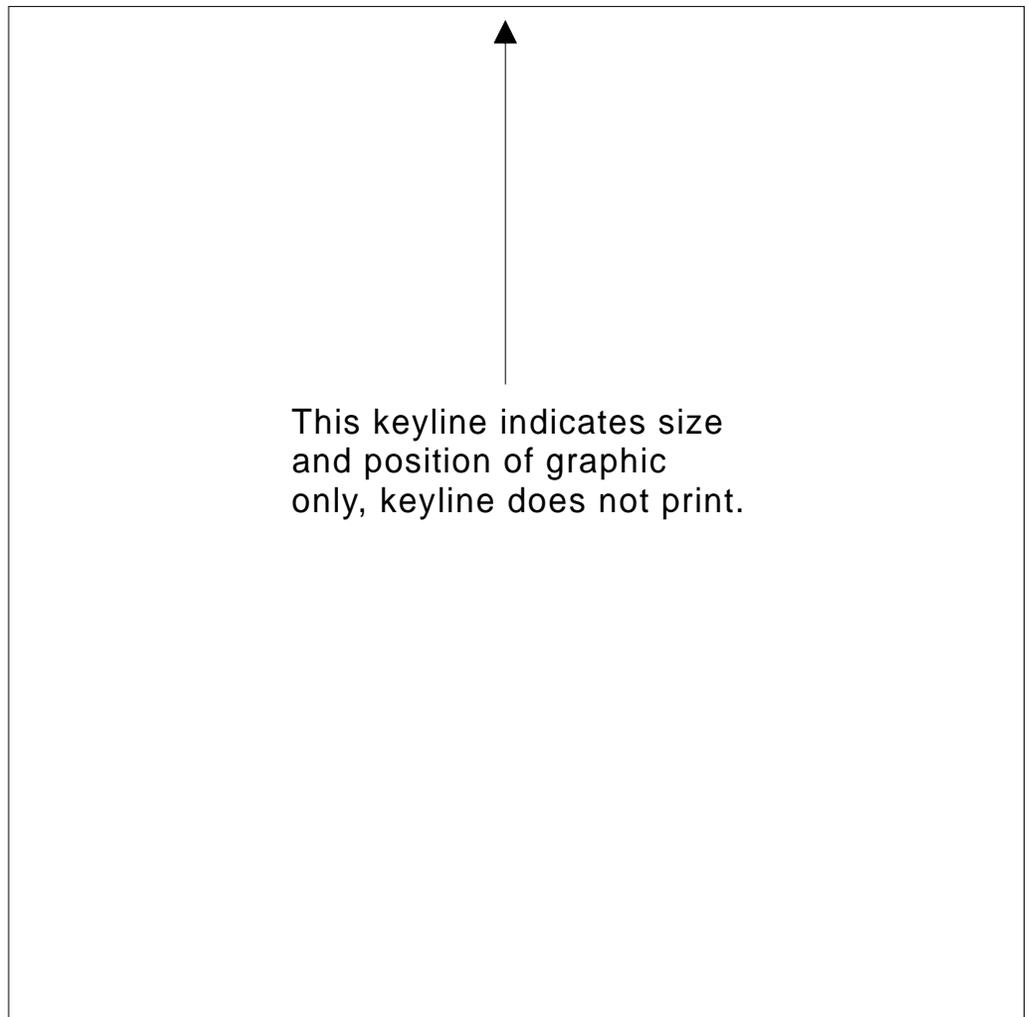


Managing Data





MVS/ESA Storage Management Library

SC26-3124-02

Managing Data

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

Third Edition (June 1994)

This edition replaces and makes obsolete the previous edition, SC26-3124-01.

The technical changes for this edition are summarized under "Summary of Changes," and are indicated by a vertical bar to the left of a change.

This edition applies to Version 1 Release 2 of DFSMS/MVS, Program Number 5695-DF1, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
Department M29, Building 86
5600 Cottle Road
SAN JOSE, CA 95193-0000
U.S.A

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1986, 1994. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Programming Interface Information	vii
Softcopy Agreement	viii
Trademarks	ix
Summary of Changes	xi
Third Edition, June 1994	xi
Product Highlights, Version 1 Release 2	xi
Publication Updates, Version 1 Release 2	xiv
Second Edition, March 1993	xiv
Product Highlights, Version 1 Release 1	xiv
First Edition, July 1992	xvii
Preface	xix
Required Publications	xix
The Storage Management Library	xix
Related Publications	xx
Referenced Publications	xx
Chapter 1. Introducing System-Managed Storage	1
The DFSMS Environment for MVS/ESA	1
Benefits of System-Managed Storage	2
Managing Data with SMS	4
Using SMS Classes and Groups	4
Using Aggregate Groups	11
Using Automatic Class Selection Routines	11
Defining the Storage Management Subsystem Configuration	13
Coexistence Considerations	14
Sharing System-Managed and Non-System-Managed Data	14
Sharing Catalogs in Multisystem Environments	14
VTOC Compatibility Between MVS/XA and MVS/ESA Systems	16
Sharing DASD Between MVS/ESA and MVS/XA Systems	16
Sharing DASD Between MVS/ESA Systems	17
Using System Group Name with Pre-DFSMS/MVS 1.2.0 Systems	18
Using Different Releases of DFDSS, DFHSM, and DFSMS/MVS Programs	19
Hardware Considerations	19
The Responsibilities of a Storage Administrator	20
Working with Your User Groups	20
Implementing the System-Managed Storage Strategy	20
Chapter 2. Establishing Installation Standards	21
Identifying Data Types	21
Data That Can Be System Managed	21
Data Types That Cannot Be System Managed	23
Developing Naming Conventions	24
Using Qualifier Levels	24
Minimizing the Effect on Existing Naming Conventions	29
Improving Catalog Performance	30
Using Multilevel Aliases	31
Specifying Catalog Performance Requirements	32

Simplifying JCL	34
JCL Keywords Used in the DFSMS Environment	34
Sample Jobs Using the JCL Keywords	36
Chapter 3. Managing Data Allocation	39
Establishing Allocation Standards	39
Using Data Class to Standardize Data Allocation	39
Ensuring Device Independence	41
Removing Volume Serial Number Dependencies	41
Using Device Independent Allocation	45
Requesting System-Determined Block Size	46
Allocating Partitioned Data Sets Extended	47
Converting Existing PDSs to PDSEs	49
Allocating New PDSEs	52
Identifying PDSEs	53
Processing Generation Data Groups	54
Defining and Allocating Generation Data Sets	54
Cataloging Generation Data Sets	55
Allocating Compressed Data	56
Selecting Compression Candidates	56
Allocating and Compressing the Data Sets	56
Chapter 4. Managing Space Use	59
Establishing Space Management Standards	59
Managing Space with Data Class	59
Managing Space with Management Class	59
Deleting Data	62
Automatic Deletion	63
Manual Deletion	66
Releasing Unused DASD Space	68
Automatic Release	68
Manual Release	70
Deallocating Data Sets at Close Time	71
Migrating Infrequently Referenced Data	71
The Data Migration Process	71
Automatic Migration	74
Manual Migration	78
Space Management During Migration	79
Chapter 5. Improving I/O Performance	83
Establishing Performance Objectives	83
Using Storage Class to Set Performance Goals	83
Using Dynamic Cache Management	85
Selecting Dynamic Cache Candidates	85
Implementing Dynamic Cache Management	86
Allocating Extended Sequential Data Sets	87
Determining Extended Sequential Data Set Candidates	88
Allocating Sequential Data Sets in Extended Format	88
Allocating Temporary Data Sets	91
Allocating Data with Lower Performance Demands	91
Selecting Lower Performance Candidates	92
Improving Data Access Rate	92
Improving the Performance of Shared Data in a Sysplex	92

Chapter 6. Backing Up Data	93
Developing a Backup Plan	93
Preparing for General Backups	93
Preparing for Disaster Recovery Backups	95
Establishing Availability Standards	97
Using Management Class Backup Attributes	97
Using the Storage Class Availability Attributes	97
Backing Up Data With Standard Availability Requirements	98
Determining Backup Frequency and Number of Versions	98
Automatic Data Backup	101
Manual Data Backup	107
Inline Backup	108
Backing Up Critical Data	108
Automatic Critical Data Backup	108
Manual Critical Data Backup	112
Archiving Data and Storing Vital Records	113
Archives	113
Vital Records	115
Performing Aggregate Backups	117
Preparing for Aggregate Backup	117
Issuing the ABACKUP Command	122
Managing ABARS Output Files	126
Chapter 7. Recovering Data	129
Developing a Recovery Plan	129
Recovering User Data	130
Data Set Recovery	130
Database Data Recovery	132
Object Data Recovery	132
Recovering Data Critical to the Installation	133
Catalog Recovery	133
Control Data Set Recovery	134
RACF Data Set Recovery	135
Critical Database Data Recovery	135
Retrieving Vital Records and Archived Data	135
Performing Aggregate Recovery	136
Preparing for Aggregate Recovery	136
Issuing the ARECOVER Command	140
Chapter 8. Protecting Data	145
Implementing a Protection Policy	145
Determining Security Categories	146
Defining Data Security Profiles	147
Using a Default Authorization Table	148
Controlling Access With RACF	149
Using RACF Profile Fields	149
Protecting Control Data Sets	150
Controlling the Use of ISMF Functions	152
Controlling the Use of Storage Management Subsystem Functions	153
Controlling Storage Management Subsystem Classes and Groups	154
Protecting User Data	156
Controlling the Use of Aggregate Backup and Recovery Commands	158
Defining Access Profile for the Storage Administration Group	158

Chapter 9. Monitoring and Reporting for Storage Management	159
Monitoring Space Use	159
Collecting Information on Data	159
Collecting Information on Volumes	161
Producing Space Use Reports	163
Measuring DASD and DFSMSHsm-Owned Tape Capacity	171
Collecting Capacity Information	172
Producing Capacity Planning Reports	173
Monitoring Performance	175
Collecting I/O Statistics	176
Reporting and Interpreting I/O Statistics	177
Collecting OAM Transaction Statistics	178
Creating and Interpreting RMF Transaction Reports	179
Collecting Cost Accounting Information	183
Determining Service Charges	183
Determining Storage Charges	185
Analyzing Tape Use to Reduce Tape Mounts	187
Using the Volume Mount Analyzer	187
Tracking Tape Mounts	190
Monitoring Tape Media Use	200
Appendix A. VSAM Data Set Allocation Parameters	209
Appendix B. DCOLLECT Record Types	211
Abbreviations	215
Glossary	217
Index	225

Notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program which does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, 208 Harbor Drive, Stamford, CN 06904, U.S.A.

Programming Interface Information

This book is intended to help you to do implementation of system-managed storage.

This book also documents General-Use Programming Interface and Associated Guidance Information and Product-Sensitive Programming Interface and Associated Guidance Information.

General-Use programming interfaces allow the customer to write programs that obtain the services of DFSMS/MVS.

General-Use Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

General-Use Programming Interface

General-Use Programming Interface and Associated Guidance Information...

End of General-Use Programming Interface

Product-Sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-Sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-Sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Product-Sensitive Programming Interface

Product-Sensitive Programming Interface and Associated Guidance Information...

End of Product-Sensitive Programming Interface

Softcopy Agreement

For online versions of this book, we authorize you to:

1. Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy; and
2. Transfer the original unaltered copy of the documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the Program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for payment of any taxes, including personal property taxes, resulting from this authorization.

THERE ARE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine readable documentation.

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries:

ACF/VTAM	ADSTAR
AIX	AIXwindows
BookManager	CICS
CICS/ESA	DB2
DATABASE 2	DFSMS
DFSMS/MVS	DFSMS/VM
DFSMSdfp	DFSMSdss
DFSMSshsm	DFSMSrmm
ECKD	Enterprise Systems Architecture/390
Enterprise Systems Architecture/370	ES/3090
ES/9000	ESA/370
ESA/390	ESCON
GDDM	Hardware Configuration Definition
Hiperbatch	Hipersorting
Hiperspace	IBM
ImagePlus	IMS/ESA
MVS/DFP	MVS/ESA
MVS/SP	MVS/XA
NetView	OpenEdition
OS/2	PR/SM
PS/2	QMF
RACF	SAA
S/390	3090
VSE/ESA	VTAM

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies:

CADAM	CADAM, Inc.
UNIX	X/Open Company, Ltd.

Summary of Changes

The summary of changes for DFSMS/MVS provides you with a highlight of the release as well as specific changes for this book.

Third Edition, June 1994

Product Highlights, Version 1 Release 2

The following section highlights the new enhancements that are part of DFSMS/MVS Version 1 Release 2.

Programming Support

Backup-while-open: Provides the capability of backing up CICS VSAM File Control data sets under the Storage Management Subsystem (SMS) while they are open for update. Improvements include data security and integrity enhancements for SMS-managed CICS VSAM data sets.

DFSMSShsm RECYCLE Enhancements: Enhances the DFSMSShsm RECYCLE command to improve performance and usability.

Tapes containing the lowest percentages of valid data (based on cartridge type, standard or enhanced capacity) are processed first, which provides the highest number of tapes being returned for reuse early in the recycling process.

Up to 15 recycle tape-processing tasks can run simultaneously on a processor, with three tape buffers being used for improved throughput.

Recycle processing can be set to automatically quiesce when a predefined net number of tapes are returned to scratch.

DFSMSShsm Tape I/O Improvements (Multiple Tape Buffers): Enhances the tape mount management (TMM) strategy with the following improvements to tape I/O processing:

- When the DFSMSdss data mover is used for level 0 direct-to-tape migration and backup operations, multiple output buffers are obtained in extended private storage (above the 16-megabyte line) and the number of in-storage moves are reduced when software compaction is not used.

DFSMSrmm Enhancements: DFSMSrmm provides this new support:

- Support for the IBM 3495-M10 Tape Library Dataserver
- Support for the new OAM CBRUXVNL installation exit
- Support for 4-digit device numbers
- Support for OAM object tape
- A new DFSMSrmm installation exit to provide support for returning volumes to scratch status
- New bypass label processing support
- Refresh of installation exits with DFSMSrmm active

- If DFSMSrmm is not licensed for use, the DFSMSrmm supplied version of the OAM installation exits are executed once only
- DFSMSrmm installation exits are no longer automatically disabled on failure

Distributed FileManager/MVS: Allows users from remote systems to access data on an MVS/ESA system. DFM provides Inplace access to remote files from applications running in client systems; substantially reducing the need for Upload/Download scenarios. DFSMSdss saves Distributed Data Management (DDM) attributes associated with a data set and preserves those attributes during copy, move, backup, and restore operations.

Expiration Date Override: Provides the capability of overriding the expiration date when deleting an SMS-managed DASD data set—via methods other than management class—using JCL DD statements, IEHPROGM SCRATCH, or SVC 99. While management class is still the recommended method for deleting SMS-managed data sets, the expiration date override capability gives the user the ability to delete a data set without requiring the operator to authorize the deletion.

OAM Object Tape Support: Tape is now a full-fledged member of the OAM storage hierarchy. Objects can have their primary and/or backup copy stored on tape cartridges. Objects can transition up and down the hierarchy to and from tape cartridges. IBM cartridge system tapes can be inside of or outside of an automated or manual tape library (3495 ATL/MTL, 3494).

OAM RMF Statistics: Collects transaction information from OAM to generate performance-related statistical reports. The reports can be used to monitor and improve OAM's performance.

OpenEdition MVS: Creates a hierarchical file system within a new kind of MVS data set called a hierarchical file system data set (an HFS data set). The major feature consists of the ability to define multiple directories in an HFS data set and byte access of the data files. Additional constructs called symbolic links, FIFO files, and character special files, are also supported. HFS files are accessible through the POSIX-compliant file system calls of OpenEdition MVS.

Sequential Access Method Compression: Gives users of extended sequential data sets a transparent data compression capability. This support improves DASD space utilization; improves cache utilization; reduces I/O needed to reference data; reduces channel, controller, and device busy time necessary to process a given amount of data; and exploits the new hardware compression facility, if it is available. If it is not available, a software equivalent (available with the appropriate level of MVS/SP) is used.

Enhanced Volume Selection for the Storage Management Subsystem: This support allows you to mix devices of varying performance characteristics, (that have the same device geometry), within one pool storage group. SMS selects the volumes that more closely match a user's requested performance and accessibility characteristics (as specified in the storage class).

System Group Name Support: Eliminates the eight system limitation in MVS/SP JES2 environments. Although an SMS configuration for an SMS complex can still only contain eight names, those names can now be a combination of system names and system group names. A system group is a sysplex minus any systems that are explicitly defined in the base configuration.

Four-Digit Device Numbers: This support, in conjunction with the appropriate MVS/SP level, lets you attach more devices than previously allowed. Device numbers can now be four digits, removing the previous limitation of 4 096 device numbers. The MVS/ESA architecture supports up to 65 536 device numbers. The actual number of devices attachable to your system depends on the hardware configuration of your processor and I/O devices, and the amount of virtual storage available below 16 megabytes.

Virtual Storage Constraint Relief (VSCR): Increases virtual storage available to DFSMSHsm. A greater number of tasks can be run simultaneously, and virtual storage problems relieved, by bringing the DFSMSHsm load module above the 16-megabyte line in extended private storage.

VSAM KSDS Compression: Provides VSAM users with a new extended key-sequenced data set (KSDS) that supports data compression. Data compression improves DASD space utilization; improves cache utilization; reduces I/O needed to reference data; reduces channel, controller, and device busy time necessary to process a given amount of data; and exploits the new hardware compression facility where it is available. If it is not available, a software equivalent (available with the appropriate level of MVS/SP) is used.

VSAM Partial Space Release: Provides the capability of releasing allocated but unused space in extended format VSAM KSDS data sets. It is supported with a management class parameter and JCL.

Data Class Attribute (Additional Volume Amount): ISMF has been enhanced to add a new data class attribute called ADDITIONAL VOLUME AMOUNT. This attribute specifies whether primary or secondary allocation amounts are to be used when a extended format VSAM KSDS data set extends to a new volume.

Tape Library Enhancements

3494/3495 Partitioning: Multiple Storage Management Subsystem complexes can now access a 3494/3495 without using a shared ACDS, COMMDS and TCDB. Partitioning support allows you to partition a 3494/3495 into multiple logical 3494/3495s, a partition per Storage Management Subsystem complex. The volumes cannot be shared across the partitions.

