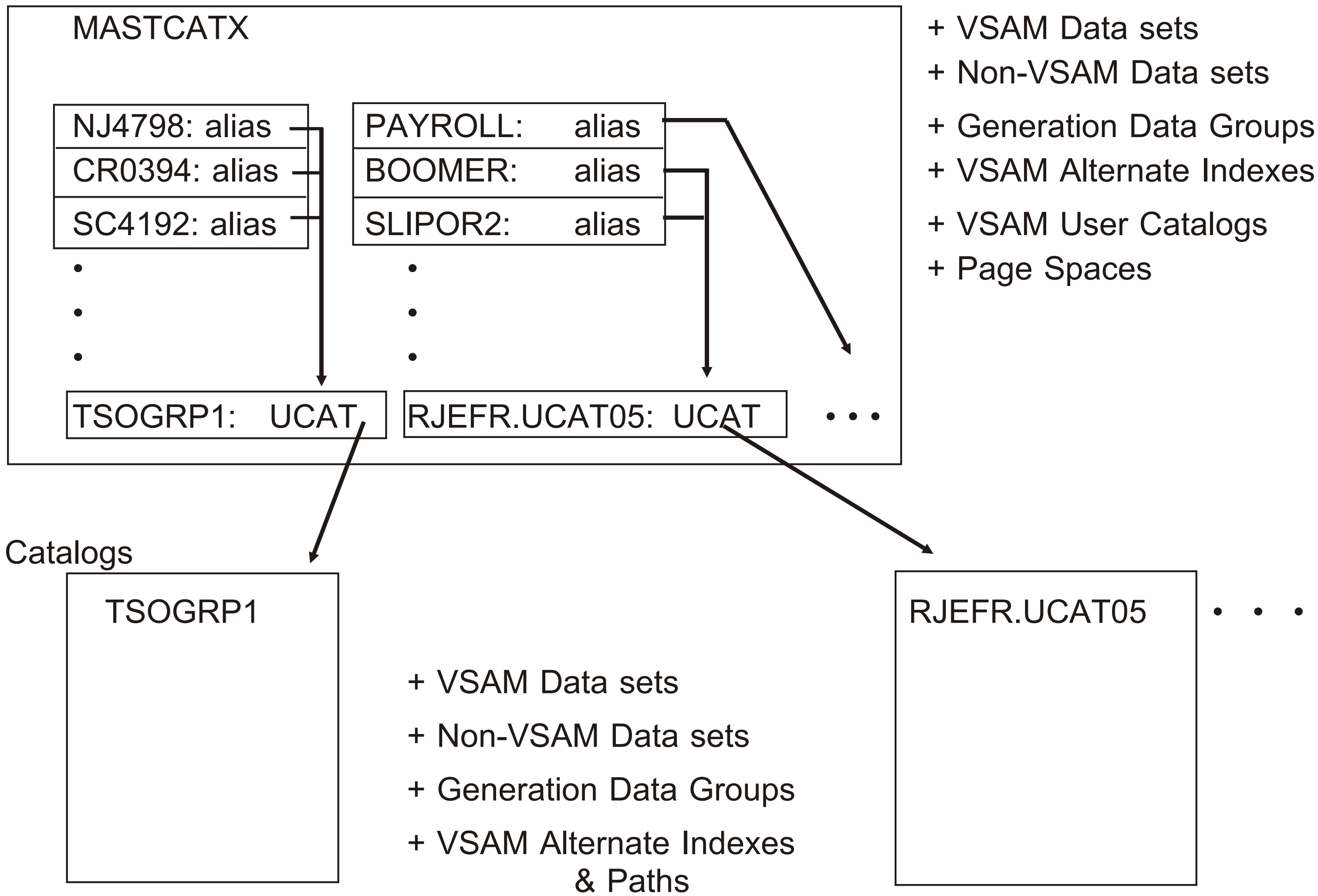
or

```
//ddname      DD        DSN=clustername,DISP=OLD
```

