These slides are based in part on materials provided by IBM’s Academic Initiative.
Data Sets

• A data set is a collection of logically related data records stored on one disk storage volume or a set of volumes.

• Data sets can be:
  – a source program
  – a library of macros
  – a set of data records used by a processing program

• Data sets can be printed or displayed on a terminal.

• The logical record is the basic unit of information used by a program running on z/OS (except z/OS UNIX)
Data Set Naming

HARRY.FILE.EXAMPLE.DATA

1°  2°  3°  4°

HLQ      LLQ
Naming Requirements

- Names must be upper case only and unique (kind of)
- Maximum of 44 characters; maximum of 22 name segments
- Each level qualifier can have 1 to 8 characters
- Each level qualifier must start with A-Z or specials “@#$”
- Remaining characters can also contain 0-9 and hyphen (”-“)
Allocating Data Sets

Allocation methods:

- ISPF data set panel, option 3.2
- TSO ALLOCATE command
- Access Method Services (more later)
- Using the Job Control Language (JCL) – next lecture

During allocation, you must specify certain details about the size and structure of the data set.
ISPF-based Allocation

![ISPF Allocation Screen](image)
Data Set List Utility

Option -->

blank Display data set list
V Display VTOC information
P Print data set list
PV Print VTOC information

Enter one or both of the parameters below:
Dname Level ... zibm030
Volume serial .

Data set list options
    Initial View . . 1  1. Volume Enter "/" to select option
                    2. Space /
                    3. Attrib /
                    4. Total /

When the data set list is displayed, enter either:
"/" on the data set list command field for the command prompt pop-up,
an ISPF line command, the name of a TSO command, CLIST, or REXX exec, or
"=" to execute the previous command.
# ISPF-based Allocation

In this section, we will discuss how to allocate a new data set using ISPF. The ISPF-based allocation allows for the creation of data sets with specific attributes, such as the management class, storage class, volume serial, device type, data class, space units, average record unit, primary quantity, secondary quantity, directory blocks, record format, record length, block size, data set name type, expiration date, and allocation of multiple volumes or blank.

When specifying the data set name type, you can choose from `LIBRARY`, `HFS`, `PDS`, or `BLANK`. The expiration date can be specified in the format `YY/MM/DD` or `YYYY/MM/DD` in Julian form.

The following command is used to allocate a new data set:

```
Command ==>) ALLOCATE NEW DATA SET
```

- **Data Set Name**: ZOS.TEST1
- **Management class**: (Blank for default management class)
- **Storage class**: (Blank for default storage class)
- **Volume serial**: VPARC1 (Blank for system default volume)**
- **Device type**: (Generic unit or device address)**
- **Data class**: (Blank for default data class)
- **Space units**: CYLINDER (BLKS, TRKS, CYLS, KB, MB, BYTES or RECORDS)
- **Average record unit**: M, K, or U)
- **Primary quantity**: (In above units)
- **Secondary quantity**: (In above units)
- **Directory blocks**: (Zero for sequential data set) *
- **Record format**: Tb
- **Record length**: R0
- **Block size**: 
- **Data set name type**: (LIBRARY, HFS, PDS, or blank) *
- **Expiration date**: YY/MM/DD, YYYY/MM/DD in Julian form

Enter `/` to select option DDDD for retention period in days
- Allocate Multiple Volumes or blank)

(*) Specifying LIBRARY may override zero directory block

(** Only one of these fields may be specified)
Records

- A record is the equivalent of a line in a textfile
- Data sets do not have EOL characters
- Records can be fixed-length or variable-length
- Record length (for fixed-length records) is referred to as the LRECL
- Indentation (offset of text in the record) often matters
Blocks

- Multiple records are grouped in a block
- Logical records are usually the smallest amount of data to be processed by applications
- Blocks are usually the unit of data read from or written to DASD
**Data set record formats**