OAM Volume Not In Library Exit: This installation exit gives installations the opportunity to insert a volume into a tape library (either manual tape library dataserver or automated tape library dataserver) during job step setup processing, device allocation processing, or library mount processing.

Device Support

3494 Tape Library Dataserver: The 3494 is an automated tape library that uses the same architecture as the 3495. A 3494 configuration can have one drive and 240 tape cartridges, up to a maximum of eight drives and 3040 cartridges. It attaches to the AS/400, RISC System/6000, and ES/9000 processor families.

3990 Model 6: The 3990 Model 6 is a new DASD control unit consisting of new electronics and licensed internal code. The 3990 Model 6 provides the following enhancements:

- The capability to establish up to 128 logical channel paths using up to 16 physical channel adapters.

- Aggregate data rate increase of up to four times that of the 3990 Model 3.
- Serviceability improvements such as the capability of synchronizing timers with the attaching hosts.

3495-M10 Tape Library Dataserver: Extends the scope of tape library support, introduced in DFSMS/MVS Version 1 Release 1, to include mountable tape cartridges in an SMS-managed manual tape library dataserver. A manual tape library dataserver is an installation-defined set of tape drives and the set of volumes that can be mounted on the drives. An SMS-managed manual tape library dataserver consists of an IBM 3495 Model M10 tape library and one to four IBM 3490 magnetic tape subsystems.

3995 Models 113 and 133: Support has been added to the OAM component of DFSMS/MVS for the IBM 3995 Models 133 and 113 optical library dataservers. The Model 133 is a channel-attached optical disk library with multi-function optical disk drives (WORM and read and write). The Model 113 is an expansion unit that attaches directly to the Model 133.

Publication Updates, Version 1 Release 2

Following are the changes that are specific to this book.

- Technical changes to reflect new features and functions
- Editorial changes to improve readability and retrievability

Second Edition, March 1993

This edition describes additional product enhancements to DFSMS/MVS 1.1.0.

The following section describes the highlights of DFSMS/MVS Version 1 Release 1.

Product Highlights, Version 1 Release 1

DFSMS/MVS Version 1 Release 1 integrates and expands the functions previously available in MVS/DFP Version 3 (5665-XA3), the Data Facility Hierarchical Storage Manager (DFHSM) Version 2 (5665-329), and Data Facility Data Set Services (DFDSS) Version 2 (5665-327). The functions of these previous offerings and major new functions are contained in the following four DFSMS/MVS functional components:

- DFSMSsdfp—Provides storage, data, program, and device management functions
- DFSMSsdss—Provides data movement, copy, backup, and space management functions
- DFSMSshsm—Provides backup, recovery, migration, and space management functions
- DFSMSrmm—Provides management functions for removable media such as tape cartridges, reels, and optical volumes.

Programming Support

Removable Media Management: The DFSMSrmm functional component helps you more effectively manage the data that your business stores on removable media, including tape reels, cartridges, and other media identified to it. DFSMSrmm provides a central, online inventory of the volumes and shelf space both on-site and at off-site storage locations. You can use DFSMSrmm to automate many of your tape management tasks.

System-managed Tape Cartridges and Tape Libraries: Tape cartridges can now be system-managed. You can define tape libraries and storage groups using ISMF. These tape libraries can contain 3490 or 3490E tape devices. The associated storage groups can contain IBM Cartridge System Tape or IBM Enhanced Capacity Cartridge System Tape. You can use the IBM 3495 Tape Library Dataserver to automate tape cartridge mounts and storage for 3490 and 3490E tape devices.

Volume Mount Analyzer: The volume mount analyzer is a new tool that uses SMF records to provide a comprehensive analysis of tape data. It also identifies suitable data set candidates for use with the tape mount management technique.

Concurrent Copy: Concurrent copy allows DFSMSdss to backup or copy data sets without locking users out of the data set until the copy is complete. The backup or copy appears as if it were taken instantaneously. You can control the use of concurrent copy using the new storage class attribute, ACCESSIBILITY, and the new management class attributes, BACKUP COPY TECHNIQUE and ABACKUP COPY TECHNIQUE. To use concurrent copy to copy data, the data must reside on a DASD volume attached through an IBM 3990 Model 3 Extended Platform Storage Control or an IBM 3990 Model 6.

Program Library Support for PDSEs: DFSMS/MVS provides the program management binder and program management loader for enhanced program management. The binder relaxes many of the limitations of the MVS/DFP linkage editor and loader, and supports a new executable unit called a *program object*. The program object eliminates many of the restrictions of the traditional load module. The loader loads both program objects and traditional load modules.

Striping Support for Sequential Data Sets: A new type of data set, extended sequential, can be defined and accessed with BSAM and QSAM. An extended sequential data set consists of stripes which reside on separate DASD volumes. When data is written to a striped data set, the data is interleaved among the stripes. A striped data set differs from a non-extended multivolume data set in that data is written to one volume of a multi-volume data set until the space for the data set on the volume is filled. All extended sequential data sets must be system-managed.

You can use striping to transfer data between DASD and memory at a faster rate than an individual DASD can handle. It is a performance enhancement mainly used for large, physical sequential data sets with high I/O activity. Whether a physical sequential data set is striped is mostly transparent to an application program. Changes have been made to data class and storage class to allow users to define extended sequential data sets. This type of data set requires the 3990-3 with Extended Platform, or 3990-6, or cached 9343 Storage Control, and ESCON channel attachment.

31-bit support for BSAM, QSAM, BPAM, and BDAM: The BDAM, BPAM, BSAM, and QSAM access methods support 31-bit addresses for most macros and allow data buffers to reside above the 16MB line for non-striped and extended sequential data sets, allowing for possible performance enhancements.

Aggregate Backup and Recovery Support (ABARS): You can use Storage Management Subsystem management classes to more easily manage your aggregate groups, or groups of data sets that you want backed up together for disaster recovery. DFSMS/MVS management class attributes include new retention attributes for aggregate groups.

Note: Because these requirements are determined by the management class of the aggregate group, the aggregate group application does not contain the fields that MVS/DFP contained for expiration date, destination, or toleration enqueue failure. Thus, if you are migrating to DFSMS/MVS from MVS/DFP, you should either print or display and write down the values of the three deleted aggregate group fields before you IPL the system with DFSMS/MVS.

Aggregate backup and recovery support has also been enhanced to include:

- The ability to create multiple aggregate backup copies
- The ability to include the integrated catalog facility user catalog and alias definitions in the aggregate backup
- The automatic calculation of required storage for recovery
- Expanded naming conflict resolution
- A reduction in the DASD storage space required for recovery by allowing recovery to tape
- An option that deletes source user DASD data sets and removes source tape data sets from the catalog after an aggregate has been successfully backed up.

Enhanced Dynamic Cache Management: DFSMS/MVS provides an enhancement to dynamic cache management. The enhancement further reduces the need for manual cache tuning and enables as many data sets as possible to benefit from the cache and non-volatile storage. A new cache management algorithm optimizes the selection of data sets that are candidates for caching, providing improved performance when the 3990 Model 3 controller's cache becomes overloaded.

Data Set Performance Statistics: DFSMS/MVS provides new SMF type 42 subtype records statistics you can use to analyze data set performance. I/O statistics are available for system-managed data sets, providing information such as the total number of I/O operations and the average I/O response time. Additional statistics are also available for system-managed data sets residing on DASD volumes connected with a 3990 Model 3 Storage Control. These statistics include the total number of I/O operations for data sets that are candidates for caching, and the number of I/O operations that actually used cache for those data sets. All of these statistics are also available by storage class.

Sharing Partitioned Data Sets Extended (PDSEs) Among Systems:

DFSMS/MVS allows users on different MVS/ESA systems to access the same partitioned data set extended (PDSE) simultaneously. Users on multiple MVS systems can concurrently create, read, or replace members of the PDSE.

Improved Performance for Copying PDSEs: DFSMS/MVS improves the performance of copying PDSEs; making it comparable to the performance of copying partitioned data sets with IEBCOPY and DFSMSdss.

Enhanced Device Console Services: Device console services are also improved, allowing you to issue the DISPLAY SMS,OAM and DISPLAY SMS,OSMC operator commands from a TSO terminal in CONSOLE mode. Also, messages for Storage Management Subsystem operator commands that are issued by the system console operator are routed back only to the console on which the command was issued.

Improved Support for Empty Data Sets: Users will not receive I/O errors or read residual data if users attempt to read certain data sets before the data has been written. This enhancement applies to all system-managed sequential data sets, striped and non-striped, and all system-managed data sets with undefined data set organizations.

Device Support

DFSMS/MVS introduces the following new device support:

IBM 3495 Tape Library Dataserver: DFSMS/MVS supports the IBM 3495 Tape Library Dataserver, which you can use to automate your tape libraries.

IBM S/370 and S/390 Optical Media Attach/2: DFSMS/MVS supports the IBM S/370 and S/390 Optical Media Attach/2, which enables MVS users to use OS/2 to read in softcopy books or other files distributed on storage media supported by OS/2 including hard disks, CD-ROMs, and floppy disks. The IBM S/370 and S/390 Optical Media Attach/2 emulates an IBM 3422 Magnetic Tape Subsystem, and is defined to the system with the device type 3423.

DFSMS/MVS supports all models of the 3390, 9345, and 3380, including devices that emulate the 3390, 9345, or 3380. However, DFSMS/MVS Version 1 Release 1 is the last release to support 3380 standard models.

First Edition, July 1992

This edition is based on and replaces *MVS/ESA Storage Management Library: Managing Storage Pools*, SC26-4656.

Preface

This publication is intended to help system programmers, storage administrators, and other data processing professionals implement system-managed storage under the MVS/ESA operating system. It provides guidance on how to establish and enforce storage management policies; manage data sets, objects, and database data; and protect the data in the system-managed environment.

This book is part of the MVS/ESA Storage Management Library (SML), a set of books that outline efficient storage management strategies and techniques for MVS/ESA systems. Although most of the tools and techniques illustrated in this book apply only to an MVS/ESA environment, you can apply the concepts of efficient storage management to the MVS/XA and MVS/370 operating systems.

Storage management consists of a variety of fairly complex tasks. There are both hardware and software products that can help you perform these tasks, providing optimal performance, availability, and use of space. For a complete list of the software products that allow you to maximize the benefits of system-managed storage, see *DFSMS/MVS Version 1 Release 2 General Information*. For a list of related publications, see *MVS/ESA SML: Storage Management Reader's Guide* (available in softcopy format only on *IBM Online Library Omnibus Edition MVS Collection*).

Required Publications

To understand this manual, you need to be familiar with the following publications:

MVS/ESA SML: Implementing System-Managed Storage

MVS/ESA SML: Managing Storage Groups

DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference

DFSMS/MVS V1R2 Using Data Sets

The Storage Management Library

Figure 1 shows the Storage Management Library. This library does not replace the publications that document the various IBM hardware and software products used for storage management. Instead, it provides a framework to help storage administrators identify when information in another library is relevant to a particular task.

The Storage Management Library is also available in BookManager-readable form on the *IBM Online Library Omnibus Edition MVS Collection*, SK2T-0710.

Figure 1 (Page 1 of 2). The Storage Management Library

Publication	Description
<i>MVS/ESA Storage Management Library: Leading a Storage Administration Group</i> , SC26-3126	Provides advice for the data processing manager on setting up a storage administration group, providing service to user groups, and planning for storage management in the future.

Figure 1 (Page 2 of 2). The Storage Management Library

Publication	Description
<i>MVS/ESA Storage Management Library: Implementing System-Managed Storage</i> , SC26-3123	Provides system programmers, storage administrators, and other data processing professionals with information on how to plan and implement the conversion of an existing MVS installation to system-managed storage. Former title: <i>Storage Management Subsystem Migration Planning Guide</i> .
<i>MVS/ESA Storage Management Library: Managing Data</i> , SC26-3124	Provides guidance to system programmers, storage administrators, and other data processing professionals on how to establish and enforce storage management policies; manage data sets, objects, and database data; and protect the data in a system-managed environment. Former title: <i>Managing Data Sets and Objects</i> .
<i>MVS/ESA Storage Management Library: Managing Storage Groups</i> , SC26-3125	Provides system programmers, storage administrators, and data processing professionals information on how to plan, implement, and maintain storage groups for DASD, tape, and optical volumes. Former title: <i>Managing Storage Pools</i> .
<i>IBM Online Library Omnibus Edition MVS Collection</i> , SK2T-0710	The Storage Management Library in BookManager-readable format is included on this CD-ROM.
<i>MVS/ESA Storage Management Library</i> , SBOF-3012	The complete Storage Management Library in hardcopy format.

Related Publications

Figure 2 lists publications that provide additional detail on the subjects discussed in this book. For information on how to order these manuals, contact your local IBM branch office.

Figure 2. Related Publications

Publication	Order Number
<i>Data Facility Hierarchical Storage Manager Disaster Backup Guide</i>	SH35-0109
<i>Interactive System Productivity Facility (ISPF) and ISPF/Program Development Facility Version 3 General Information</i>	GC34-4133
<i>MVS/ESA General Information</i>	GC28-1422
<i>MVS/ESA Resource Measurement Facility Version 4 General Information</i>	GC28-1028

Referenced Publications

Figure 3 lists the publications referenced in this book:

Figure 3 (Page 1 of 3). Referenced Publications

Short Title	Publication Title	Order Number
Classified Data Erasure Program: User's Guide and Reference	<i>Classified Data Erasure Program Version 2 Release 1: User's Guide and Reference</i>	SC26-4781
DB2 Administration Guide	<i>IBM DATABASE 2 Version 3 Administration Guide</i>	SC26-4888

Figure 3 (Page 2 of 3). Referenced Publications

Short Title	Publication Title	Order Number
DFSMS/MVS V1R2 DFM/MVS Guide and Reference	<i>DFSMS/MVS Version 1 Release 2 Distributed FileManager/MVS Guide and Reference</i>	SC26-4915
DFSMS/MVS V1R2 DFSMSShsm Implementation and Customization Guide	<i>DFSMS/MVS Version 1 Release 2 DFSMSShsm Implementation and Customization Guide</i>	SH21-1078
DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide	<i>DFSMS/MVS Version 1 Release 2 DFSMSrmm Implementation and Customization Guide</i>	SC26-4932
DFSMS/MVS V1R2 Installation Exits	<i>DFSMS/MVS Version 1 Release 2 Installation Exits</i>	SC26-4908
DFSMS/MVS V1R2 Managing Catalogs	<i>DFSMS/MVS Version 1 Release 2 Managing Catalogs</i>	SC26-4914
DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment	<i>DFSMS/MVS Version 1 Release 2 Managing Data Availability in a System-Managed Storage Environment</i>	SC26-4928
DFSMS/MVS V1R2 DFSMSShsm Managing Your Own Data	<i>DFSMS/MVS Version 1 Release 2 DFSMSShsm Managing Your Own Data</i>	SH21-1077
DFSMS/MVS V1R2 OAM Application Programmer's Reference	<i>DFSMS/MVS Version 1 Release 2 Object Access Method Application Programmer's Reference</i>	SC26-4917
DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Object Support	<i>DFSMS/MVS Version 1 Release 2 Object Access Method Planning, Installation, and Storage Administration Guide for Object Support</i>	SC26-4918
DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	<i>DFSMS/MVS Version 1 Release 2 Object Access Method Planning, Installation, and Storage Administration Guide for Tape Libraries</i>	SC26-3051
DFSMS/MVS V1R2 Planning for Installation	<i>DFSMS/MVS Version 1 Release 2 Planning for Installation</i>	SC26-4919
DFSMS/MVS V1R2 DFSMSdss Storage Administration Guide	<i>DFSMS/MVS Version 1 Release 2 DFSMSdss Storage Administration Guide</i>	SC26-4930
DFSMS/MVS V1R2 DFSMSShsm Storage Administration Guide	<i>DFSMS/MVS Version 1 Release 2 DFSMSShsm Storage Administration Guide</i>	SH21-1076
DFSMS/MVS V1R2 DFSMSdftp Storage Administration Reference	<i>DFSMS/MVS Version 1 Release 2 DFSMSdftp Storage Administration Reference</i>	SC26-4920
DFSMS/MVS V1R2 DFSMSdss Storage Administration Reference	<i>DFSMS/MVS Version 1 Release 2 DFSMSdss Storage Administration Reference</i>	SC26-4929
DFSMS/MVS V1R2 Using Advanced Services for Data Sets	<i>DFSMS/MVS Version 1 Release 2 Using Advanced Services for Data Sets</i>	SC26-4921
DFSMS/MVS V1R2 Using Data Sets	<i>DFSMS/MVS Version 1 Release 2 Using Data Sets</i>	SC26-4922
DFSMS/MVS V1R2 Using ISMF	<i>DFSMS/MVS Version 1 Release 2 Using the Interactive Storage Management Facility</i>	SC26-4911
DFSMS/MVS V1R2 DFSMSrmm Guide and Reference	<i>DFSMS/MVS Version 1 Release 2 DFSMSrmm Guide and Reference</i>	SC26-4931

Figure 3 (Page 3 of 3). Referenced Publications

Short Title	Publication Title	Order Number
DFSMS/MVS V1R2 Using the Volume Mount Analyzer	<i>DFSMS/MVS Version 1 Release 2 Using the Volume Mount Analyzer</i>	SC26-4925
DFSORT Application Programming Guide	<i>DFSORT Application Programming Guide</i>	SC33-4035
ICFRU Program Description/Operations Manual	<i>Integrated Catalog Forward Recovery Utility: Program Description/Operations Manual</i>	SH20-6952
IMS/ESA Database Administration Guide	<i>IMS/ESA Version 3 Release 1 Database Administration Guide</i>	SC26-4281
IMS/ESA System Administration Guide	<i>IMS/ESA Version 3 Release 1 System Administration Guide</i>	SC26-4282
JCL User's Guide	<i>MVS/ESA JCL User's Guide</i>	GC28-1473
MVS/ESA Global Resource Serialization	<i>MVS/ESA Planning: Global Resource Serialization</i>	GC28-1621
MVS/ESA SPL: SMF	<i>MVS/ESA System Management Facilities (SMF)</i>	GC28-1457
System/390 MVS Sysplex Hardware and Software Migration	<i>System/390 MVS Sysplex Hardware and Software Migration</i>	GC28-1210
RACF Auditor's Guide	<i>Resource Access Control Facility (RACF) Auditor's Guide</i>	SC28-1342
RACF Command Language Reference	<i>Resource Access Control Facility (RACF) Command Language Reference</i>	SC28-0733
RACF Security Administrator's Guide	<i>Resource Access Control Facility (RACF) Security Administrator's Guide</i>	SC28-1340
SLR User's Guide	<i>Service Level Reporter User's Guide</i>	SH19-6442
System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism	<i>System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism</i>	GC28-1208
ADSM Administrator's Guide	<i>ADSTAR Distributed Storage Manager: Administrator's Guide</i>	SH35-0117
User Exits	<i>MVS/ESA System Programming Library: User Exits</i>	GC28-1836
Using 3390 in an MVS Environment	<i>Using 3390 in an MVS Environment</i>	GC26-4574

Chapter 1. Introducing System-Managed Storage

System-managed storage is the IBM* automated approach to managing auxiliary storage. It uses software programs to manage data security, placement, migration, backup, recall, recovery, and deletion to ensure that current data is available when needed, and obsolete data is removed from storage.