❒ **Remember, all VSAM data sets are cataloged**

VSAM Catalog Hierarchy

VSAM Master Catalog = System Master Catalog



+ VSAM Data sets
+ Non-VSAM Data sets
+ Generation Data Groups
+ VSAM Alternate Indexes & Paths
+ VSAM User Catalogs
+ Page Spaces

+ VSAM Data sets
+ Non-VSAM Data sets
+ Generation Data Groups
+ VSAM Alternate Indexes & Paths

<u>A</u>ccess <u>M</u>ethod <u>S</u>ervices

❒ **The VSAM utility program**

❒ **Program name: IDCAMS**

Batch program

TSO command processor

❒ **Primary way to create / delete VSAM data sets**

Cannot use JCL alone

✗ Except when using SMS (Storage Management Subsystem)

❒ **Provides many additional useful support functions**

❒ **Does work based on user <u>commands</u>**

Typical Access Method Services JCL

```
//--------        JOB       -----

//STEPX           EXEC      PGM=IDCAMS

//SYSPRINT        DD        SYSOUT=*

//SYSIN           DD        *
      .
      .     AMS command statements, such as
      .
      PRINT   INDATASET(VSM2.MIL.HRS.H2HI)   HEX
      DELETE   (VSM3.TABLES.LV009,   VSM2.TABLES.LV00A)
      DELETE   VSM.MM.LOMFRT.BANDIT
```

AMS Command Syntax

Syntax

COMMAND   parameters

- ❐ **Columns 2 - 72**
- ❐ **Parameters**
  - — **Positional**
  - — **Keyword**
    - **+ Reserved word**
    - **+ Reserved word(value(s))**

Lists of values must be enclosed in parentheses, separated by commas and / or blanks

If there's only one element in the list, may omit the parentheses

Examples

```
PRINT   INDATASET(PORBLE.MARBLE.MUDDLE)   HEX   COUNT(50)

DELETE   (EKR59.TABLES.LV009,   EKR59.TABLES.LV00A)

DELETE   LOMFRT.BANDIT.FILE
```

Abbreviations / Comments / Continuation

- ❒ **Most commands and parameters have abbreviations you may code instead of the entire reserved word:**

```
PRINT   INDATASET(...

PRINT  IDS(...
```

- ❒ **Comments:** **/* ..... */**

- ❒ **Continuation**

  Use '—' or '+'

  Just use '—' here; differences not worth discussing

```
PRINT   IDS (VSM2.MIDORI.KEYS) —
        HEX —
        SKIP(31) —
        COUNT(19)
```

  Lack of a continuation character means end of a command

- ❒ **Spaces may be freely inserted except in the middle of a value**

Types Of Commands

Functional

Accomplish work

Return a condition code value

Modal

Conditional

Test / set condition codes

IF-THEN-ELSE, DO-END, NULL, SET, PARM, CANCEL

- ❐ **All functional AMS commands are also native TSO commands**
- ❐ **No modal commands are TSO commands**

Check out the AMS Reference manual

ESDS Characteristics

- ❒ **Records in physical sequence**
- ❒ **Fixed, variable, spanned records supported**
- ❒ **Physical organization:**

LR	**LR**	**LR**	**LR**	**LR**	**LR**	**FS**	**RDF**	**RDF**	**CIDF**

- ❒ **Free space is unused and unusable**
- ❒ **Sequential retrieval supported in all languages**
- ❒ **Random retrieval possible in Assembler, PL/I, and C but not COBOL**
- ❒ **Update in place (but no change in record length) supported**
- ❒ **No record deletion or insertion**
- ❒ **Extend by adding records to the end**
- ❒ **Alternate index support only in Assembler, PL/I, or through CICS support in any CICS-supported language**

DEFINE CLUSTER

❐ **Creates catalog records for VSAM data sets**

Cluster catalog record

Data component catalog record

Index component catalog record (KSDS and variable-length record RRDS only)