**F**
- Fixed records.
- Fixed blocked records. \( \text{BLKSIZE} = n \times \text{LRECL} \)

**FB**
- Block
- Fixed blocked records. \( \text{BLKSIZE} = n \times \text{LRECL} \)

**V**
- Record
- Block
- Variable records.

**VB**
- Record
- Block
- Variable blocked records. \( \text{BLKSIZE} \geq 4 + n \times \text{largest LRECL} \)

**U**
- Record
- Block
- Undefined records. No defined internal structure for access method.

Record and block descriptors words are each 4 bytes long.
Example Data Set Specification

DSORG=PS
RECFM=FB
LRECL=80
BLKSIZE=27920

DATASET.TEST.SEQ1
The Volume Table of Contents (VTOC)

VTOC:

- Lists the data sets on a volume
- Lists the free space on the volume
The Volume Table of Contents (VTOC)
Extents

LABEL (volser)

VTOC

MY.DATA
YOUR.DATA
free space

tracks

tracks

tracks

Extents
Catalogs

A catalog associates a data set name with the volume on which the data set is located; locating a data set requires:

- Data set name
- Volume name
- Unit (volume device type)

Typical z/OS systems include a master catalog and numerous user catalogs.
Creating a Catalog
Catalog Structure

- Master Catalog
  - Data Set - SYS1.A1
  - or
  - HLQs (alias) IBMUSER...USER

- USERCAT.IBM
  - User Catalog
    - Data Set with HLQ = IBMUSER
      - volume (wrk002)
        - unit (3390)
      - IBMUSER.A2
      - IBMUSER.A3

- USERCAT.COMPANY
  - User Catalog
    - Data Set with HLQ = USER
      - volume (012345)
        - unit (tape)
      - USER.TAPE.A1
Defining an Alias

ISPFF Command Shell

Enter TSO or Workstation commands below:

```plaintext
>>> def alias (name('zos') relate('catalog.zos'))
```

Place cursor on choice and press enter to Retrieve command

```plaintext
=> def ucat (name('catalog.zos') volume(vpmvse) cylinder(2 1))
=>
=>
=>
=>
=>
=>
=>
=>
```
Locating a data set in z/OS
Inspecting a Catalog
Cataloged and Uncataloged Data Sets

- Uncataloged reference
  // DD
  DSN=PAY.D1
  DISP=OLD
  UNIT=3390
  VOL=SER=MYVOL

- Cataloged reference
  // DD
  DSN=PAY.D2
  DISP=OLD
Types of Data Sets

• Sequential Data Set: written and read in sequential order

• Partitioned Data Set (PDS or library):
  – Collection of a directory and sequential data sets (called members)
  – Member names can be 8 bytes long

• VSAM – both a type of data set and an access method
PDS vs. PDSE

PDS Extended (PDSE) is an extension of PDS with the following advantages:

- Space reclaimed automatically when a member is deleted (PDS has a compaction utility)
- Flexible size
- Can be shared
- Faster directory searches
Comparison of MVS Data Sets and UNIX HFS

MVS Data Sets
- Master Catalog
  - User Prefix
    - SMITHA
  - Partitioned Data Set
    - SMITHA.TEST1.C
  - Sequential Data Set
    - VSAM Data Set
    - Member
      - SMITHA.TEST1.C
        - PGMA
        - PGMB

Hierarchical File System
- Root/
  - User Directory
    - /u/smitha
  - Subdirectory
    - /u/smitha/test1
    - File
      - /u/smitha/test1
        - pgma.c
        - pgmb.c
Access Methods

- An *access method* defines the technique used to store and retrieve data
- This definition includes system-provided programs and utilities to define and process data sets.
Commonly Used Access Methods

**QSAM**  Queued Sequential Access Method (for most simple data sets)

**BSAM**  Basic Sequential Access Method (for special cases)

**BDAM**  Basic Direct Access Method (becoming obsolete)

**BPAM**  Basic Partitioned Access Method (for libraries)

**VSAM**  Virtual Storage Access Method (for complex applications)
VSAM

VSAM provides more complex functions than other common disk access methods; VSAM knows four record formats:

- Key Sequence Data Set (KSDS) – most commonly used; like a hash table
- Entry Sequence Data Set (ESDS) – for simple sequential access
- Relative Record Data Set (RRDS) – access records by number
- Linear Data Set (LDS) – like UNIX files (and used by zFS)
Simple VSAM Control Interval

VSAM Control Intervals are the basic unit for DASD reads and writes used by VSAM. They contain records and metadata:

| R1 | R2 | R3 | free space in CI | R | D | F | R | D | F | C | I | D | F |
|----|----|----|------------------|---|---|---|---|---|---|---|---|---|---|---|

Record Descriptor Fields
VSAM Index Structure

Index Set

- Forward horizontal pointer at same level
- Vertical pointers to next lower level index records
- Just one CI in the top
Key Sequence Data Set (KSDS)
Entry Sequence Data Set (ESDS)
Relative Record Data Set (RRDS)
Linear Data Set (LDS)
# Data Set List Utility: Listing

![Screenshot of the Data Set List Utility](image)

<table>
<thead>
<tr>
<th>Command</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIBM030</td>
<td>ALIAS</td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.EXEC</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.ISPF.ISPAPPRF</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.JCL</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.LOAD</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.OUTPUT</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.SOURCE</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.SPFLOGI.LIST</td>
<td></td>
<td>VPARC1</td>
</tr>
</tbody>
</table>

End of Data Set list
Data Set List Utility: Listing (More)

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Dorg</th>
<th>Recfm</th>
<th>Lrecl</th>
<th>Blksz</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIDM030</td>
<td>PS</td>
<td>F</td>
<td>170</td>
<td>170</td>
</tr>
<tr>
<td>ZIDM030_DATA</td>
<td>P0-E</td>
<td>FB</td>
<td>80</td>
<td>32720</td>
</tr>
<tr>
<td>ZIDM030_EXEC</td>
<td>P0-E</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
</tr>
<tr>
<td>ZIDM030_ISPF.ISPPROF</td>
<td>P0-E</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
</tr>
<tr>
<td>ZIDM030_JCL</td>
<td>P0-E</td>
<td>FD</td>
<td>80</td>
<td>000</td>
</tr>
<tr>
<td>ZIDM030_LOAD</td>
<td>P0-E</td>
<td>U</td>
<td>0</td>
<td>6144</td>
</tr>
<tr>
<td>ZIDM030_OUTPUT</td>
<td>P0-E</td>
<td>FB</td>
<td>80</td>
<td>800</td>
</tr>
<tr>
<td>ZIDM030_SOURCE</td>
<td>P0-E</td>
<td>FB</td>
<td>80</td>
<td>800</td>
</tr>
<tr>
<td>ZIDM030.SPFL061.LIST</td>
<td>PS</td>
<td>VA</td>
<td>125</td>
<td>125</td>
</tr>
</tbody>
</table>

*************** End of Data Set list ***************
Data Set List Utility: Commands

![Screenshot of the Data Set List Utility](image)

<table>
<thead>
<tr>
<th>Command</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIBM030_DATA</td>
<td>Info - I</td>
<td>#ALIAS</td>
</tr>
<tr>
<td>ZIBM030_EXEC</td>
<td>VPARC1</td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.ISPF.ISPPROF</td>
<td>VPMVSE</td>
<td>VPMVSE</td>
</tr>
<tr>
<td>ZIBM030.JCL</td>
<td>VPARC1</td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030_LOAD</td>
<td>VPARC1</td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030_OUTPUT</td>
<td>VPARC1</td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.SOURCE</td>
<td>VPARC1</td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM030.SPFLOGI.LIST</td>
<td>VPMVSC</td>
<td>VPMVSC</td>
</tr>
</tbody>
</table>