System-managed storage is tailored to your needs. You define the requirements for performance, security, and availability, along with storage management policies used to automatically manage the direct access, tape, and optical devices used by its MVS/ESA* operating systems.

The combination of system-managed storage and related hardware and software products is called the *DFSMS* environment*.

The DFSMS Environment for MVS/ESA

To implement the DFSMS environment and to take advantage of all the functions available with MVS/ESA, you need to install a specific set of software products. Figure 4 shows the relationship among the IBM products that form the DFSMS environment.

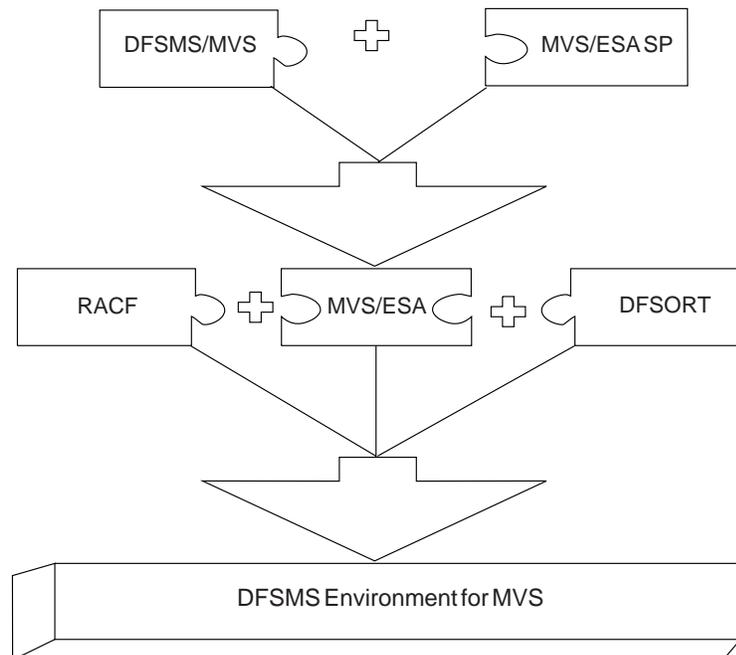


Figure 4. Products in the DFSMS Environment

The **DFSMSdfp*** functional component of DFSMS/MVS* provides the storage, program, data, and device management functions of MVS/ESA. The Storage Management Subsystem (SMS) component of DFSMSdfp is fundamental in providing these functions.

The **DFSMSdss*** functional component of DFSMS/MVS copies and moves data for MVS/ESA.

The **DFSMShsm*** functional component of DFSMS/MVS provides the backup, recovery, migration, recall, and space management functions in the DFSMS environment.

The **DFSMSrmm*** functional component of DFSMS/MVS provides the management functions for removable media, including tape cartridges and reels, and optical volumes.

DFSORT sorts, merges and copies data.

RACF* controls access to data and other resources in MVS/ESA.

Benefits of System-Managed Storage

With the Storage Management Subsystem (SMS), you can define performance goals and data availability requirements, create model data definitions for typical data sets, and automate data backup. SMS can automatically assign, based on installation policy, those services and data definition attributes to data sets when they are created. IBM storage management-related products determine data placement, manage data backup, control space usage, and provide data security.

The goals of system-managed storage are to:

- Improve the use of the storage media; for example, by reducing out-of-space abends and providing a way to set a free-space requirement.
- Reduce the labor involved in storage management by centralizing control, automating tasks, and providing interactive controls for storage administrators.
- Reduce the user's need to be concerned with the physical details of performance, space, and device management. Users can focus on using data instead of managing data.

The benefits of system-managed storage are:

Simplified Data Allocation

System-managed storage enables users to simplify their data allocations. For example, without using the Storage Management Subsystem, an MVS/ESA user would have to specify the unit and volume on which the system should allocate the data set. The user would also have to calculate the amount of space required for the data set in terms of tracks or cylinders. This means the user has to know the track size of the device which will contain the data set.

With system-managed storage, users can let the system select the specific unit and volume for the allocation. They can also specify size requirements in terms of megabytes or kilobytes. This means the user does not need to know anything about the physical characteristics of the devices in the installation.

Improved Allocation Control

System-managed storage enables you to set a requirement for free space across a set of direct access storage device (DASD) volumes. You can then provide adequate free space to avoid out-of-space abends. The system automatically places data on a volume containing adequate free space.

You can also set a threshold for scratch tape volumes in tape libraries, to ensure enough cartridges are available in the tape library for scratch mounts.

Improved Input/Output (I/O) Performance Management

System-managed storage enables you to improve DASD I/O performance across the installation and at the same time reduce the need for manual tuning by defining performance goals for each class of data. You can use cache statistics recorded in system management facilities (SMF) records to help evaluate performance. You can also improve sequential performance by using extended sequential data sets. The DFSMS environment makes the most effective use of the caching abilities of the IBM 3990 Model 3 and Model 6 Storage Controls, as well as other new models.

Automated DASD Space Management

System-managed storage enables you to automatically reclaim space which is allocated to old and unused data sets or objects. You can define policies that determine how long an unused data set or object will be allowed to reside on *primary storage* (storage devices used for your active data). You can have the system remove obsolete data by migrating the data to other DASD, tape, or optical volumes, or you can have the system delete the data. You can also release allocated but unused space which is assigned to new and active data sets.

Automated Tape Space Management

System-managed storage enables you to fully use the capacity of your tape cartridges and to automate tape mounts. Using tape mount management techniques, DFSMSHsm can fill tapes to their capacity. With 3490E tape devices, Enhanced Capacity Cartridge System Tape, 36-track recording mode, and the improved data recording capability, you can increase the amount of data that can be written on a single tape cartridge.

You can also use the IBM 3495 or 3494 Tape Library Dataserver to automatically mount tape volumes and manage the inventory in an automated tape library. If you do not have an automated tape library dataserver, you can still take advantage of system-managed tape by using manual tape libraries and the 3495 Model M10 Tape Library Dataserver.

Automated Optical Space Management

System-managed storage enables you to fully use the capacity of your optical cartridges and to automate optical mounts. Using a 3995 Optical Library Dataserver, you can automatically mount optical volumes and manage the inventory in an automated optical library.

Improved Data Availability Management

System-managed storage enables you to provide different backup requirements to data residing on the same DASD volume. Thus, you do not have to treat all data on a single volume the same way.

You can use DFSMSHsm to automatically back up CICS/ESA* and DATABASE 2* (DB2*) databases, partitioned data sets extended (PDSEs), and physical sequential, partitioned, virtual storage access method (VSAM), hierarchical file system (HFS), and direct access data sets. You can also back up other types of data and use concurrent copy to maintain access to critical data sets while they are being backed up. Concurrent Copy, along with Backup-While-Open, has an added advantage that it avoids the invalidation of a backup of a CICS VSAM KSDS due to a control area or control interval split.

You can also create a logical grouping of data sets, so that the group is backed up at the same time to allow for recovery of the application defined by the group. This is done with the aggregate backup and recovery support (ABARS) provided by DFSMSHsm.

Simplified Conversion of Data to Different Device Types

System-managed storage enables you to move data to new volumes without requiring users to update their job control language (JCL). Because users in a DFSMS environment do not need to specify the unit and volume which contains their data, it does not matter to them if their data resides on a specific volume or device type. This allows you to easily replace old devices with new ones.

You can also use system-determined block sizes to automatically reblock physical sequential and partitioned data sets that can be reblocked.

Managing Data with SMS

In the DFSMS environment, you use SMS classes and groups to set service requirements, performance goals, and data definition models for your installation. You use the Interactive Storage Management Facility (ISMF) to create the appropriate classes and groups, and Automatic Class Selection (ACS) routines to assign them to data according to your installation's policies.

Using SMS Classes and Groups

On systems that do not use DFSMS, storage management consists mostly of manual operations performed on individual data sets, and manual and automated operations performed on volumes. With SMS, you can automate storage management for individual data sets and objects, and for DASD, optical, and tape volumes. You use SMS classes and groups to define the goals and requirements that the system should meet for a data set or object. Figure 5 on page 5 shows the relationship of the classes and groups to your goals and requirements.

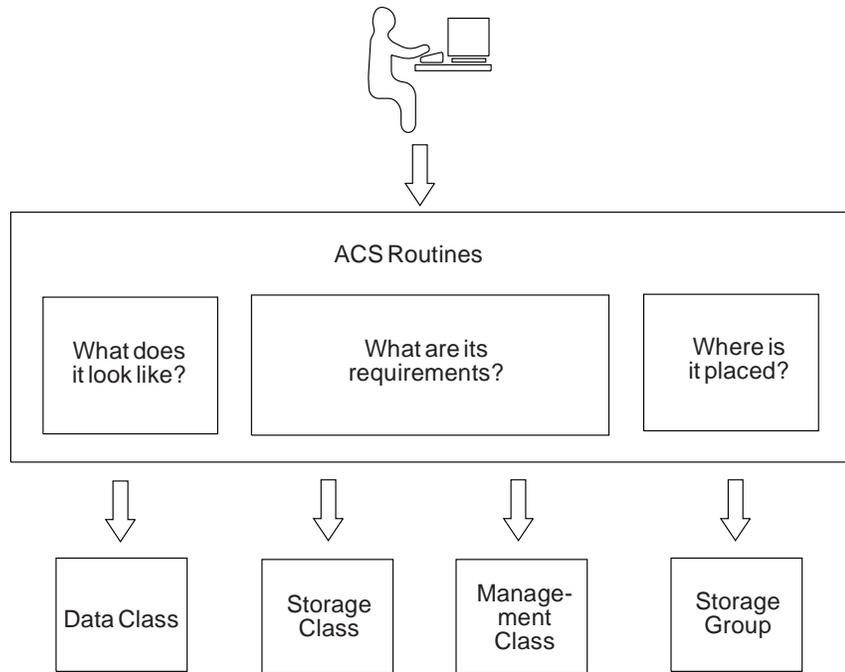


Figure 5. Allocating Data Sets or Storing Objects. You use data class to define model allocation characteristics for data sets; storage class to define performance and availability goals; management class to define backup and retention requirements; and storage group to create logical groupings of volumes to be managed as a unit.

Figure 6 shows how a data set, object, DASD volume, tape volume, or optical volume becomes system-managed.

	DASD	Optical	Tape
Data Set ₁	Assign Storage Class	Not applicable	Not system-managed ₂
Object ₃	Stored	Stored	Stored
Volume	Assign Storage Group	Define OAM Storage Groups	Assign Storage Group ₄

Figure 6. When A Data Set, Object, or Volume Becomes System-Managed

Notes:

1. A DASD data set is *system-managed* if you assign it a storage class. If you do not assign a storage class, the data set is directed to a *non-system-managed* DASD or tape volume — one that is not assigned to a storage group.
2. You can assign a storage class to a tape data set to direct it to a *system-managed* tape volume. However, only the tape volume is considered *system-managed*, not the data set.
3. OAM objects each have a storage class; therefore, objects are *system-managed*. The optical or tape volume on which the object resides is also *system-managed*.
4. Tape volumes are added to tape storage groups in tape libraries when the tape data set is created.

Using Data Classes

A *data class* is a collection of allocation and space attributes that you define. It is used when data sets are created. You can simplify data set allocation for your users by defining data classes that contain standard data set allocation attributes. You can use data classes with both system-managed and non-system-managed data sets, but some data class characteristics are only available with system-managed requests.

Data class attributes define space and data characteristics of data sets that are normally specified on JCL DD statements, TSO/E ALLOCATE commands, access method services (IDCAMS) DEFINE commands, dynamic allocation requests, and ISPF/PDF panels. For tape data sets, data class attributes can also specify the type of cartridge and recording method, and if the data is to be compacted. Users then need only specify the appropriate data classes to create standardized data sets.

You can use data class to specify that a sequential data set should be striped. Extended sequential data sets are allocated across DASD volumes to sustain a data rate you specify.

You can also use the data class automatic class selection (ACS) routine to automatically assign data classes to new data sets. For example, data sets with the low-level qualifiers LIST, LISTING, OUTLIST, or LINKLIST are usually utility output data sets with similar allocation requirements, and can all be assigned the same data class.

As Figure 7 shows, data sets can be assigned a data class during data set creation.

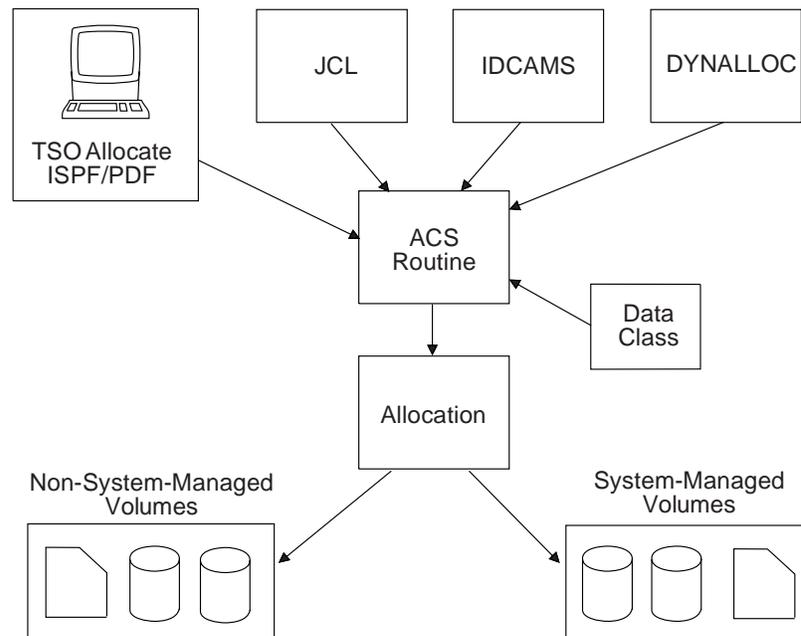


Figure 7. Using Data Class. Data classes can be used for new allocations of both system-managed and non-system-managed data sets.

Even though data class is optional, we usually recommend that you assign data classes to system-managed and non-system-managed data. Although the data class is not used after the initial allocation of a data set, the data class name is kept in the catalog entry for system-managed data sets for future reference. The data class name is not saved for non-system-managed data sets, although the allocation attributes in the data class are used to allocate the data set.

For objects on tape, we recommend that you do not assign a data class via the ACS routines. To assign a data class, specify the name of that data class on the SETOAM command.

If you change a data class definition, the changes only affect new allocations. Existing data sets allocated with the data class are not changed.

Using Storage Classes

A *storage class* is a collection of performance goals and availability requirements that you define. It is used to select a device to meet those goals and requirements. Only system-managed data sets and objects can be assigned a storage class. Storage classes free users from having to know about the physical characteristics of storage devices and manually placing their data on appropriate devices.

Some of the availability requirements you can specify with storage classes can only be met by DASD volumes attached through a 3990 Model 3 or Model 6 Storage Control or a similar device. You can specify attributes that require the use of the 3990 Model 3 or Model 6 Storage Control dual copy, DASD fast write, and concurrent copy features. The performance goals can be met through devices attached through storage controls with or without cache, depending on the goals you set. Figure 8 shows the storage control configurations needed to use all storage class attribute values.

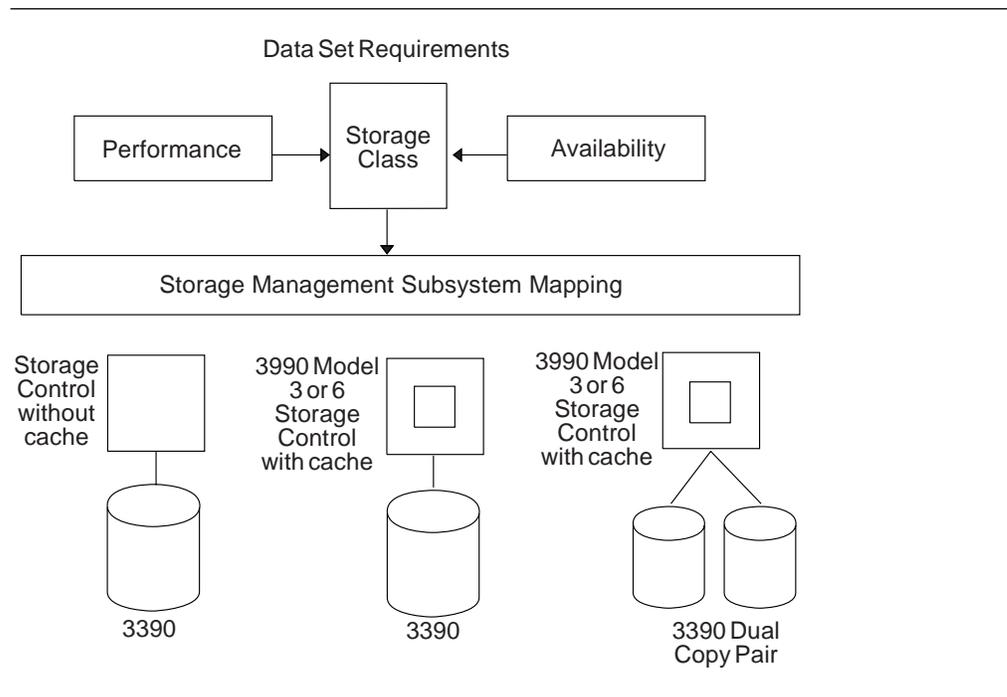


Figure 8. Using Storage Class. Storage classes make the best use of 3990 Model 3 or 6 Storage Controls.