❐ **Reserves space on volume(s)**

❐ **Does <u>not</u> put any records into data set**

DEFINE CLUSTER - ESDS Example

```
DEF   CL  (NAME(ADJ578.PEPLHIST.W001LNA)          –
               REC(5000 400)                      –
               VOL(MICKEY)                        –
               CISZ(4096)                         –
               NIXD                               –
               RECSZ (420 420)                    –
               SPEED   )                          –
      DATA (NAME(ADJ578.PEPLHIST.W001LNA.DATA) )
```

❒ **Here's a possible DEFINE CLUSTER command for an ESDS**

We'll discuss the parameters in just a minute, but first we examine some alternative ways of coding this same DEFINE CLUSTER command (in terms of continuation and punctuation) ...

DEFINE CLUSTER - ESDS Example 2

```
DEF   CL   (NAME(ADJ578.PEPLHIST.W001LNA)                –
             REC(5000 400)   VOL(MICKEY)  CISZ(4096)  NIXD     –
             RECSZ (420 420)    SPEED)                         –
      DATA  (NAME(ADJ578.PEPLHIST.W001LNA.DATA))
```

DEFINE CLUSTER - ESDS Example 3

```
DEF   CL(  –
           NAME(ADJ578.PEPLHIST.W001LNA)     –
           REC(5000 400)                     –
           VOL(MICKEY)                       –
           CISZ(4096)                        –
           NIXD                              –
           RECSZ(420 420)                    –
           SPEED                )            –
      DATA(  –
              NAME(ADJ578.PEPLHIST.W001LNA.DATA)   )
```

DEFINE CLUSTER - ESDS Example 4

```
DEF   CL( –
          NAME(ADJ578.PEPLHIST.W001LNA)              –
          REC(5000 400)                              –
          VOL(MICKEY)                                –
          CISZ(4096)                                 –
          NIXD                                       –
          RECSZ (420 420)                            –
          SPEED                                      –
                                 )                   –
      DATA( –
          NAME(ADJ578.PEPLHIST.W001LNA.DATA)         –
                                 )
```

❒ **You get the idea; now, let's examine the parameters ...**

VSAM Data Set Names

❒ **VSAM cluster and component names are made up of <u>qualifiers</u>**

A qualifier is 1 - 8 alphanumeric and national (@, #, $) characters, the first of which is not numeric

❒ **A cluster or component name consists of one or more qualifiers separated by periods, up to a maximum of 44 characters total (including the periods)**

VSAM Data Sets — Cluster and Component Names

- ❒ **Typically, most installations have further restrictions for data set names based on naming conventions or standards**

- ❒ **A common convention is for the cluster level catalog record name to have a low level qualifier of CLUSTER**

- ❒ **Another common convention is for the name for the data component of a VSAM cluster to consist of the cluster name followed by a low level qualifier of DATA**

  For example, a VSAM data set named HLQ.FRWE.CLUSTER would have a data component name of HLQ.FRWE.DATA

- ❒ **Similarly, the index component of a KSDS VSAM data set would have a name consisting of the cluster name followed by a low level qualifier of INDEX**

  Using our example, if HLQ.FRWE.CLUSTER is a VSAM KSDS, the index component would be named HLQ.FRWE.INDEX

- ❒ **These are not requirements, just conventions**

Cluster and Component Names in DEFINE

❒ **In an AMS DEFINE command, when you are defining a cluster, you code:**

```
DEFINE CLUSTER (attributes) ...
```

Although you may code the attributes in any order, the first attribute, logically, is the cluster name, so you typically code something like:

```
DEFINE CLUSTER (NAME(clustername) ... )
```

As part of the same command, you code information to describe the data component:

```
DEFINE CLUSTER (NAME(clustername) ... )        -
          DATA (attributes)
```

Since attributes defined at the cluster level generally are inherited to the data component level, you typically only need to define the name attribute for the data component:

```
DEFINE CLUSTER (NAME(clustername) ... )        -
          DATA (NAME(componentname))
```