*End of Data Set list*
Data Set List Utility: Info

```
Command ==> 

Data Set Name . . . : 21BM030.DATA

General Data
Volume serial . . . : VPARC1
Device type . . . . : 3390
Organization . . . : PS
Record format . . . : F
Record length . . . : 170
Block size . . . . . : 170

Data Set Information

Current Allocation
Allocated cylinders . . . : 2
Allocated extents . . . : 1

Current Utilization
Used cylinders . . . : 1
Used extents . . . : 1

Creation date . . . . : 2007/03/24
Referenced date . . . : 2007/03/29
Expiration date . . . : ***None***
```
Data Set List Utility: Volume View

Enter one or both of the parameters below:
- Dename Level
- Volume serial

Data set list options
- Initial View
  - Volume
  - Space
  - Atrib
  - Total

Enter "/" to select option
- Confirm Data Set Delete
- Confirm Member Delete
- Include Additional Qualifiers
- Display Catalog Name

When the data set list is displayed, enter either:
- "/" on the data set list command field for the command prompt pop-up,
- an ISPF line command, the name of a TSO command, CLIST, or REXX exec, or
- "=" to execute the previous command.
## Data Set List Utility: Volume DS List

![Screenshot of Data Set List Utility](image)

<table>
<thead>
<tr>
<th>Command</th>
<th>Message</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNTS.DSNDDB.ACCCTS.ACCTS.I0001.A001</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ACCOUNTS.DSNDDB.ACCCTS.XCLIENT.I0001.A001</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>COUPLE.ATR.SVSCPLEX.ARCHIVE.A00000000.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>COUPLE.ATR.SVSCPLEX.DELAYED.OR.A00000000.D</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>COUPLE.ATR.SVSCPLEX.RESTART.A00000000.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>COUPLE.ATR.SVSCPLEX.RESTART.SVSCPLEX.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>COUPLE.ATR.SVSCPLEX.RM.DATA.SVSCPLEX.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>DB2CHECK.WORK</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>IBMUSER.DOIR.D</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>IBMUSER.DOIR.I</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>IBMUSER.ISPF.ISPPROF</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>SYS1.VIOCIX.VPARC1</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>SYS1.VVDS.VVPARCI</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM001.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM001.JCL</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM001.LOAD</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM001.OUTPUT</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM001.SOURCE</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM002.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM002.JCL</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM002.LOAD</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM002.OUTPUT</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM002.SOURCE</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM003.DATA</td>
<td></td>
<td>VPARC1</td>
</tr>
<tr>
<td>ZIBM003.ISPF.ISPPROF</td>
<td></td>
<td>VPARC1</td>
</tr>
</tbody>
</table>
What Is DFSMS?

Data Requirements

<table>
<thead>
<tr>
<th>Creation</th>
<th>Usage</th>
<th>Retention</th>
</tr>
</thead>
<tbody>
<tr>
<td>What does it look like?</td>
<td>What are the storage service requirements?</td>
<td>What are the management service requirements?</td>
</tr>
</tbody>
</table>

Logical

Data Class

Physical

Storage Administrator

Storage Groups

End User
Automatic Class Selection (ACS)

ACS uses coded criteria to determine allocation parameters:

1. Data Class (RECORG, RECFM, LRECL, PDS/PDSE, SPACE, ...)

2. Management Class (migration, backup frequency, automatic deletion, ...)

3. Storage Class

4. Storage Group (device media)

ACS uses the DD’s DATACLS, MGMTCLAS, STORCLAS, DSN and DISP operands for criteria determination.
Copying between Data Sets and UNIX Files

In order to copy files from or to UNIX, use the following commands in the ISPF Command Shell:

OGET '/u/$USER/file' KC02292.FOO(BAR)
OPUT KC02292.FOO(BAR) '/u/$USER/file'
Questions