With storage class, you can assign a data set to dual copy volumes to ensure continuous availability for the data set. With dual copy, two current copies of the data set are kept on separate DASD volumes. If the volume containing the primary copy of the data set is damaged, the companion volume is automatically brought online and the data set continues to be available and current.

You can use the ACCESSIBILITY attribute of the storage class to request that concurrent copy be used when data sets or volumes are backed up.

You can specify an I/O response time objective with storage class. During data allocation, the system attempts to select the closest available volume to the specified performance objective.

For objects, the system uses the performance goals you set in the storage class to place the object on DASD, optical, or tape volumes. The storage class is assigned to an object when it is stored or when the object is transitioned. The ACS routines can override this assignment.

If you change a storage class definition, the changes affect the performance service levels of existing data sets that are assigned that class when the data sets are subsequently opened. However, the definition changes do not affect the location or allocation characteristics of existing data sets.

Using Management Classes

A *management class* is a collection of management attributes that you define. It is used to control the retention, migration, backup and release of allocated but unused space for data sets, or to control the retention, backup, and class transition of objects. Management classes let you define management requirements for individual data sets, rather than defining the requirements for entire volumes.

If you do not explicitly assign a management class to a system-managed data set or object, the system uses the default management class. You can define your own default management class when you define your SMS base configuration.

For objects, you can:

- Assign a management class when it is stored, or
- Assign a new management class when the object is transitioned, or
- Change the management class by using the OAM Application Programming Interface (OSREQ CHANGE function).

The ACS routines can override this assignment for objects.

Figure 9 on page 9 shows that you can use management class attributes to do the following:

- Allow early migration for old generations of a generation data group (GDG) by specifying the maximum number of generations to be kept on primary storage, and what to do with rolled-off generation data sets
- Delete selected old and unused data sets from DASD volumes
- Release allocated but unused space from data sets
- Migrate unused data sets to tape or DASD volumes
- Specify how often to back up data sets, and whether concurrent copy should be used during backup

- Specify how many backup versions to keep for data sets
- Specify how long to save backup versions
- Specify the number of versions of aggregate backups to keep and how long to retain those versions
- Establish the expiration date for objects
- Establish transition criteria for objects
- Indicate if automatic backup is needed for objects

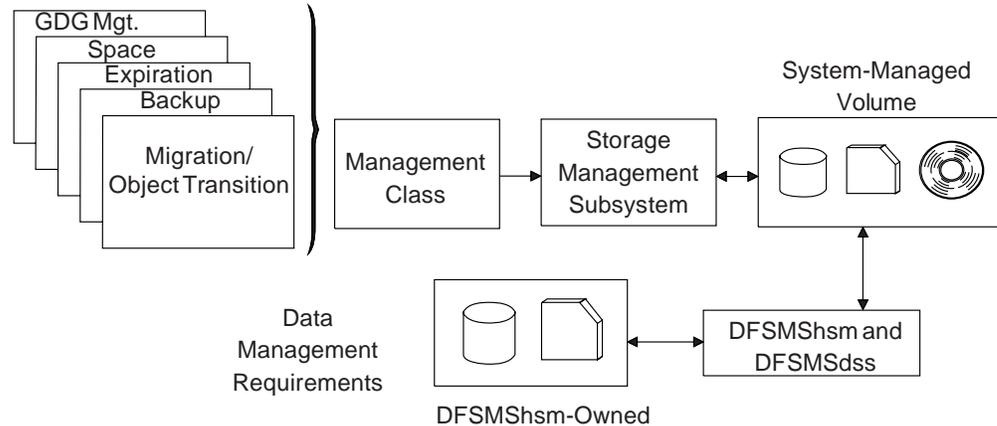


Figure 9. Using Management Class

By classifying your data according to management requirements, you can define unique management classes to fully automate your data set and object management. For example, you can use management classes to control the migration of CICS* user databases, DB2 user databases and archive logs, test systems and their associated data sets, and IMS archive logs. You can specify that DB2 image copies, and IMS image copies and change accumulation logs, be written to primary volumes and then migrated directly to migration level 2 tape volumes.

For objects, you use the class transition attributes to define when an object is eligible for a change in its performance objectives or management characteristics. For example, after a certain number of days you might want to move an object from a high-performance DASD volume to a slower optical volume. You can also use the management class to specify that the object should have a backup copy made when the OAM Storage Management Component (OSMC) is executing.

If you change a management class definition, the changes affect the management requirements of existing data sets and objects that are assigned that class.

Using Storage Groups

A *storage group* is a collection of storage volumes and attributes that you define.

The collection can be a group of:

- System paging volumes
- DASD volumes
- Tape volumes
- Optical volumes
- Combination of DASD and optical volumes that look alike

- DASD, tape and optical volumes treated as a single object storage hierarchy.

Storage groups, along with storage classes, help reduce the requirement for users to understand the physical characteristics of the storage devices which contain their data.

You can direct new data sets to as many as 15 storage groups, although only one storage group is selected for the allocation. The system uses the storage class attributes, volume and storage group SMS status, MVS volume status, and available free space to determine the volume selected for the allocation. In a tape environment, you can also use tape storage groups to direct a new tape data set to an automated or manual tape library.

DFSMSHsm uses some of the storage group attributes to determine if the volumes in the storage group are eligible for automatic space or availability management.

Figure 10 is an example of using storage groups to group storage volumes for specific purposes.

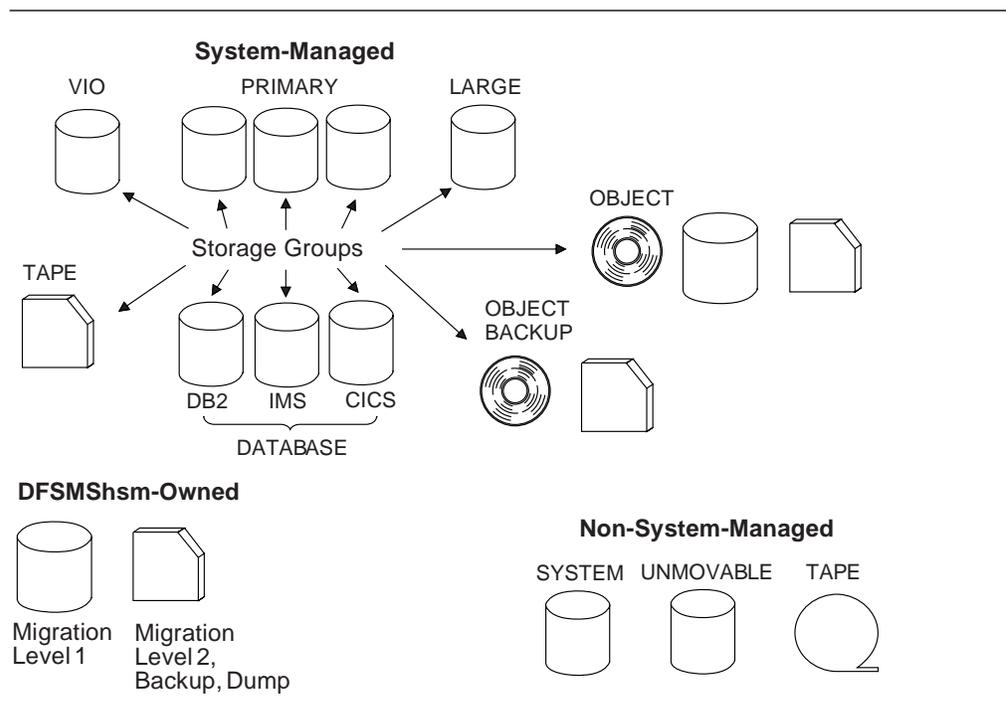


Figure 10. Using Storage Groups. In this example, DASD volumes are grouped so that primary data sets, large data sets, DB2 data, IMS data, and CICS data are all separated. The VIO storage group uses system paging volumes for small temporary data sets. The tape storage groups are used to group tape volumes that are held in tape libraries. The object storage group can span optical, DASD and tape volumes; the object backup storage group can contain either optical or tape volumes within one OAM invocation. Some volumes are not system-managed, and other volumes are owned by DFSMSHsm for use in data backup and migration. DFSMSHsm migration level 2 tape cartridges can be system-managed if you assign them to a tape storage group.

Unlike data, storage, and management classes, users cannot specify a storage group when they allocate a data set, although they can specify a unit and volume. Whether or not you honor their unit and volume request is your decision, but we recommend you discourage users from directly requesting specific devices. It is more effective for your users to specify the logical storage requirements of their

data by storage and management class, which you can then verify in the automatic class selection routines.

For objects, there are two types of storage groups, OBJECT and OBJECT BACKUP. An OBJECT storage group is assigned by OAM when the object is stored; the storage group ACS routine can override this assignment. There is only one OBJECT BACKUP storage group and all backup copies of all objects are assigned to this storage group.

Using Aggregate Groups

An *aggregate group* is a collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy. If a disaster occurs, you can use these backups at a remote or local site to recover critical applications.

You can use aggregate groups as a supplement to using management class for applications that are critical to your business. You can associate an aggregate group with a management class. The management class specifies backup attributes for the aggregate group, such as the copy technique for backing up DASD data sets on primary volumes, the number of aggregate versions to retain, and how long to retain versions. Aggregate groups simplify the control of backup and recovery of critical data sets and applications.

Although SMS must be used on the system where the backups are performed, you can recover aggregate groups to systems that are not using SMS, provided that the groups do not contain data which requires that SMS be active, such as PDSEs. You can use aggregate groups to transfer applications to other data processing installations or migrate applications to newly-installed DASD volumes. You can transfer the application's migrated data, along with its active data, without recalling the migrated data.

Using Automatic Class Selection Routines

General-use programming interface

You use automatic class selection (ACS) routines to assign class and storage group definitions to data sets, database data, and objects. You write ACS routines using the ACS language, which is a high-level programming language. Once written, you use the ACS translator to translate the routines to object form so they can be stored in the SMS configuration.

The ACS language contains a number of read-only variables, which you can use to analyze new data allocations. For example, you can use the read-only variable &DSN to make class and group assignments based on data set or object collection name, or &LLQ to make assignments based on the low-level qualifier of the data set or object collection name. You cannot alter the value of read-only variables.

You use the four read-write variables to assign the class or storage group you determine for the data set or object, based on the routine you are writing. For example, you use the &STORCLAS variable to assign a storage class to a data set or object.

For a detailed description of the ACS language and its variables, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

For each SMS configuration, you can write as many as four routines: one each for data class, storage class, management class, and storage group. Use ISMF to create, translate, validate and test the routines.

Figure 11 shows the order in which ACS routines are processed. Data can become system-managed if the storage class routine assigns a storage class to the data, or if it allows a user-specified storage class to be assigned to the data. If this routine does not assign a storage class to the data, the data cannot reside on a system-managed volume.

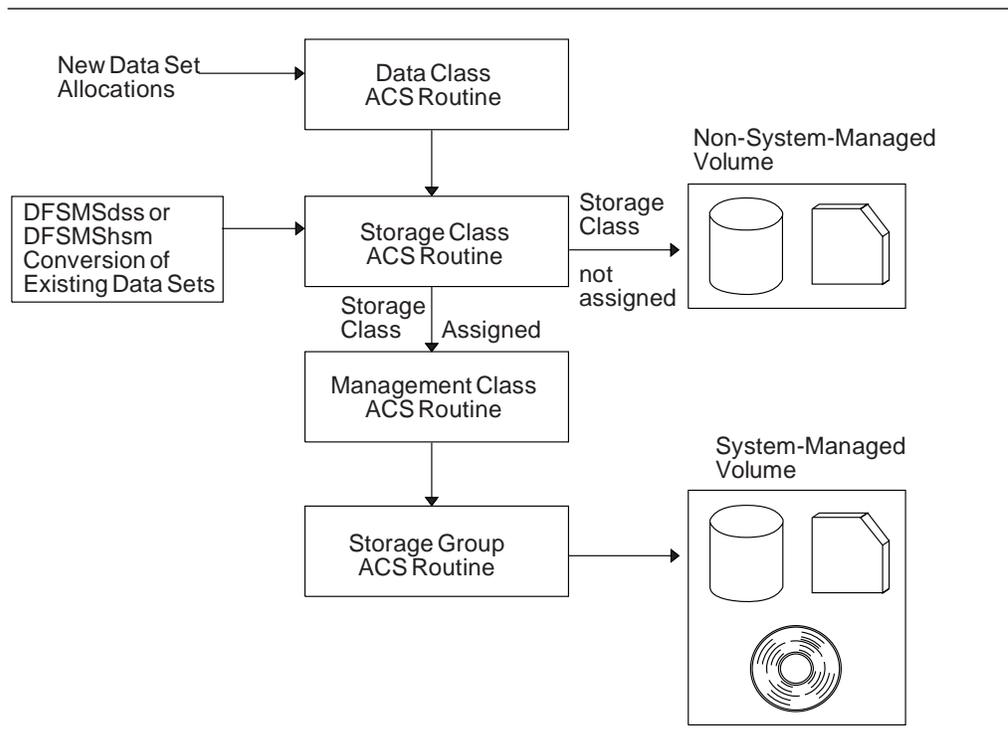


Figure 11. Processing ACS Routines

Because data allocations, whether dynamic or through JCL, are processed through ACS routines, you can enforce installation standards for data allocation on system-managed and non-system-managed volumes. ACS routines also enable you to override user specifications for data, storage, and management class, and requests for specific storage volumes.

You can use the ACS routines to determine the SMS classes for data sets created by the Distributed FileManager/MVS. If a remote user does not specify a storage class, and if the ACS routines decide that the data set should not be system-managed, the Distributed FileManager/MVS terminates the creation process immediately and returns an error reply message to the source. Therefore, when you construct your ACS routines, consider the potential data set creation requests of remote users.

_____ End of General-use programming interface _____

Defining the Storage Management Subsystem Configuration

An SMS configuration is composed of a set of data class, management class, storage class, storage group, optical library and drive definitions, tape library definitions, and ACS routines to assign the classes and groups. It also includes the aggregate group definitions and the SMS base configuration. The SMS base configuration contains default information such as default management class and default device geometry. It also identifies the systems in the installation for which the subsystem manages storage.

This information is stored in SMS *control data sets*, which are VSAM linear data sets. You can define these control data sets using the access method services DEFINE CLUSTER command.

You must define the control data sets before activating SMS. Although you need only allocate the data sets from one system, the active control data set (ACDS) and communications data set (COMMDS) must reside on a device that can be accessed by every system to be managed with the SMS configuration.

SMS uses the following types of control data sets:

Source Control Data Set (SCDS)

This contains the information that defines a single storage management policy, called an SMS configuration. You can define more than one SCDS, but only one can be used to activate a configuration at any given time.

Active Control Data Set (ACDS)

This contains an SCDS that has been activated to control the storage management policy for the installation. When you activate a configuration, SMS copies the existing configuration from the specified SCDS into the ACDS. By using copies of the SMS classes, groups, volumes, optical libraries, optical drives, tape libraries, and ACS routines rather than the originals, you can change the current storage management policy without disrupting it. For example, while SMS uses the ACDS, you can:

- Create a copy of the ACDS
- Create a backup copy of an SCDS
- Modify an SCDS
- Define a new SCDS

The ACDS must reside on a shared device to ensure that all systems in the installation use the same active configuration.

Communications Data Set (COMMDS)

Contains the name of the ACDS and enables communication between SMS systems in a multisystem environment. The COMMDS also contains space statistics, SMS status, and MVS status for each system-managed volume.

Although only one COMMDS is used at a time for an SMS installation, we recommend that you have more COMMDSs on different volumes for recovery purposes.

Coexistence Considerations

This section discusses coexistence for environments using any combination of DFSMS/MVS, MVS/DFP Version 3, and MVS/XA* Data Facility Product Version 2.

Sharing System-Managed and Non-System-Managed Data

You can share system-managed data sets between systems that are using SMS and share the same control data sets. You can share any non-system-managed data sets between MVS/ESA and MVS/XA operating systems without regard for SMS or class definitions. However, you must ensure that operating systems using SMS do not share system-managed data with MVS/XA Data Facility Product Version 2 operating systems.

If you transfer data between operating systems using the TSO TRANSMIT and RECEIVE commands, or the NetView* File Transfer Program Version 2 for MVS, SMS class names are not transmitted with the data set because SMS control information is stored externally, not in the data set.

If SMS receives a data set from either a system-managed or non-system-managed environment, and ACS processing determines that the data set is to be system-managed, SMS assigns classes as for any new data set.

If the data set was originally system-managed and is transmitted to an MVS/XA Data Facility Product Version 2 operating system, the target system sees no indication of system management, and allocation proceeds normally.

If you are sending application JCL to other sites that use the facilities provided by DFSMS/MVS or MVS/DFP Version 3 (such as JCL parameters used with SMS), you need to synchronize the use of SMS at the sites.

Sharing Catalogs in Multisystem Environments

To successfully share a catalog between an MVS/XA Data Facility Product Version 2 operating system and an MVS/ESA operating system, you must ensure that all of the following are true:

- The catalog does not reside on a system-managed volume.
- No data sets cataloged in the catalog reside on a system-managed volume.
- The appropriate toleration PTFs have been applied to the Version 2 system.
- The volume is initialized as shared.
- The catalog is defined with SHAREOPTIONS(3 4).

In a DFSMS environment, all permanent DASD data sets must be cataloged. Only integrated catalog facility catalogs can be used with SMS. Although a catalog contains entries for system-managed data, the catalog itself does not have to be system-managed. However, we recommend that catalogs containing entries for system-managed data sets also be system-managed. A catalog can contain entries for data sets that are system-managed, and entries for data sets that are not system-managed.

A *shared catalog* is a basic catalog structure that is used by more than one system. It must be defined with SHAREOPTIONS(3 4), and reside on a DASD volume that has been initialized as shared. All VSAM volume data sets (VVDSs) are defined as shared.

By default, catalogs are defined as shared. You can specify that a catalog is not to be shared by specifying SHAREOPTIONS (3 3). Only define a catalog as unshared if you are certain it will not be shared. Place unshared catalogs on volumes that have been initialized as unshared. Catalogs that are defined as unshared that reside on shared volumes may become damaged.

If you are using SMS in a multisystem environment, ensure that all systems sharing system-managed data are using DFSMS/MVS or MVS/DFP Version 3. If a system using MVS/XA Data Facility Product Version 2 accesses a catalog that contains entries for system-managed data sets, the catalog may be permanently damaged because the MVS/XA Data Facility Product Version 2 system cannot recognize SMS information.