Recall the data component's name is likely to be the clustername followed by the low level qualifier DATA

If you do not code a name for the data component, AMS will assign a name for you, but it's a system-defined name, so it does not follow our convention

Specifying Volumes in DEFINE

- ❐ **All VSAM files are disk-resident**

- ❐ **A disk volume is uniquely identified by a six character volume serial**

- ❐ **To indicate which volume a new VSAM data set is to reside on, you must specify this volume serial using the VOLUME parameter:**

```
VOLUME(volser)
```

  If your data set is likely to be larger than one volume, you must specify a list of candidate volume serials, for example:

```
VOLUME(MICKEY, DOPEY, DAFFY, SNEEZY)
```

- ❐ **The parameter named VOLUME may be abbreviated VOL:**

```
VOL(VS0001, WHOOPY)
```

- ❐ **Typically, volumes are assigned by a data administrator**

- ❐ **In installations that use SMS (the Storage Management Subsystem), you can let SMS choose the volume by coding**

```
VOL(*)
```

  SMS is not discussed in this course

Specifying Data Set Organization in DEFINE

❒ **To indicate what VSAM organization a new file is to have, include one of these keywords:**

NONINDEXED	- ESDS
INDEXED	- KSDS
NUMBERED	- RRDS
LINEAR	- LSDS (LDS)

❒ **The default, if you omit all of these words, is INDEXED**

❒ **Each of these words has an abbreviation:**

Term	**Abbreviation**
NONINDEXED	NIXD
INDEXED	IXD
NUMBERED	NUMD
LINEAR	LIN

Specifying Record Size in DEFINE

❒ **The RECORDSIZE parameter takes two values**

  The average record size

  The maximum record size

❒ **For fixed length records, you must specify the same value twice:**

```
RECORDSIZE(240   240)
```

❒ **The abbreviation for RECORDSIZE is RECSZ:**

```
RECSZ(240   240)
```

Specifying Control Interval Size in DEFINE

❒ **Use the CONTROLINTERVALSIZE parameter:**

```
CONTROLINTERVALSIZE(8192)
```

❒ **This parameter may be abbreviated CISZ:**

```
CISZ(8192)
```

❒ **If you specify a value not in the list, VSAM rounds up to the next value in the list**

If you specify a value greater than 32768, the DEFINE request fails

Specifying Data Set Size in DEFINE

- ❐ **You must tell AMS how large a new VSAM file is expected to be by specifying how many records the data set has, or how many tracks or cylinders or megabytes or kilobytes of space the data set will require**

  Specify a primary (initial) amount and, optionally, a secondary (subsequent) amount

```
RECORDS(30000   400)

TRACKS(300  15)

CYLINDERS(20   1)

MEGABYTES(200   10)

KILOBYTES(170   64)
```

- ❐ **These parameters may be abbreviated REC, TRK, CYL, MB, and KB, respectively:**

```
REC(30000   400)

TRK(300   15)

CYL(20   1)

MB(200   10)

KB(170   64)
```

- ❐ **Choose and specify only one of these parameters for a single DEFINE**

The SPEED Parameter of DEFINE CLUSTER

❒ **This parameter consists of one of two alternatives:**

SPEED | RECOVERY

RECOVERY, the default, says when you load your VSAM data set to preformat all data CA's with an end-of-file marker

✗ Thus, if you have a problem during the load, you may pick up where you left off

SPEED says only write the embedded end-of-file marker when the load is completed

SHAREOPTIONS (n m)

❐ **The "SHAREOPTIONS" setting only applies when multiple users are accessing the same VSAM cluster simultaneously, all with "DISP=SHR" in their JCL**

"DISP=OLD" forces exclusive use, as always

n – Cross Address Space Sharing (within a single system)	m – Cross Systems Sharing (shared DASD)
1 – One update <u>or</u> any number of reads	1,2 – Reserved
2 – One update <u>and</u> any number of reads	
3 – Any number of updates and any number of reads	3 – Any number of updates and any number of reads
4 – Any number of updates and any number of reads; VSAM refreshes buffers for each direct request	4 – Any number of updates and any number of reads; VSAM refreshes buffers for each direct request