To share catalogs between systems when some of the systems are running DFSMS/MVS or MVS/DFP Version 3 **without SMS active**, and the others are running MVS/XA Data Facility Product Version 2, you must apply toleration PTFs to the Version 2 systems. Otherwise, an MVS/ESA catalog which is accessed by an MVS/XA Data Facility Product Version 2 system may be permanently damaged, requiring a catalog recovery. These toleration PTFs **do not** allow the MVS/XA Data Facility Product Version 2 system to recognize SMS information, however.

For information on catalog sharing, see *DFSMS/MVS V1R2 Managing Catalogs*.

Figure 12 shows how you can combine MVS/XA and MVS/ESA operating systems.

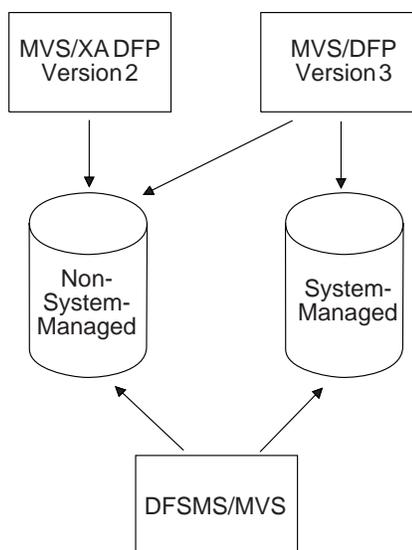


Figure 12. Configuration Combining MVS/XA and MVS/ESA Systems

If the configuration you use during SMS testing is a combination of an MVS/XA Data Facility Product Version 2 system and DFSMS/MVS or MVS/DFP Version 3, you must isolate the system-managed volumes from the MVS/XA Data Facility Product Version 2 system, preventing the MVS/XA Data Facility Product system from:

- Deleting or renaming system-managed data sets
- Creating new data sets on system-managed volumes

- Using the access method services ALTER REMOVEVOLUMES command against the master catalog to perform a VSAM volume cleanup of a system-managed volume
- Sharing catalogs with the DFSMS/MVS or MVS/DFP Version 3 system

If you have a multiaccess spool job entry subsystem (JES) installation, you might need to direct batch jobs to a particular DFSMS/MVS, MVS/DFP Version 3, or MVS/XA Data Facility Product Version 2 system to preserve the integrity of system-managed data sets.

VTOC Compatibility Between MVS/XA and MVS/ESA Systems

Product-sensitive programming interface

The structure of the volume table of contents (VTOC) is the same in MVS/XA and MVS/ESA. However, MVS/DFP Version 3 and DFSMS/MVS use a previously unused field in the format 1 data set control block (DSCB) at offset 78 (X'4E'). Before you convert from MVS/XA Data Facility Product Version 2 or prior versions or releases to MVS/DFP Version 3 or DFSMS/MVS, you should ensure that this field contains binary zeros and that there is no software on your system other than the MVS/ESA operating system that changes this field. You can get unpredictable errors if this field is used to maintain information by programs other than the operating system.

For information about inspecting the contents of DSCBs and fixing any problems that you find, see *DFSMS/MVS V1R2 Planning for Installation* and your IBM service representative.

End of Product-sensitive programming interface

Sharing DASD Between MVS/ESA and MVS/XA Systems

We recommend that you do not share DASD in an environment that has MVS/XA Data Facility Product Version 2 combined with DFSMS/MVS or MVS/DFP Version 3. If you must share DASD in such an environment, share only the non-system-managed volumes, and be sure these volumes do not contain shared catalogs.

If you attempt to share system-managed volumes between MVS/XA Data Facility Product Version 2 and DFSMS/MVS or MVS/DFP Version 3, you will have the following compatibility problems:

- The MVS/XA Data Facility Product Version 2 operating system does not track DASD space usage for SMS.
- The MVS/XA Data Facility Product Version 2 operating system does not recognize the non-VSAM volume records (NVRs) in the VSAM volume data set (VVDS). Thus:
 - Scratching or renaming a non-VSAM data set leaves NVRs in the VVDS with no associated data set.
 - VSAM volume cleanup with the ALTER REMOVEVOLUMES command removes the NVRs for non-VSAM data sets without removing the VTOC entries for those data sets.

- The MVS/XA Data Facility Product Version 2 operating system can allocate non-system-managed data sets on a system-managed volume.

Sharing DASD Between MVS/ESA Systems

You can combine up to eight MVS/ESA operating system images or system groups that are using SMS in a single installation, and share both system-managed and non-system-managed DASD volumes. The system group name represents all systems in the sysplex of the same name that are not explicitly specified in the SMS base configuration.

In an SMS complex, the operating systems communicate by sharing a common configuration stored in the ACDS and common system-managed volume statistics stored in the COMMDS. The volumes that contain these SMS control data sets must be accessible from all systems that are part of the SMS complex. The SCDS must be accessible from all systems that need to perform an ACTIVATE of the configuration. If you have more than 16 systems in an SMS complex, you need to define the ACDS and COMMDS on a volume attached through a 3990 Model 6 storage control (the 3990 Model 3 does not have enough paths to make it possible to share attached volumes with more than 16 systems).

Prior to DFSMS/MVS 1.2, logical connectivity for all system-managed volumes and storage groups was controlled at the individual system level. Allocations, deletions, and accesses could only be performed on systems that had the logical (SMS and MVS) and physical (hardware) connectivity. This also applied to DFSMSHsm operations. Job failures would occur otherwise. In addition, the required catalogs needed to be accessible.

With DFSMS/MVS 1.2, when you define a volume or storage group to have connectivity to a system group, the volume or storage groups must be accessible to all systems that are part of the system group. Otherwise, job failures will occur.

When a common set of classes, groups, ACS routines, and a base configuration are applied across an MVS/ESA multisystem environment, the environment is a simple one. However, if SMS is not active on one of the systems, that system is not able to:

- Create data sets on system-managed volumes
- Delete system-managed data sets
- Extend system-managed data sets to new volumes
- Use JCL keywords supported by SMS

The COMMDS does not record DASD space usage changes for the system that has not activated SMS.

Using System Group Name with Pre-DFSMS/MVS 1.2.0 Systems

The system group name support may coexist with previous MVS/DFP and DFSMS/MVS releases which support SMS in the following situations:

- System group names are not exploited. If only system names are defined in the configuration, DFSMS/MVS 1.2.0 is SMS-compatible with previous releases of MVS/DFP or DFSMS/MVS.
- All pre-DFSMS/MVS 1.2.0 systems are explicitly defined in the configuration and only DFSMS/MVS 1.2.0 systems are represented by system group names. The pre-DFSMS/MVS 1.2.0 systems will interpret their name as a system name and will treat the system group names(s) as a system name. Only the DFSMS/MVS 1.2.0 systems (which contain the system group name support) will exploit the system group names. Refer to Figure 13 for an example of this coexistence environment.

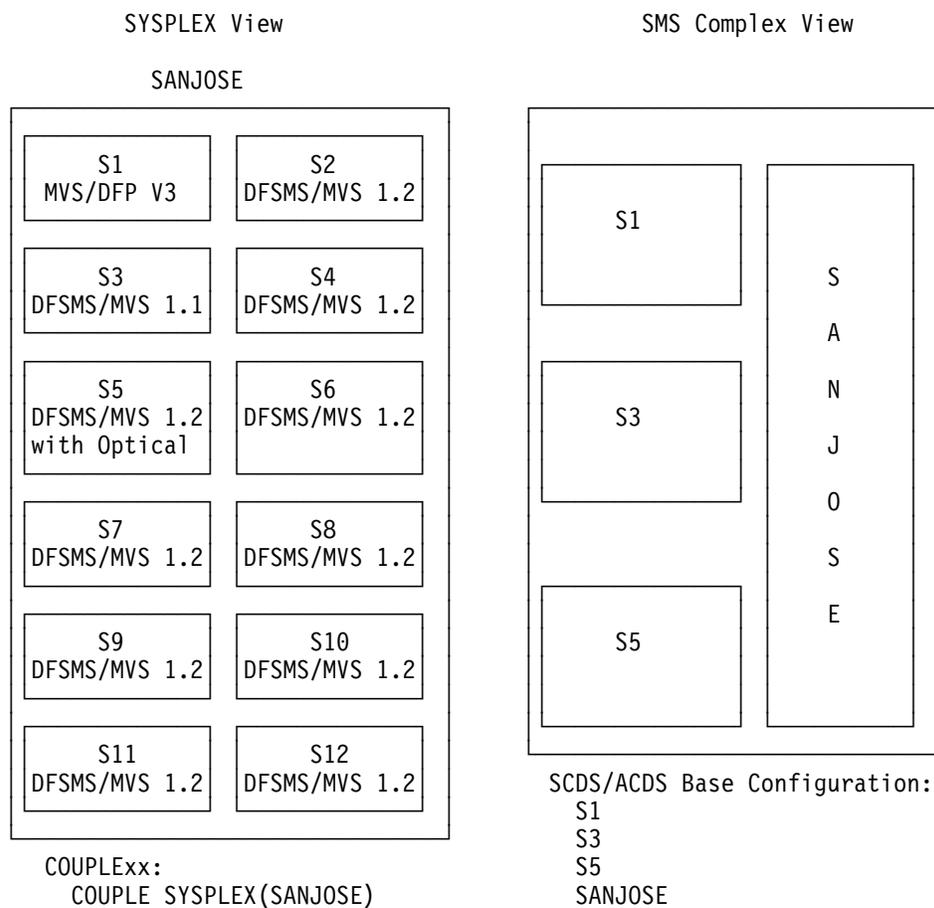


Figure 13. Coexistence when System Group Name Is Exploited

Note: The DFSMS/MVS 1.2.0 systems (which contain the system group name support) can coexist within the same SMS complex with either down-level systems or with systems using optical libraries for image processing applications.

Restrictions:

- You cannot use system group names in a JES3 environment.

- When defining an optical library in an SMS configuration, you can only connect it to one individual system and not to a system group.
- When defining a tape library, you can connect it to individual systems or system groups. However, to vary a tape library online or offline to a particular system, you must connect the tape library to individual system entries.

Using Different Releases of DFDSS, DFHSM, and DFSMS/MVS Programs

When converting to a new release of DFDSS or DFHSM, or to DFSMS/MVS, keep the following in mind:

- Although the DFSMSdss functional component of DFSMS/MVS can restore data dumped by DFDSS, you might not be able to use DFDSS to restore data dumped by DFSMSdss, particularly PDSEs dumped concurrently, extended sequential data sets, and VSAM data sets dumped using VALIDATE (default keyword). However, DFDSS can restore data dumped by DFSMSdss using VALIDATE if you have the VALIDATE PTF (support) on your DFDSS system.
- If you are going to have more than one release level of DFHSM running on shared systems, or you are going to have DFHSM and DFSMS/MVS with the DFSMSHsm functional component running on shared systems, apply the appropriate toleration PTFs to all versions of DFHSM to ensure data integrity and compatibility between the release levels and DFSMS/MVS.
- You cannot use the DFSMSHsm ARECOVER command with the INSTRUCTION or ACTIVITY parameters to recover the instruction data set or activity log of aggregate backups made with DFHSM. Otherwise, DFHSM aggregate backups can be recovered by DFSMSHsm. However, SMS attributes and storage requirements are not written to the ARECOVER activity log.

For more information on these compatibility considerations, see *DFSMS/MVS V1R2 Planning for Installation* and *DFSMS/MVS V1R2 DFSMSHsm Implementation and Customization Guide*.

Hardware Considerations

We recommend that you assign ranges of device numbers to specific device types. In a multi-system environment, you might have some devices defined on more than one system. This enables you to switch devices, such as user terminals, from the primary system to a backup system and the reverse.

You can have over 4096 unit control blocks (UCBs). If you have very large processors and large numbers of processors in your sysplex, you might need addressability to many devices.

The 3390 Model 9 offers more capacity at a lower cost than other 3390 models. While performance without cache might be acceptable for much of the data that is migrated to the 3390 Model 9, configuring with a cache-capable 3990 storage control can improve throughput and performance and also provide access to availability functions, such as concurrent copy and fast dual copy.

You can improve the throughput of your DASD subsystems by using the 3990 Model 6 Storage Control, with its increased cache sizes of up to one gigabyte. For some SMS complex configurations, you might need DASD with enhanced connectivity (an IBM 9340 or DASD attached to a 3990 Model 6 Storage Control).

The 3995 Model 153 or 113 Optical Library Dataserver enables you to keep certain data online much longer than was previously affordable. You might also save money by putting OAM object backup copies on tape instead of optical.

To take advantage of hardware compression for SAM and extended format VSAM data sets, the system uses host-based compression, such as on the IBM 9021 711-based models and 9121 511-based models, if available; otherwise, the system uses the slower Software Compression Facility.

The Responsibilities of a Storage Administrator

The primary responsibility of a storage administrator is to provide improved user services. To provide the best service to your users, you must have a good working relationship with your user groups, and understand the strategy used to address storage management problems.

Working with Your User Groups

Working with your user groups is an ongoing task that is essential for successfully carrying out your storage administration tasks. If you are to assume the responsibility of centrally managing your users' data, you must work closely with them to understand their data processing needs, negotiate service-level agreements with them, and provide the education necessary for them to cooperate with your efforts.

See *MVS/ESA SML: Leading a Storage Administration Group* for recommendations on how to work effectively with your user groups. It includes sample service-level agreements and a user's guide that you can tailor.

Implementing the System-Managed Storage Strategy

The tasks in this publication support the overall strategy to implement system-managed storage and address storage management problems. These tasks are:

- Establishing installation standards
- Managing data allocation
- Managing space use
- Improving I/O Performance
- Managing backup and recovery of data
- Protecting data
- Monitoring space use, DASD and tape capacity, and data performance

Chapter 2. Establishing Installation Standards

Establishing standards, such as naming conventions and allocation policies, helps you manage storage more efficiently and improves service to your users. With them, your installation will be better prepared to make a smooth transition to system-managed storage. This chapter helps you establish installation standards. It describes the following tasks:

- Identifying data types
- Developing naming conventions
- Improving catalog performance
- Simplifying JCL

Negotiate with your user group representatives to agree on the specific policies for the installation, how soon you can implement them, and how strongly you will enforce them. Document negotiated policies in a service-level agreement.

You can also provide your users with an online user's guide covering all your policies. For a sample user's guide and information on service-level agreements, see *MVS/ESA SML: Leading a Storage Administration Group*.

Identifying Data Types

You can simplify storage management by limiting the number of data sets and volumes that cannot be system-managed.

This section describes the data that can and cannot be system-managed.

Data That Can Be System Managed

These are some common types of data that can be system managed. For details on how these data types can be system managed using SMS storage groups, see *MVS/ESA SML: Managing Storage Groups*.

Temporary Data Data sets used only for the duration of a job, job step, or terminal session, and then deleted. These data sets can be cataloged or uncataloged, and can range in size from small to very large.

Permanent Data Data sets consisting of:

- **Interactive Data**
TSO user data sets and ISPF/PDF libraries you use during a terminal session. Data sets classified in this category are typically small, and are frequently accessed and updated.

- Batch Data

Data that is classified as either *online-initiated*, *production*, or *test*.

- Online-initiated are background jobs that an online facility, such as TSO, generates.
- Production batch refers to data created by specialized applications (such as payroll), that could be critical to the continued operation of your business or enterprise.

- VSAM Data

Data organized with VSAM, including VSAM data sets that are part of an existing database.

- Large Data

For most installations, large data sets occupy more than 10% of a single DASD volume. Note, however, that what constitutes a large data set is installation-dependent.

- Multivolume Data

Data sets that span more than one volume.

Database Data

Data types usually having varied requirements for performance, availability, space, and security. To accommodate special needs, database products have specialized utilities to manage backup, recovery, and space usage. Examples include DB2, IMS, and CICS data.

System Data

Data used by MVS to keep the operating system running smoothly. In a typical installation, 30% to 50% of these data sets are high performance and are used for cataloging, error recording, and other system functions. Because these critical data sets contain information required to find and access other data, they are read and updated frequently, often by more than one system in an installation.

Performance and availability requirements are unique for system data. The performance of the system depends heavily upon the speed at which system data sets can be accessed. If a system data set such as a master catalog is unavailable, the availability of data across the entire system and across other systems can be affected.

Some system data sets can be system-managed if they are uniquely named. These data sets include user catalogs. Place other system data sets on non-system-managed volumes.

Object Data

Also known as byte-stream data, this data is used in specialized applications such as image processing, scanned correspondence, and seismic measurements. Object data typically has no internal record or field structure and, once written, the data is not changed or updated. However, the data can be referenced many times during its lifetime.

Under DFSMS/MVS, you can extend the system-managed storage environment to include:

- Workstation data stored on MVS
- Tape volumes residing in either an automated tape library dataserer or a manual tape library dataserer

Data Types That Cannot Be System Managed

This section describes the data that cannot be system managed.

Uncataloged Data

All permanent DASD data under the control of SMS must be cataloged in integrated catalog facility catalogs using standard search order. The catalogs contain the information required for locating and managing system-managed data.

If data sets are cataloged, users do not need to know which volumes the data sets reside on when they reference them; they do not need to specify unit type or volume serial number. This is essential in an environment with storage groups, where users do not have private volumes.

Objects, stored in groups called *collections*, must have their collections cataloged in integrated catalog facility catalogs because they, and the objects they contain, are system-managed data. The object access method (OAM) identifies an object by its collection name and the object's own name. An object is described only by an entry in a DB2 object directory. An object collection is described by a collection name catalog entry and a corresponding OAM collection identifier table entry. Therefore, an object is accessed by using the object's collection name and the catalog entry.

When objects are written to tape, they are treated as tape data sets and OAM assigns two tape data set names to the objects. Objects in an object storage group being written to tape are stored as a tape data set named OAM.PRIMARY.DATA. Objects in an object backup storage group being written to tape are stored as a tape data set named OAM.BACKUP.DATA. Each tape containing objects has only one tape data set, and that data set has one of the two previous names. Because the same data set name can be used on multiple object-containing tape volumes, the object tape data sets are not cataloged.

If you don't already have a policy for cataloging all permanent data, it is a good idea to establish one now. For example, you can enforce standards by deleting uncataloged data sets.

Unmovable Data Sets

Unmovable data sets cannot be system-managed. These data sets include:

- Data sets identified by the following data set organization (DSORGs):
 - Partitioned unmovable (POU)
 - Sequential unmovable (PSU)
 - Direct access unmovable (DAU)
 - Indexed-sequential unmovable (ISU)
- Data sets with user-written access methods
- Data sets containing processing control information on the device or volume on which they reside, including:

- Absolute track data that is allocated in absolute DASD tracks or on split cylinders
- Location dependent direct data sets

All unmovable data sets must be identified and converted for use in a system-managed environment. For information on identifying and converting unmovable data sets, see *MVS/ESA SML: Implementing System-Managed Storage*.

Developing Naming Conventions

Although naming conventions are not a prerequisite for conversion, they make more efficient use of DFSMS. You can also reduce the cost of storage management significantly by grouping data that shares common management requirements. Naming conventions are an effective way of grouping data. They also:

- Simplify service-level assignments to data
- Facilitate writing and maintaining ACS routines
- Allow data to be mixed in a system-managed environment while retaining separate management criteria
- Provide a filtering technique useful with many storage management products.
- Simplify the data definition step of aggregate backup and recovery support

Naming conventions are particularly important to data in a distributed environment. For more information, see *ADSM Administrator's Guide*.

Although naming conventions are a good approach to using system-managed storage efficiently and getting the best use from ACS routines, establishing data naming conventions is not a prerequisite for converting to DFSMS.

This section provides some guidelines for establishing good naming conventions for data sets, database data, and objects in a non-distributed environment.

Using Qualifier Levels

This section explains the highest-level and lowest-level qualifiers (HLQ and LLQ, respectively), and other levels. Most naming conventions are based on the HLQ and LLQ of the data name.

Other levels of qualifiers can be used to identify generation data sets and database data. They can also be used to help users to identify their own data.

Note: Do not duplicate the examples in this section—they are used for illustration purposes only. Your installation might be set up differently, and therefore requires careful planning when implementing data naming conventions.

Highest-Level Qualifiers

The HLQ identifies the owner or owning group of the data, and it can indicate data type. Do not embed information that is subject to frequent change in the HLQ, such as department number, application location, output device type, job name, or access method.

Set a standard within the HLQ. For example, you can require HLQs to conform to the format in Figure 14.

Figure 14. Setting the High-Level Qualifier Standard.

1st Character	2nd Character	Remaining Characters
Type of user	Type of data	Project name, code, or userid
A - Accounting Support	P - Production data	Example:
D - Documentation	D - Development data	3000 = Project code
E - Engineering	T - Test data	
F - Field Support	M - Master data	
M - Marketing Support	U - Update data	
P - Programming	W - Work data	
\$ - TSO userid		

Conventions for HLQs make it easy for you to filter on them, as in Figure 15.

Figure 15. Using HLQs for Filtering

Filtering Criteria	Description
PE*	All production data belonging to the engineering group
*1234	All data relevant to the 1234 project
T	All test data

Lowest-Level Qualifiers

The LLQ determines the contents and storage management processing of the data. You can use LLQs to identify data requirements for:

- Migration (data sets only)
- Backup (data sets and objects)
- Archiving (data sets)
- Retention or expiration (data sets and objects)
- Class transitions (objects only)
- Release of unused space (data sets only)

The retention and expiration of objects on tape volumes are determined on two levels. Tape volumes containing objects have a tape data set expiration date and an expiration date of when the last object on the tape is going to expire. For information on deleting expired objects on tape, see “Deleting Data” on page 62.

Mapping storage management requirements to data names is especially useful in a system-managed environment. In an environment without storage groups, data with differing requirements is often segregated onto separate volumes that are monitored and managed manually. LLQ data naming conventions allow data to be mixed together in a system-managed environment and still retain the separate management criteria.

Figure 16 on page 26 shows some examples of how you can use LLQ naming standards to indicate the storage management processing criteria. The first column lists the LLQ of a data name. An asterisk indicates where a partial qualifier can be used. For example, LIST* indicates that only the first four characters of the LLQ must be LIST; valid qualifiers include LIST1, LISTING, LISTOUT. The remaining columns show the storage management processing information for the data listed. For descriptions and effective uses of storage management attributes, see Chapter 4, “Managing Space Use” on page 59 and Chapter 6, “Backing Up Data” on page 93.

Low-Level Qualifier	Exp Days Non-Usage	Max Ret Period	Partial Release	Migrate Days Non-Usage	Cmd/Auto Migrate	No.GDG Primary	Backup Freqcy	Backup Versns	Retain Days Only Backup	Retain Days Extra Backup	Adm/User Backup	Auto Back
ASM.....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
CLIST....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
COB*....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
CNTL....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
DATA....	400	400	YES	15	BOTH	--	2	2	400	60	BOTH	YES
*DATA....	400	400	YES	15	BOTH	--	2	2	400	60	BOTH	YES
FOR*....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
INCL*....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
INPUT....	400	400	YES	15	BOTH	--	2	2	1100	120	BOTH	YES
ISPROF...	400	400	YES	30	BOTH	--	0	2	60	30	BOTH	YES
JCL.....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
LIST*....	2	2	YES	NONE	NONE	--	NONE	NONE	--	--	BOTH	NO
*LIST....	2	2	YES	NONE	NONE	--	NONE	NONE	--	--	BOTH	NO
LOAD*....	400	400	YES	15	BOTH	--	1	2	--	--	BOTH	YES
MACLIB...	400	400	YES	15	BOTH	--	1	2	400	60	BOTH	YES
MISC.....	400	400	YES	15	BOTH	--	2	2	400	60	BOTH	YES
NAMES....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
OBJ*....	180	180	YES	7	BOTH	--	3	1	180	30	BOTH	YES
PLI.....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
PL1.....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
REPORT...	400	400	YES	15	BOTH	--	2	2	400	60	BOTH	YES
SCRIPT...	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
SOURCE...	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
SRC.....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES
SYSOUT...	2	2	YES	NONE	NONE	--	NONE	NONE	--	--	BOTH	NO
TEMP.....	180	180	YES	7	BOTH	--	3	1	180	30	BOTH	YES
TEST.....	400	400	YES	15	BOTH	--	2	2	400	60	BOTH	YES
TEXT.....	NOLIM	NOLIM	YES	15	BOTH	--	0	5	1100	120	BOTH	YES

Figure 16. Storage Management Based on Data Lowest-Level Qualifier

For generation data sets, the LLQs precede the generation numbers. The format for generation data set names can follow the example in Figure 17:

Figure 17. Generation Data Set Name Format.

HLQ	Other Qualifier Levels	LLQ	Generation Number
hlq	B	llq	GnnnnVnn

The LLQ is the primary way to describe the use and content of the object data. Figure 18 shows how the object collection name can be used for filtering by management class and storage class ACS routines.

Low-Level Qualifier	Exp Days Non-Usage	Max Ret Period	Auto Back	Time Since Creation Years	Months	Days	Time Since Last Use Years	Months	Days	Mnthly on Day	Qtr on
NOTES...	05	15	NO	--	--	--	--	--	--	--	--
LETTERS..	NOLIMIT	30	NO	--	--	--	--	--	--	--	--
REPORTS..	NOLIMIT	NOLIMIT	YES	--	06	--	--	--	--	--	--
CHECKS...	NOLIMIT	NOLIMIT	YES	--	--	30	--	--	--	--	--
FOLIOS...	NOLIMIT	NOLIMIT	YES	07	--	--	--	--	--	--	--
AUDITS...	NOLIMIT	NOLIMIT	YES	--	--	--	--	--	--	--	--

Figure 18. Storage Management Based on Object LLQ

Other Qualifier Levels

You can use other levels of qualifiers to identify specific data types, for example, database data. The format for DB2 database names typically follows the example in Figure 19.

Figure 19. Database Data Name Format.

HLQ	Cluster/Data Component Indicator	User Database Name	Table Space or Index Space Name	Hard-Coded Constant	Table Space Extension
VSAM catalog name	DSNDBx	dbname	tblspacename	I0001	Annn

For more information on DB2 naming conventions, see *DB2 Administration Guide*.

Users might want to have qualifier levels for their own use. For example, you can designate one qualifier level for application groups to indicate program levels, such as version or release number, change number, and cyclic level. Another level of qualifier can enable the users to uniquely identify the particular type of data they own, such as part number, location, and branch office number.

Sample Filtering Routine

General-Use Programming Interface

The sample ACS routine in Figure 20 shows how you can filter data names with HLQs and LLQs.

```

PROC STORCLAS

/*****/
/*          C H A N G E   H I S T O R Y          */
/*          =====                          */
/*          */
/*  DATE      RESP  DESCRIPTION OF CHANGE      */
/*  -----   ---  -----                    */
/*  90/12/04  GTB   INITIAL EXAMPLE ROUTINE CREATED.  */
/*          */
/*          */
/* PURPOSE:  THIS ROUTINE ASSIGNS A STORAGE CLASS TO DATA SETS,
/*           BASED ON THEIR DATA SET NAME HIGH LEVEL QUALIFIER
/*           (HLQ), AND LOWEST LEVEL QUALIFIER (LLQ).
/*          */
/*          */
/* INPUT:    THE FOLLOWING ACS VARIABLES ARE REFERENCED:
/*          */
/*           &HLQ          &UNIT
/*           &LLQ
/*          */
/*          */
/* OUTPUT:   THE FOLLOWING STORAGE CLASS NAMES:
/*          */
/*           CRITICAL      MEDIUM
/*           FASTREAD      STANDARD
/*           FASTWRIT      SYNCWRIT
/*          */
/*          */
/* RETURN    ZERO IS THE ONLY RETURN CODE. ALLOCATIONS ARE NOT
/* CODES:     FAILED IN THIS ROUTINE.
/*          */
/*****/

```

Figure 20 (Part 1 of 3). Filtering on HLQs and LLQs

```

/*****/
/* FILTER LIST SPECIFICATIONS FOR VALID DASD UNIT TYPES THAT CAN BE */
/* SYSTEM MANAGED. */
/*****/

FILTLIST VALID_UNITS INCLUDE('3380','3390', 'SYSDA','')

/*****/
/* FILTER LIST SPECIFICATIONS FOR DATA SET NAME LOWEST LEVEL */
/* QUALIFIERS OF "DATANN" (WHERE 'NNN' IS A NUMBER RANGE FROM 000 - */
/* 999), AND "INPUT". */
/*****/

FILTLIST DSILLQ INCLUDE(DATA%%, 'INPUT')

/*****/
/* FILTER LIST SPECIFICATIONS FOR DATA SET NAME LOWEST LEVEL */
/* QUALIFIERS OF "LISTING", "OUTLIST", "LISTOUT", "LISTNN" (WHERE */
/* 'NNN' IS A NUMBER RANGE FROM 000 - 999), AND "REPORT". */
/*****/

FILTLIST DSOLLQ INCLUDE(*LIST*, 'REPORT')

/*****/
/* FILTER LIST SPECIFICATIONS FOR DATA SET NAME HIGH LEVEL QUALIFIERS */
/* FOR PRODUCTION ENGINEERING DATA. */
/*****/

FILTLIST PEHLQ INCLUDE(PE*)

/*****/
/* FILTER LIST SPECIFICATIONS FOR DATA SET NAME HIGH LEVEL QUALIFIERS */
/* FOR TEST DATA. */
/*****/

FILTLIST THLQ INCLUDE(%T*)

/*****/
/* FILTER LIST SPECIFICATIONS FOR DATA SET NAME HIGH LEVEL QUALIFIERS */
/* FOR DEVELOPMENT DATA, EXCLUDE DATA SETS THAT HAVE A 3000 PROJECT */
/* ACCOUNT CODE. */
/*****/

FILTLIST DHLQ INCLUDE(%D*) EXCLUDE(%D3000)

/*****/
/* START OF SELECT-WHEN-OTHERWISE GROUP OF STATEMENTS TO ASSIGN VALID */
/* STORAGE CLASSES */
/*****/

SELECT

/*****/
/* VERIFY THAT UNIT TYPE IS VALID FOR ALLOCATION AS A SYSTEM-MANAGED */
/* DEVICE. IF NOT, ASSIGN A NULL STORAGE CLASS. */
/*****/

        WHEN (&UNIT ^= &VALID_UNITS)
            SET &STORCLAS = ' '

/*****/
/* WHEN THE DATA SET IS PRODUCTION ENGINEERING INPUT DATA, ASSIGN A */
/* STORAGE CLASS NAME OF "FASTREAD". */
/*****/

        WHEN (&HLQ = &PEHLQ && &LLQ = &DSILLQ)
            SET &STORCLAS = 'FASTREAD'

```

Figure 20 (Part 2 of 3). Filtering on HLQs and LLQs

```

/*****
/* WHEN THE DATA SET IS PRODUCTION ENGINEERING OR TEST OUTPUT, ASSIGN */
/* A STORAGE CLASS NAME OF "FASTWRIT". */
/*****

        WHEN ((&HLQ = &PEHLQ | &HLQ = &THLQ) && &LLQ = &DSOLLQ)
            SET &STORCLAS = 'FASTWRIT'

/*****
/* WHEN THE DATA SET IS TEST INPUT DATA, ASSIGN A STORAGE CLASS NAME */
/* OF "MEDIUM". */
/*****

        WHEN (&HLQ = &THLQ && &LLQ = &DSILLQ)
            SET &STORCLAS = 'MEDIUM'

/*****
/* WHEN THE DATA SET IS DEVELOPMENT INPUT DATA, ASSIGN A STORAGE */
/* CLASS NAME OF "CRITICAL" */
/*****

        WHEN (&HLQ = &DHLQ && &LLQ = &DSILLQ)
            SET &STORCLAS = 'CRITICAL'

/*****
/* WHEN THE DATA SET IS DEVELOPMENT OUTPUT, ASSIGN A STORAGE CLASS */
/* NAME OF "SYNCWRIT". */
/*****

        WHEN (&HLQ = &DHLQ && &LLQ = &DSOLLQ)
            SET &STORCLAS = 'SYNCWRIT'

/*****
/* ASSIGN "STANDARD" STORAGE CLASS NAME TO ALL REMAINING DATA. */
/*****

        OTHERWISE SET &STORCLAS = 'STANDARD'

        END /* END OF SELECT-WHEN-OTHERWISE GROUP OF STATEMENTS */

END /* END OF STORAGE CLASS ROUTINE PROC */

```

Figure 20 (Part 3 of 3). Filtering on HLQs and LLQs

End of General-Use Programming Interface

Minimizing the Effect on Existing Naming Conventions

When establishing data naming conventions, minimize any requirements to rename existing data by adopting the following guidelines:

- If naming conventions are already in place at your installation, use them as a base when building new standards.
- If your installation does not have a data set naming policy and a corresponding enforcement method (using RACF, ACS routines, or both), you need to establish them.

To establish data set naming conventions with the least disruption to your installation and users' environment, do:

- Identify the data set owners.
- Analyze the existing data set names. Work with the data set owners to categorize data sets by use and type (for example, categorize user reports, interface files, and program libraries).
- Find the most consistently used qualifier and use it as your installation standard for each specific data set type. You can also write ACS routines to filter on this qualifier.
- Adapt your ACS routines to filter on both the old naming convention for existing data sets and new data sets that meet the data set naming standard.

Developing Naming Conventions

This is an example of developing installation naming conventions with the least effect on your environment:

- Assume that, after you categorize data sets into their different uses and types, you discover that 100 or more of these data sets are identified as PL/I source code.
- An analysis of the PL/I source data set names shows:
 - 50% contain the qualifier PLISRCE in their names.
 - 25% contain the qualifier SRCELIB in their names.
 - 15% contain the qualifier PLILIB in their names.
- Consequently, you adopt PLISRCE as the standard for all **new** PL/I source data sets created at your installation.
- You continue to recognize the data set qualifiers SRCELIB and PLILIB as acceptable names, and manage them in the same way you manage those data sets with the qualifier PLISRCE. This gives your users time to conveniently change their data set names. For example, they might rename these data sets in conjunction with a new release of an existing application.
- Require that any PL/I source code data sets failing to follow this naming convention either be renamed to the installation standard or excluded from system-managed volumes. Set a deadline.

The objective of this is to avoid creating new naming conventions and to limit renaming data sets. However, you might require that you accept some deviations from standard naming practices. For example, your ACS routines must be flexible enough to filter on **any** level qualifier within a data set name, rather than just the LLQ.

Improving Catalog Performance

Integrated catalog facility catalogs are a prerequisite for system-managed storage. They offer significant advantages over OS CVOLs and VSAM catalogs and are a functional replacement for these catalogs.

If you are using CVOLs and VSAM catalogs, you need to identify them for conversion to integrated catalogs. *MVS/ESA SML: Implementing System-Managed Storage* presents the procedure and some recommendations for this task.

As part of the conversion process, you need to remove JOBCAT and STEPCAT DD statements from JCL. This step might increase the size of some catalogs. “Using Multilevel Aliases” explains how to resolve problems associated with large catalogs.

You also need to consider the effect on catalog performance. “Specifying Catalog Performance Requirements” on page 32 recommends using storage class and IDCAMS DEFINE command parameters to maximize performance.

Using Multilevel Aliases

Based on your current catalog structure, removing JOBCAT and STEPCAT DD statements could result in an excessive number of data set entries under one HLQ, and cause device contention problems. Sharing data between remote sites can also cause catalog contention. If you have a catalog containing data set entries for many different applications, a catalog failure will affect all of these applications.

Use the multilevel alias (MLA) facility to provide a more flexible naming scheme and allow you to split large catalogs. With MLA, you can define catalog aliases of up to four qualifier levels.

With MLA, catalog selection is based on a right-to-left search of the qualifiers in the data set name for a matching alias. You specify the number of qualifiers to be used on a system in the SYSCATnn member of SYS1.NUCLEUS or the LOADxx member of SYS1.PARMLIB. If you are sharing catalogs across systems, the master catalogs on each system must support the same number of aliases.

The example in Figure 21 illustrates how data sets are cataloged according to MLAs:

Figure 21. Determining Catalog Use With MLA

Aliases in Master Catalog	Corresponding User Catalog	Data Set Names
USER1	SYS1.UCAT.ONE	USER1.JCL.CNTL
USER1.TEST	SYS1.UCAT.TWO	USER1.TEST.JCL.CNTL USER1.TEST.DATA
USER1.PROD	SYS1.UCAT.THREE	USER1.PROD.CNTL USER1.PROD.DATA1

This example defines three multilevel aliases associated with the HLQs of USER1, USER1.TEST, and USER1.PROD. Examples of fully-qualified data set name entries are shown for each alias, as well as the appropriate catalog where the data sets would be cataloged. It also shows how an existing catalog, based on the HLQ USER1, can be split into three smaller catalogs.