Notes For SHAREOPTIONS

- ❒ For SHAREOPTIONS 1, VSAM provides read and write integrity

- ❒ For SHAREOPTIONS 2, VSAM provides write integrity; read integrity is not guaranteed (for example, if a record is updated or deleted)

- ❒ For SHAREOPTIONS 3, VSAM provides no data set integrity

- ❒ For SHAREOPTIONS 4, VSAM will prevent updates or inserts that would change the high used RBA or the RBA of the high key data CI

Maintaining Integrity While Sharing VSAM Data Sets

Cross Region sharing, SHAREOPTIONS 2, 3, or 4

Issue ENQ / DEQ macros for each read and write request

For ESDS and RRDS do not allow secondary allocation (or provide mechanism to detect and communicate this happening to all users)

Issue "VERIFY" macro prior to "GET" for ESDS and RRDS

Establish way to indicate buffers are invalid and to communicate this to other users

Maintaining Integrity While Sharing VSAM Data Sets, 2

Cross System sharing, SHAREOPTIONS 3, 4

Run "VERIFY" AMS command before OPEN

Issue RESERVE / DEQ macros for each I/O request

All of the above (both this page and the previous page) require code written in Assembler (or calls to Assembler subroutines)

- **Or**

  Run an umbrella program that does this for you (CICS, IMS, etc.)

- **Or**

  Do not share data sets

  - ✗ Schedule jobs carefully
  - ✗ Code DISP=OLD in DD statements

REPRO Command

❒ **DEFINE CLUSTER defines a VSAM file, and reserves space for it**

❒ **The REPRO command loads records into the cluster**

❒ **The parameters for the REPRO command identify the input data location and the output cluster you are loading**

Optionally, you may REPRO only some of the records

✗ While creating test data, for example, you may want to build a small file (example of using SKIP and COUNT parameters coming up in a few pages)

REPRO Capabilities

$$\begin{tikzcd} \begin{array}{l} \{\ \ \text{SAM} \\ \ \ \mid \text{ISAM} \\ \ \ \mid \text{ESDS} \\ \ \ \mid \text{KSDS} \\ \ \ \mid \text{RRDS}\ \} \end{array} \arrow[r] & \begin{array}{l} \{\ \ \text{ESDS} \\ \ \ \mid \text{KSDS} \\ \ \ \mid \text{RRDS} \\ \ \ \mid \text{SAM}\ \} \end{array} \end{tikzcd}$$

Or

$$\begin{tikzcd} \text{LSDS} \arrow[r] & \text{LSDS} \end{tikzcd}$$

- ❒ **Output file must already be defined (VSAM) or allocated (SAM)**

- ❒ **If output is KSDS, input records must already be in correct key sequence**

- ❒ **May REPRO all or part of the input file**

- ❒ **For LSDS, must REPRO whole file, and both input and output must be LSDSs**

REPRO - Parameters

❒ **You identify the source data location by specifying either**

The INDATASET parameter and a fully-qualified data set name

✗ AMS will dynamically allocate the named file

✗ Warning: the allocation will be done with a default disposition of OLD

➢ Meaning no one else may access the file while the REPRO is going on

Or the INFILE parameter and a DDname

✗ You must then also supply a DD statement with that name in the JCL for the step

✗ But this DD statement may explicitly specify DISP=SHR so other users can access the source data at the same time

REPRO - Parameters, continued

❒ **You identify the target data location by specifying either**

The OUTDATASET parameter and a fully-qualified data set name (the cluster name)

- ✗ AMS will dynamically allocate the named file
- ✗ Again, the allocation will be done with a default disposition of OLD
  - ➢ But this is OK for the output file: no one else should be accessing the file while you are loading it