You can change the current MLA search level with the operator command shown in Figure 22, where *n* is the alias level with the value 1, 2, 3, or 4. This command is only in effect until the next IPL.

```
MODIFY CATALOG,ALIASLEVEL(n)
```

Figure 22. Using Multilevel Alias Levels

For more information on using multilevel aliases, see *DFSMS/MVS V1R2 Managing Catalogs*.

Specifying Catalog Performance Requirements

The physical placement of the integrated catalog facility catalog directly controls its performance characteristics. We recommend that you allocate the catalog to system-managed storage, and use the following performance attributes in storage class:

Figure 23. Specifying Storage Class Performance Attributes for Catalogs

Storage Class Attribute	Recommended Value
Direct Millisecond Response	between 5 and 10 milliseconds
Direct Bias	-
Sequential Millisecond Response	10 milliseconds
Sequential Bias	-

For detailed information on storage class performance attributes, see “Using Storage Class to Set Performance Goals” on page 83. For sample storage classes, see *MVS/ESA SML: Implementing System-Managed Storage*.

When you define catalogs, consider the performance parameters. The following are recommended parameter values for the IDCAMS DEFINE command. They apply to a catalog containing an average number of entries; make adjustments according to the number of entries to be contained in the catalogs and the rate of access to the catalog.

- STRNO

Specify STRNO(3). Allows three concurrent requests to the catalog.

You can judge the suitability of this value by using Resource Management Facility (RMF) to observe enqueues resource SYSZRPLW.*catname*, where *catname* is the name of a catalog. An indication is given by the I/O service times, not necessarily the number and frequency of enqueues for this name.

- BUFFERS

BUFNI/BUFND: Specify a value equal to the STRNO value + 1. Using the suggested values for STRNO, this is BUFND(4) and BUFNI(4).

- FREESPACE

Specify FREESPACE(20 20). Having free space within a data set reduces the number of CI/CA splits.

- IMBED

Places the lowest level of the index physically adjacent to the data it references, which results in less seek time than would normally result when this option is not used. If the catalog resides on cached DASD, you will probably get no performance advantage from IMBED.

- REPLICATE

Causes each nonembedded index control interval to be replicated around a track in the index space. This results in less time spent searching a track for an index control interval. If the catalog resides on cached DASD, you will probably get no performance advantage from REPLICATE.

Avoid REPLICATE without the IMBED option. Otherwise, DASD space will be used to replicate each index record without placing the sequence set records into the data component of the catalog. Rotational delay is reduced, but DASD seeks between the index and the data components result.

- CONTROLINTERVALSIZE

Specify 4096 bytes for the control interval size for the data component of the BCS. This is a compromise between conservation of buffer space and avoidance of spanned records.

- SHAREOPTIONS

To maintain integrity if the catalog is shared among users and systems, specify SHAREOPTIONS(3 4). This permits UPDATE access to one user and READ access to others.

Figure 24 shows an example of allocating a catalog, using the recommended parameter values.

```
//DEFCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  DEFINE USERCATALOG -
    ( NAME(SMLTEST.ICFCAT.VPRM901) -
      VOLUME(PRM901) -
      CYLINDERS(20 20) -
      FREESPACE(20 20) -
      ICFCATALOG -
      IMBED REPLICATE -
      SHAREOPTIONS(3 4) /* FOR SHARED CATALOGS */ -
      STRNO(3) ) -
    DATA ( CONTROLINTERVALSIZE(4096) BUFND(4) ) -
    INDEX ( BUFNI(4) )
/*
```

Figure 24. Allocating an Integrated Catalog Facility Catalog

For more information on IDCAMS DEFINE command, see *DFSMS/MVS V1R2 Access Method Services for ICF*.

Simplifying JCL

Several JCL keywords can help you simplify the task of creating data sets and make the allocation process under TSO more consistent. In addition, you can allocate VSAM data sets through JCL.

With the use of data classes, you have less use for the JCL keywords:

VOL
UNIT
DCB
AMP

Continue to use the DSN, DISP and SPACE keywords to create system-managed data.

JCL Keywords Used in the DFSMS Environment

You can use these JCL keywords to create VSAM and non-VSAM data sets. They are used only in the DFSMS environment. For a detailed description of the keywords and their use, see *MVS/ESA JCL User's Guide*.

Keyword	Description
DATACLAS	<p>Data class is the template for data set allocation. Provides a facility for specifying RECOR, LRECL, RECFM, KEYLEN, and KEYOFF. Replaces DCB and AMP.</p> <p>These rules apply:</p> <ul style="list-style-type: none">• If DATACLAS is specified or defaulted for an unsupported data set, it is ignored.• If DATACLAS is specified for an existing system-managed or non-system-managed data set, the keyword is ignored.
STORCLAS	<p>Storage class specifies level of service desired. Replaces VOL and UNIT. If you use the STORCLAS keyword, you cannot use the VOL=REF keyword.</p> <p>These rules apply:</p> <ul style="list-style-type: none">• If STORCLAS is given for an existing data set, the keyword is ignored. This applies to both system-managed and non-system-managed data sets.• If STORCLAS is explicitly specified for a new data set that cannot be system-managed, the keyword is ignored, and a warning message displays.
MGMTCLAS	<p>Management class specifies management criteria. Replaces DFSMSshm parameters for volume and system.</p> <p>These rules apply:</p> <ul style="list-style-type: none">• If MGMTCLAS is given for an existing data set, the keyword is ignored. This applies to both system-managed and non-system-managed data sets.• If MGMTCLAS is explicitly specified for a new data set that is not system-managed, the keyword is ignored, and a warning message displays.

AVGREC Average record defines the scale modification to an average block request on the SPACE keyword. Possible values are:

- U** Use primary and secondary quantities as is.
- K** Multiply primary and secondary quantity by 1 024.
- M** Multiply primary and secondary quantity by 1 048 576.

AVGREC is ignored if the SPACE keyword specifies anything other than an average block request.

RECORDG Record organization specifies the type of data set:

- KS** VSAM key sequenced
- ES** VSAM entry sequenced
- RR** VSAM relative record
- LS** VSAM linear

If both RECORDG and RECFM are omitted, then DSORG defaults to physical sequential or partitioned data set.

If you use the RECORDG keyword, you cannot use the RECFM attribute.

RECFM Record format. Replaces the RECFM field in the DCB.

If you use the RECFM keyword, you cannot use the RECORDG attribute.

KEYOFF Key offset specifies the offset to key. This is similar to the relative key position (RKP) field in the DCB.

KEYLEN Key length. Replaces the KEYLEN field in the DCB.

RETPD Retention period is the number of days during which a data set cannot be deleted without using the PURGE parameter on the DELETE command. Replaces the RETPD field in label.

If you use the RETPD keyword, you cannot use the EXPDT attribute.

Note: DFSMSHsm can migrate the data set, but does not delete it before the ending date.

EXPDT Expiration date says how long a data set remains accessible on the system. This is similar to the RETPD keyword above. However, EXPDT uses an expiration date rather than a retention period. Replaces the EXPDT field in the label.

If you use the EXPDT keyword, you cannot use the RETPD attribute.

LIKE Like requires that the properties of an existing cataloged data set are used to allocate a new data set. Replaces DCB=*dsname*.

The following properties are copied:

- RECORDG
- SPACE
- LRECL
- RECFM
- KEYLEN
- KEYOFF

If you use the LIKE keyword, you cannot use the REFDD attribute.

- REFDD** Reference DD indicates that both the properties on the JCL statement and from the data class of a previous DD statement are used to allocate a new data set. Replaces DCB=*.*ddname*. The same properties listed for the LIKE keyword are copied.
- If you use the REFDD keyword, you cannot use the LIKE keyword.
- SECMODEL** Security model specifies the name of a *generic* or *discrete* model profile to be used by RACF to create a discrete profile for the data set. For information on RACF security profiles, see Chapter 8, “Protecting Data” on page 145.

Sample Jobs Using the JCL Keywords

This section gives examples of how to use JCL keywords when:

- Creating a data set
- Creating a VSAM cluster
- Specifying retention period
- Specifying expiration date

Allocating Data

Figure 25 shows an example of JCL used to create a data set in a system-managed environment.

```
//NEWDATA DD DSN=DEPTM86.NEW.PLI,DISP=(,CATLG),
//          SPACE=(1,(5,5)),AVGREC=M,
//          RECFM=VB,LRECL=80
```

Figure 25. Creating a Data Set in a DFSMS Environment

These are some characteristics of the JCL in a system-managed environment:

- The device-dependent volume serial and unit information is no longer required, because the volume is assigned within a storage group selected by the ACS routines. This eliminates the need for device awareness.
- The LRECL and RECFM parameters are independent keywords. This makes it easier to override individual attributes that are assigned default values by the data class, selected by the ACS routines, that might not be appropriate for the data set being allocated.
- The SPACE parameter is coded with the average number of bytes per record, and the number of bytes required for the data set allocation. These are the values that the system uses to calculate the least number of tracks required for the space allocation. This also eliminates the need for device awareness, replacing the TRK or CYL unit specification.

Note: For variable block data sets, the average number of bytes per record is not necessarily the same as the LRECL value. In the example, the average record length is 1, whereas the LRECL is 80.

The AVGREC attribute indicates the scale factor for the primary and secondary allocation values. In the example, an AVGREC value of M indicates that the primary and secondary values of 5 are each to be multiplied by 1 048 576.

The SPACE parameter would result in a primary allocation of 5 MB and a secondary allocation of 5 MB.

The JCL to allocate a data set under SMS is simpler and has no device-dependent keyword. For information on managing data allocation, see Chapter 3, “Managing Data Allocation” on page 39.

Creating a VSAM Cluster

A VSAM cluster can be created as Figure 26 shows:

```
//VSAMDATA DD DSN=DEPTM86.NEW.VSAM,DISP=(,CATLG),  
//          SPACE=(1,(2,2)),AVGREC=M,  
//          RECOrg=KS,KEYLEN=17,KEYOFF=6,LRECL=80
```

Figure 26. Creating a VSAM Cluster in the DFSMS Environment

In the DFSMS environment, you can allocate VSAM data sets using JCL or data class. In JCL, DISP=(OLD,DELETE) is no longer ignored for VSAM data sets, and causes the VSAM data set to be deleted at deallocation. A data set with a disposition of MOD is treated as a NEW allocation if it does not already exist; otherwise, it is treated as an OLD allocation.

You cannot use certain parameters in JCL when allocating VSAM data sets, although you can use them in the IDCAMS DEFINE command. For a list, see Appendix A, “VSAM Data Set Allocation Parameters” on page 209.

Using Retention Period and Expiration Date

Parameters for specifying retention period and expiration date control the time during which a data set is protected from being deleted by the system. For example, the DD statement in Figure 27 protects the data set from deletion for 365 days:

```
//RETAIN DD DSN=DEPTM86.RETPD.DATA,DISP=(,CATLG),RETPD=365
```

Figure 27. Specifying Retention Period

The DD statement in Figure 28 protects the data set from deletion until January 1, 1995:

```
//RETAIN DD DSN=DEPTM86.EXPDT.DATA,DISP=(,CATLG),EXPDT=95001
```

Figure 28. Specifying Expiration Date

The RETPD and EXPDT keywords apply alike to system-managed and non-system-managed data sets.

The VTOC entry for both non-VSAM and VSAM data sets contains the expiration date used in the JCL, the TSO ALLOCATE command, or the IDCAMS DEFINE command. The expiration date can also come from the data class definition. The expiration date is placed in the VTOC either directly from the date specification, or after it is calculated from the retention period specification. The expiration date in the catalog entry exists for information purposes only.

You can use management class to limit or ignore RETPD and EXPDT parameters given by a user. If a user specifies values that exceed the maximum allowed by the management class definition, the retention period is reset to the allowed maximum. For more information on using management class to control retention period and expiration date, see “Deleting Data” on page 62.

Chapter 3. Managing Data Allocation

Inefficient space usage and poor data allocation cause problems with space and performance management. In a DFSMS environment, you can enforce good allocation practices to help reduce some of these problems. This chapter describes how to do this by:

- Establishing allocation standards
- Ensuring device independence
- Allocating data sets as PDSEs
- Processing GDGs
- Allocating Compressed Data

Establishing Allocation Standards

This section describes how to use data class to establish standards for data allocation. For sample data classes, descriptions, and ACS routines, see *MVS/ESA SML: Implementing System-Managed Storage*.

Using Data Class to Standardize Data Allocation

You can define data classes containing standard data set allocation attributes. Users then only need to use the appropriate data class names to create standardized data sets. To override values in the data class definition, they can still provide specific allocation parameters.

Data classes can be determined from the user-specified value on the DATACLAS parameter, from a RACF default, or by ACS routines. Data classes can also be determined for objects by a specification using the SETOAM command in the CBROAMxx member of SYS1.PARMLIB. However the data class is determined, it is assigned to the data by the data class ACS routine. ACS routines can also override user-specified or RACF default data classes.

Data Class Attributes

The data class attributes you can use are:

- Data class name
- Data class description
- Data set organization (RECORG)
- Data set name type (DSNTYPE)
- Record format (RECFM)
- Logical record length (LRECL)
- Key length (KEYLEN)
- Key offset (KEYOFF)
- Space attributes (AVGREC, AVE VALUE, PRIMARY, SECONDARY, DIRECTORY)
- Retention period or expiration date (RETPD or EXPDT)
- Number of volumes the data set can span (VOLUME COUNT)
- Allocation amount when extending VSAM extended data set (ADDITIONAL VOLUME AMOUNT)
- VSAM index options (IMBED or REPLICATE)
- Control interval size for VSAM data components (CISIZE DATA)
- Percentage of control interval or control area free space (% FREESPACE)

- VSAM share options (SHAREOPTIONS)
- Compaction option for data sets (COMPACTION)
- Tape media type (MEDIA TYPE)
- Tape media format requirements (RECORDING TECHNOLOGY)

You can specify the data class space attributes to control DASD space waste. For example, the primary space value should specify the total amount of space initially required for output processing. The secondary allocation allows automatic extension of additional space as the data set grows and does not waste space by over-allocating the primary quantity. You can also use the data class space attributes to relieve users of the burden of calculating how much primary and secondary space to allocate.

The COMPACTION attribute specifies whether data is to be compressed on DASD if the data set is allocated in the extended format. The attribute alone also allows you to use the improved data recording capability (IDRC) of your tape device when allocating tape data sets.

The MEDIA TYPE and RECORDING TECHNOLOGY attributes are used for tape data sets only. MEDIA TYPE allows you to select the mountable tape media cartridge type. RECORDING TECHNOLOGY allows you to select the format to use when writing to that device.

The read-compatible special attribute indicator in the tape device selection information (TDSI) allows an 18 track tape to be mounted on a 36 track device for read access. The attribute increases the number of devices that are eligible for allocation when you are certain that no more data will be written to the tape.

For detailed information on specifying data class attributes, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Planning and Defining Data Classes

Use your service-level agreement (SLA) for reference when you plan your data classes. (For information about SLAs, see *MVS/ESA SML: Leading a Storage Administration Group*.) SLAs identify users' current allocation practices and their requirements. For example, based on user requirements, you might create a data class to allocate standard control libraries. Figure 29 shows the values you might include in the library definition.

```

RECFM = FB
LRECL = 80
SPACE
  AVGREC = U
  AVG VALUE = 80
  PRIMARY = 5000
  SECONDARY = 5000
  DIRECTORY = 62

```

Figure 29. Sample Data Class Values

If you want the data class to supply the default value of a parameter, do not specify a value for that parameter in the JCL or dynamic allocation. Users cannot override the data class attributes of dynamically allocated data sets if you use the IEFDB401 user exit.

You can override a data class attribute using JCL or dynamic allocation parameters. However, overriding a subparameter of a parameter, will override ALL of the subparameters for that parameter. For example, SPACE=(TRK,(1)) in JCL will cause primary, secondary, and directory quantities, as well as AVGREC and AVGUNIT, in the data class to be overridden. DFSMS usually does not change values that are explicitly specified because doing so would alter the original meaning and intent of the allocation. There is an exception — if it is clear that a PDS is being allocated (DSORG=PO or DSNTYPE=PDS is specified), and no directory space is indicated in the JCL, then the directory space from the dataclass is used even though SPACE=(TRK,(1)) was specified.

Data class names should indicate the type of data they are assigned to. This makes it easier for users to identify the template they need to use for allocation.

You define data classes using the ISMF Data Class Application. Users can access the Data Class List panel to determine which data classes are available and the allocation values that each data class contains.

For more information on planning and defining data classes, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Ensuring Device Independence

To take full advantage of system-managed storage and improve space usage, you need to eliminate device dependencies. This allows the system to place data on the most appropriate device in the most efficient way. This section describes how to ensure device independence.

Removing Volume Serial Number Dependencies

In the DFSMS environment, you control volume selection through the storage class and storage group definitions you create, and by ACS routines. This means that users do not have to specify volume serial numbers with the VOL=SER parameter, or code a specific device type with the UNIT= parameter on their JCL. When converting data sets for use in DFSMS, they do not have to remove these parameters from existing JCL because volume and unit information can be ignored with ACS routines. However, work with users to evaluate UNIT and VOL=SER dependencies before conversion.

Note: If you keep the VOL=SER parameter for a non-SMS volume, but you are trying to access a system-managed data set, then SMS might not find the data set.

For information on converting to the DFSMS environment, see *MVS/ESA SML: Implementing System-Managed Storage*.

Using a Default Unit for Non-System-Managed Data

We recommend using ISMF Control Data Set Application DEFAULT UNIT parameter to make the UNIT parameter optional for new non-system-managed data sets. This ensures JCL used for data set creation is consistent for both system-managed and non-system-managed data sets.

Because non-system-managed data set allocations require unit specifications, the DEFAULT UNIT parameter can be used to assign a default unit during allocation if no unit is specified in JCL. This means users can omit the UNIT parameter and

their allocation request will not fail even if it is directed to non-system-managed storage. For system-managed data sets, the DEFAULT UNIT parameter is ignored.