Or the OUTFILE parameter and a DDname

- ✗ You must then also supply a DD statement with that name in the JCL for the step, pointing to the target cluster
- ✗ This DD statement must explicitly specify DISP=OLD so other users cannot access the source data at the same time

REPRO - Examples

```
REPRO   INFILE(FILEX)   OUTFILE(FILEY)
```

Requires something like this in the JCL:

```
//FILEX   DD   DSN=SRT99U.TRAIN.SOURCE,DISP=SHR
//FILEY   DD   DSN=WQPVFI.RAIN.TARGET,DISP=OLD
```

Uses standard allocation for both input and output

```
REPRO   INDATASET(SRT99U.TRAIN.SOURCE)   –

      OUTDATASET(WQPVFI.RAIN.TARGET)
```

Requires no additional JCL in the step

Uses dynamic allocation

Note that this causes both data sets to be allocated with DISP=OLD

REPRO - Examples, 2

```
REPRO   IFILE(FILE1)   OFILE(FILE2)   SKIP (350)
```

Requires something like this in the JCL:

```
//FILE1   DD   DSN=SRT99U.TRAIN.SOURCE,DISP=SHR
//FILE2   DD   DSN=WQPVFI.RAIN.TARGET,DISP=OLD
```

Uses standard allocation for both input and output

Skips past first 350 records of input file

Note the abbreviation for INFILE is IFILE, and the abbreviation for OUTFILE is OFILE

```
REPRO   IDS(SRT99U.TRAIN.SOURCE)   –

      ODS  (WQPVFI.RAIN.TARGET)   COUNT(3000)
```

Requires no additional JCL

Again, uses dynamic allocation for both data sets

Copies only the first 3000 records

Note the abbreviation for INDATASET is IDS, and the abbreviation for OUTDATASET is ODS

REPRO - Examples, 3

```
REPRO   IFILE(FILE3)   –
      ODS(WQPVFI.RAIN.TARGET)   –
      SKIP(1000)   COUNT(5000)
```

Requires something like this in the JCL:

```
//FILE3   DD   DSN=SRT99U.TRAIN.SOURCE,DISP=SHR
```

Mix and match standard and dynamic allocation

- ✗ This is generally the best way to go
  - ➢ Allows input file to be held with SHR option so other users can read the data
  - ➢ Holds output file exclusively, so no one else can get to the data while it is being written out
  - ➢ Requires only one extra DD statement instead of two

Skips first 1000 records then copies next 5000 records

REPRO and the REPLACE Option

❒ **If the target data set of a REPRO function is a KSDS or an RRDS with records already present, you may specify REPLACE or NOREPLACE (the default)**

> **With REPLACE, if an incoming record has the same key or slot number as a record already in the file, the new record replaces the existing record**

> **With NOREPLACE, records having the same key or slot number cause the 'DUPLICATE RECORD' condition to be raised, and after some number of these, the process is halted**

> **In either case, records that do not duplicate existing records are simply placed into the correct location in the data set**

PRINT Command

- ❐ **The AMS PRINT command allows you to examine all or some of the records in a VSAM file**

- ❐ **Using the PRINT command, you specify**

  What object you want to print

  - ✗ Use INFILE with a DDNAME; the JCL must then include a DD statement with that DDname that points to the cluster
  - ✗ Or use INDATASET with the cluster name; as before, the cluster will be allocated with a disposition of OLD

  The format you want the records to be printed in

  - ✗ CHARACTER (abbreviation: CHAR) - write the records to the print line with no conversion
    - ➢ This can be a problem if data contains non-character fields such as packed decimal or binary integer data
  - ✗ HEX - write out each byte of input data as two hex digits
  - ✗ DUMP (the default) - Write out records in both CHAR and HEX, similar to a memory dump
    - ➢ This is useful to see both character string and non-character string data

  Which records you want printed

  - ✗ Use SKIP and COUNT parameters as with REPRO
  - ✗ Default is to print all records

PRINT Examples

```
PRINT   INFILE(DDANY)
```

Requires something like this in the JCL:

```
//DDANY   DD   DSN=EAGLE.CUREALL.APPLIC.TEST,DISP=SHR
```

Uses standard allocation

Will print the entire cluster in 'DUMP' format

```
PRINT   IFILE(DDANY)   OFILE(DDOUT)   CHAR
```

Requires something like this in the JCL:

```
//DDANY   DD   DSN=EAGLE.CUREALL.APPLIC.TEST,DISP=SHR
//DDOUT   DD   SYSOUT=E,COPIES=20
```

Uses standard allocation for the cluster

Will print the entire file in 'CHARACTER' format

Will produce 20 copies of the listing

PRINT Examples, 2

```
PRINT   IDS(EAGLE.CUREALL.APPLIC.TEST)   –
        SKIP(4550)   COUNT(20)
```

Requires no additional JCL

Dynamic allocation is used here for the cluster (so the DISP will be OLD)

Skip the first 4550 records and print the next 20 in 'DUMP' format

DELETE Command

❒ **Use the IDCAMS DELETE command to delete a VSAM data set from the system**

Remove the data (and index) components from the system

Remove the cluster, data (and index) catalog records for this data set from the catalog

❒ **The DELETE command has as its operand the name of the cluster you wish to delete**

❒ **If you want to delete many clusters, provide the cluster names in a list bounded by parentheses**

Names separated by blanks, commas, or both

Up to 100 entries may be specified

If only one cluster to delete, do not need the parentheses

❒ **In z/OS 1.11 and later, you can specify DELETE *entryname* MASK**

Where *entryname* is a filter mask that includes wildcards:

- ✗ % - a single mactch-any character
- ✗ * - a single level qualifier
- ✗ ** - any number of contiguous qualifiers

DELETE Examples

```
DELETE   TRANTOR.MASTER.FILE

DEL   FILMORE.MINOR.FILE   PURGE

DEL   (ANYMORE.MASTER.FILE   NEVRMORE.MISTER.FOIL  –
      CLEANTHE.FILE.FLOOR)

DELETE   IKJ55Z.*.TESTX

DEL ADMINS.TR%%%.* MASK

DELETE SCOMSTO.**.LIBRARY MASK
```

❒ **The abbreviation for DELETE is DEL**

❒ **The PURGE parameter overrides any date protection**

❒ **A DELETE of the cluster record automatically DELETEs the data component and index component and their corresponding catalog records**

❒ **Note that you never delete a VSAM cluster using DDNAME: you always name the cluster in an IDCAMS DELETE command, not in JCL**

We discuss an exception to this later

Computer Exercise: AMS and ESDS

Set up for the labs:

From ISPF option 6, issue the following command:

```
===> ex '________.e540.library(e540strt)  ex'
```

This will run a small dialog that prompts you for a high level qualifier to use for your data sets (it's set to use your TSO id as the default, and that is fine in most installations). This builds three data sets for you:

<hlq>.E540.CNTL	for JCL for the labs
<hlq>.E540.SOURCE	for your PL/I programs
<hlq>.E540.LOAD	for compiling and linking your programs

The dialog also puts some members in the first two libraries for future labs.

Now, in your E540.CNTL library, in member RUNAMS, code the JCL and AMS command statements necessary to perform the following functions in a single job:

1. Define an ESDS cluster called <hlq>.E540.ESDS, where <hlq> is your TSO logon id, or your team's logon id.

   Use the following information:

   records are fixed length, 100 bytes each

   there will be about 200 records

   name the data component <hlq>.E540.ESDS.DATA

   use a CI size of 4096

   place the data set on the _______ with volser _______.

2. Load your ESDS from the cataloged data set ____________________ (which we'll call INPUTA from now on). INPUTA is a sequential, non-VSAM data set.

3. Print the records in your ESDS (note do not print the records in INPUTA).

4. Delete your ESDS.

* Be sure to SHR INPUTA on the REPRO step!