To use DEFAULT UNIT, you can specify an esoteric or generic device name SMS can use when no UNIT parameter is specified in the JCL for a new non-system-managed data set. For example, if you define a default unit, users can specify the JCL in Figure 30 to use the default unit if the allocation is directed to non-system-managed storage. If you do not define a default unit, the JCL in Figure 30 will fail if the allocation is directed to non-system-managed storage.

```
//SYSDATA DD DSN=SYSA.TEST.DATA,DISP=(,CATLG),VOL=SER=D65DM3,  
//          SPACE=(23200,(10,2),RLSE),AVGREC=U
```

Figure 30. Removing the UNIT Parameter in JCL

Note:

If you specify an esoteric or generic device name in either the UNIT parameter or the DEFAULT UNIT for a non-system-managed data set, then make certain that it is defined on the system performing the allocation. Although the esoteric name is ignored for a system-managed data set, the esoteric must still exist for non-system-managed data sets, including DD DUMMY or DD DSN=NULLFILE statements that include the UNIT parameter.

For object data written to tape, the TAPEUNITNAME parameter on the SETOAM command in the CBROAMxx member in SYS1.PARMLIB is used to request the use of an esoteric or generic device type. This is a required parameter for any object or object backup storage group whose objects are written to tape.

Converting Track and Cylinder Allocation Requests to Byte Requests

In the DFSMS environment, you use default device geometry to convert track and cylinder allocation requests to megabyte or kilobyte requests. Default device geometry is used for the allocation of a new system-managed data set when a generic UNIT has not been explicitly specified. This can only be used for system-managed data sets. Using default device geometry isolates users from actual physical devices; the amount of allocated space is consistent, regardless of which device is used.

You use the ISMF Control Data Set Application to specify a default device geometry. For more information on how to specify both default unit and default device geometry, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Requesting Nonspecific Multivolume Data Allocation

SMS allows both nonspecific and specific volume allocation for multivolume data sets. All system-managed data set allocations are nonspecific, except those that are assigned a storage class with GUARANTEED SPACE = YES and which specify specific volume serials. Volume serial numbers will be ignored unless you specify GUARANTEED SPACE = YES.

To request a nonspecific multivolume data set allocation, you can either specify a volume count, a unit count, or a data class with a predefined volume count in the JCL. You can also be automatically assigned a data class by the data class ACS routine. These are some of the many methods you can use to request a nonspecific multivolume data set allocation.

All of the volumes selected for a multivolume data set allocation belong to the same storage group. For nonspecific multivolume allocations, if the storage group assigned to the data set does not contain as many volumes as indicated by the volume count, the allocation can still proceed. This is not the case when you request a specific multivolume allocation with `GUARANTEED SPACE = YES`.

If you use nonspecific allocation to create a multivolume data set, SMS selects the first volume and then selects other volumes as needed. The primary space request will be allocated only on the first volume. With `GUARANTEED SPACE = YES`, SMS will allocate the primary space on each of the volumes for specific and nonspecific requests.

Figure 31 shows an example of JCL that uses nonspecific allocation. SMS selects the volumes for the allocation. If you specify `GUARANTEED SPACE = YES` for the data set,

this example allocates 100MB on each of the five volumes. When all of the primary space is used, the 25MB secondary space is allocated on the last volume.

If you specify `GUARANTEED SPACE = NO` for the data set, the primary and secondary space is allocated on the first volume. For subsequent volumes, space is allocated differently for VSAM and non-VSAM data sets. For VSAM data sets, the primary and secondary space is allocated on each of the subsequent volumes. For extended format VSAM data sets, the data class `ADDITIONAL VOLUME AMOUNT` attribute determines if a primary or secondary allocation amount is used when extending to the new volume. For non-VSAM data sets, only the secondary extents are allocated on the subsequent volumes.

```
//ENGINEER DD DSN=DEPTM86.ENG.MULTIFILE,DISP=(,CATLG),LRECL=100,  
//          SPACE=(1,(100,25)),AVGREC=K,RECFM=VB,DSORG=PS,  
//          UNIT=(,5)
```

Figure 31. Using Nonspecific Allocation for System-Managed Multivolume Data Sets

The following examples show various ways you can request nonspecific allocations for multivolume data sets without specifying a device type. None of the following examples use or require storage classes with `GUARANTEED SPACE=YES`.

In Figure 32, only a volume count is specified in the JCL.

```
//FINANCE DD DSN=DEPTM86.FIN.MULTIFILE,DISP=(,CATLG),LRECL=100,  
//          SPACE=(200,(100,50)),AVGREC=K,RECFM=VB,DSORG=PS,  
//          VOL=(,,4)
```

Figure 32. Specifying Volume Count Only in JCL for Nonspecific Allocation

In Figure 33 on page 44, both the unit count and the volume count parameters are specified. Note that when the unit count value is different from the volume count value, the higher value is used. In this example, a volume count of 5 is used.

```
//PAYROLL DD DSN=DEPTM86.PAY.MULTIVOL,DISP=(,CATLG),DSORG=PS,  
//          SPACE=(80,(100,25)),AVGREC=K,LRECL=255,RECFM=VB,  
//          VOL=(,,5),UNIT=(,3)
```

Figure 33. Specifying Volume Count and Unit in JCL for Nonspecific Allocation

Finally, you can also perform a nonspecific allocation by specifying a data class in the JCL that has a volume count already defined in it. The example in Figure 34 shows the data class, TMMACTV, with a predefined volume count:

```
//DATA DD DSN=DEPTM86.GEN.MULTIFILE,DISP=(,CATLG),DSORG=PS,  
//      SPACE=(132,(100,50)),AVGREC=K,RECFM=VB,LRECL=255,  
//      DATACLAS=TMMACTV
```

Figure 34. Specifying a Data Class with Volume Count for Nonspecific Allocation

You can use data class to provide nonspecific multivolume allocations for very large databases and associated data sets. A storage class defined with GUARANTEED SPACE=YES is not required to allocate a database data set that is multivolume.

Requesting Specific Multivolume Data Allocation

For special requirements, you use the storage class GUARANTEED SPACE attribute to allow specific volume allocation and preallocation of space on system-managed volumes. When GUARANTEED SPACE=YES, SMS honors a VOL=SER allocation for a new data set. This allows you to create a data set on a specific volume or to preallocate a multivolume data set.

Restrict the use of GUARANTEED SPACE to the storage administrators, system programmers, and database administrators. Also, limit the number of storage classes that have this attribute and use ACS routines and RACF to control the use of these classes.

Note: A specific multivolume request is successful only if the volume serial numbers you specify actually exist; and you defined all the volume serial numbers to the same storage group; and the storage group ACS routine can derive the storage group to which the specified volumes belong.

The JCL in Figure 35 on page 45 is an example of using specific allocation for a system-managed multivolume sequential data set. SMS allocates the data set on volumes PRM901, PRM902, PRM903, PRM904, and PRM905. The storage class GSPACE should specify GUARANTEED SPACE=YES. In this example, SMS initially allocates 100KB on each of the five volumes. When all of the primary space is used, a 75KB secondary extent is allocated on the last volume (PRM905).

```
//SPECIAL DD DSN=DEPTM86.ENG.MULTIFILE,DISP=(,CATLG),DSORG=PS,  
//          SPACE=(1,(100,75)),AVGREC=K,  
//          LRECL=100,RECFM=VB,STORCLAS=GSPACE,  
//          VOL=SER=(PRM901,PRM902,PRM903,PRM904,PRM905)
```

Figure 35. Using Specific Volume Allocation for System-Managed Multivolume Data Sets

Using Device Independent Allocation

One way users have improved DASD space utilization is to make allocations device independent. This was achieved by using average blocks as the allocation unit rather than specifying cylinders or tracks. Average block allocation has the following advantages over track and cylinder allocation:

- Easier migration to new devices by making JCL and CLISTs device independent
- More efficient use of space
- Better enables users to think of space in terms of the application's requirements for storage (for example, the number of records and their average size)

In a system-managed storage environment, device independent allocation is enhanced through the use of specific JCL keywords on DD allocation statements. This allocation method requires you to specify the average record size for the data set on the SPACE parameter. This value is the same as the LRECL for data sets having fixed-length records and represents an average record size for data sets with variable-length records.

The total number of records in the data set is specified using the primary and secondary quantities on the DD allocation statement. The system then calculates the least number of tracks necessary to contain the number of records specified.

You can also specify the AVGREC keyword on the DD allocation statement to allocate in average records instead of average blocks. This causes the first SPACE parameter is interpreted as the size of an average record instead of the size of an average block. The value you specify for AVGREC is the scale factor used for the primary and secondary values. For example, an AVGREC value of K indicates that the primary and secondary allocation values are multiplied by 1024. To allocate enough space for a data set with 10240 records of 80 bytes each, write an allocation statement as Figure 36 shows:

```
SPACE=(80,(10,1),RLSE),AVGREC=K
```

Figure 36. Using Average Record Length and the AVGREC Subparameter

Implementing Device Independent Allocation as Installation Standard

Data class provides an easy way to implement allocation in terms of records as an installation standard. Figure 37 on page 46 is an example of using ISMF to specify both average record length and AVGREC in a data class definition. Like the JCL in Figure 36, the data class definition allocates enough space for a data set with 10240 records of 80 bytes each.

```

                                DATA CLASS DEFINE                                Page 1 of 2
COMMAND ==>

SCDS NAME:      ACTIVE
DATA CLASS NAME: SRCFLIB

TO DEFINE DATA CLASS, SPECIFY:
DESCRIPTION ==>
====>
RECORG          ==>      (KS, ES, RR, LS or blank)
RECFM           ==>      (any valid RECFM combination or blank)
LRECL           ==>      (1 to 32761 or blank)
KEYLEN          ==>      (0 to 255 or blank)
KEYOFF          ==>      (0 to 32760 or blank)
SPACE AVGREC    ==> K    (U, K, M or blank)
                ==> 80   (0 to 65535 or blank)
                ==> 10   (0 to 999999 or blank)
                ==> 1    (0 to 999999 or blank)
                ==>      (0 to 999999 or blank)
DATA SET NAME TYPE ==>      (EXT, HFS, LIB, PDS or blank)
                If EXT ==>      (P=Preferred, R=Required or blank)

USE ENTER TO PERFORM VERIFICATION; USE DOWN
COMMAND TO VIEW NEXT PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

```

Figure 37. Specifying Average Block Size in Data Class Definition Panel

Requesting System-Determined Block Size

Block sizes are chosen to accommodate the size of the records created, to meet the I/O performance needs of the application, to use space effectively, and to guarantee device independence. Using a system-determined block size ensures both device independence and efficient use of space.

We recommend using system-determined block size for most non-VSAM physical sequential system-managed or non-system-managed DASD and tape data sets.

To use system-determined block size, you can omit the block size specification, or specify `BLKSIZE=0`. This ensures that data sets are allocated with optimum block sizes for the target device. The block size is determined by record format, logical record length, and geometry of the target device.

The data set is also marked reblockable. This means that the block size is recalculated if the system moves the data set to a different device type. To force a system-determined block size at OPEN time, you can modify the DCB open installation exit.

For more information on using system-determined block size, see *DFSMS/MVS V1R2 Using Data Sets*. For detailed information on specifying system-determined block size, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Note: System-determined block size is not used in all situations because some data sets, such as load modules, cannot or should not be reblocked.

Although using system-determined block sizes is the most efficient way to guarantee device independence and efficient use of space, some users might continue to specify a value for the DCB `BLKSIZE` subparameter. If they do, choosing effi-

cient block sizes is important. Block sizes that are too small waste space in overhead (such as interblock gaps). For example, on a 3390, a block size of 80 bytes uses only 12% of the track's capacity. A block size larger than half a track, though, prevents the placement of another equal length record on the track and the residual space is lost; the amount of residual space decreases as block size approaches one track. For more information about how different block sizes affect space use, see *Using 3390 in an MVS Environment*. If block sizes of data sets are significantly increased, you might need to increase the region size of programs using the data sets for increased buffer storage.

Evaluating Existing Data Block Sizes

In addition to using a system-determined block size for new data sets, users can evaluate whether their existing data sets have efficient block sizes. Using the ISMF Data Set Application, you can construct a filtered list of data sets for which ISMF provides useful status information, including efficient block sizes. For more information on ISMF, see *DFSMS/MVS V1R2 Using ISMF*.

Reblocking Data Using System-Determined Block Sizes

You can use DFSMSdss to reblock existing data sets as necessary. With the DFSMSdss COPY or RESTORE commands and the REBLOCK parameter, you reblock data sets using a system-determined block size. If you use the DFSMSdss reblock exit routine, you might want to set the exit return code to zero to ensure that a system-determined block size is used.

If the CONVERSION parameter of the DFSMSHsm SETSYS command is set, reblocking can also be done automatically during the MIGRATE and RECALL processes. You can also specify the OPTIMUMDASDBLOCKING parameter to allow DFSMSHsm to use a defined optimum block size when writing to DFSMSHsm-owned DASD.

DFSMSHsm, which provides support for using DFSMSdss for moving data sets between user-owned and DFSMSHsm-owned volumes, also provides DFSMSdss reblock support for current PDSs. DFSMSHsm requests DFSMSdss to reblock a PDS, if you specify the REBLOCKTOANY parameter on the SETSYS CONVERSION command.

Allocating Partitioned Data Sets Extended

Partitioned data sets extended (PDSEs) are a type of system-managed data set. You can use them for source (programs and text) libraries, macros, and program object libraries. A PDSE, however, is a data set that has an improved indexed directory structure and a different member format. It has the following advantages over the PDS:

- The size of a PDSE directory is flexible and can expand to accommodate the number of members stored in it. (The size of a PDS directory is fixed at allocation time.)
- PDSE members are indexed in the directory by member name. This eliminates the need for time-consuming sequential directory searches.
- The logical requirements of the data stored in a PDSE are separated from the physical (storage) requirements of that data, which simplifies data set allocation.

- PDSEs provide more efficient use of DASD space. For example, by moving or deleting a PDSE member, you free space that is immediately available for the allocation of a new member, without first having to compress the data set to consolidate the fragmented space for reuse. This makes PDSEs less susceptible to space-related abends than PDSs.
- The number of PDSE members stored in the library can be large or small without concern for performance or space considerations.
- The ability to update a member in place is possible with PDSs and PDSEs. But with PDSEs, you can extend the size of members and the integrity of the library is maintained while simultaneous changes are made to separate members within the library.
- The maximum number of extents of a PDSE is 123; the PDS is limited to 16.
- PDSEs are device-independent because they do not contain information that depends on location or device geometry.
- All members of a PDSE are reblockable.
- PDSEs can contain program objects built by the program management binder that cannot be stored in PDSs.

You can also share PDSEs within and across systems. With systems that support PDSEs (MVS/DFP 3.2.0 or higher level), multiple users are allowed to read PDSE members while the data set is open.

If you have DFSMS/MVS installed, you can extend the sharing to enable multiple users on multiple systems to concurrently create new PDSE members and read existing members.

Using the PDSESHARING keyword in the SYS1.PARMLIB member, IGDSMSxx, you can specify:

NORMAL	Allows multiple users to read any member of a PDSE.
EXTENDED	Allows multiple users to read any member or create new members of a PDSE.

All systems sharing PDSEs need to be upgraded to DFSMS/MVS to use the extended PDSE sharing capability.

After updating the IGDSMSxx member of SYS1.PARMLIB, you need to issue the SET SMS ID=xx command for **every** system in the complex to activate the sharing capability.

For additional information on PDSEs, see *DFSMS/MVS V1R2 Using Data Sets*.

Although SMS supports PDSs, you should consider converting these to the PDSE format. The following sections describe this process.

Converting Existing PDSs to PDSEs

Converting PDSs to PDSEs is beneficial, but be aware that certain data sets are unsuitable for conversion to, or allocation as, PDSEs, because the system does not retain the original block boundaries. Also, data sets requiring device dependency are inappropriate to convert or allocate because PDSEs are device-independent.

To reclaim unused space in those data sets that cannot be converted, use the DFSMSdss COMPRESS command to compress PDSs in place. This consolidates space that is no longer used within a PDS and makes it available at the end of the data set. For large or critical PDSs, you might want to copy or back up the data sets before you compress them. This maintains data set availability should the compress fail.

General-Use Programming Interface

Selecting Data Sets to Allocate as PDSEs

As a storage administrator, you can code appropriate ACS routines to select data sets to allocate as PDSEs and prevent inappropriate PDSs from being allocated or converted to PDSEs.

By using the &DSNTYPE read-only variable in the ACS routine for data class selection, you can control which PDSs are to be allocated as PDSEs. The following values are valid for DSNTYPE in the data class ACS routines:

- &DSNTYPE = 'LIBRARY' for PDSEs.
- &DSNTYPE = 'PDS' for PDSs.
- &DSNTYPE is not specified. This indicates that the allocation request is provided by the user through JCL, the TSO/E ALLOCATE command, or dynamic allocation.

If you specify a DSNTYPE value in the JCL, and a different DSNTYPE value is also specified in the data class selected by ACS routines for the allocation, the value specified in the data class is ignored.

The following sections explain how to use ACS routines to filter on PDSEs.

Data Class Filtering: By coding the appropriate data class ACS routines, you prevent certain data sets from being allocated as PDSEs. Figure 38 on page 50 shows a sample data class ACS routine that prevents this situation.

```

PROC DATACLAS

/*****
/* ACS DATA CLASS ROUTINE EXTRACT TO PREVENT A LOAD MODULE DATA SET  */
/* FROM BEING ALLOCATED AS A PDSE                                     */
/*                                                                    */
/* NOTE TO READER.                                                 */
/* THE LOADLIB DATA CLASS IS DEFINED WITH A DATA SET NAME TYPE   */
/* (DSNTYPE) OF PDS.                                             */
*****/

SELECT
    WHEN ((&DSORG = 'PO' && &LLQ = 'LOAD') && &DSNTYPE <=> 'PDS')
        SET DATACLAS = 'LOADLIB'

    WHEN (&DSNTYPE <=> 'LIBRARY' && &LLQ = 'LOAD'
        )
        SET DATACLAS = 'LOADLIBE'

.
.
.
.
END /* END OF SELECT */

.
.
.
.
END /* END OF DATACLAS PROC */

```

Figure 38. Selecting PDSE Candidates Using Data Class ACS Routine

Using this type of filtering requires good data set naming standards and practices. This filtering allows you to identify data sets to be excluded from being allocated as PDSEs; you can filter on the lowest-level qualifier of the data set name in the ACS routine. The organization of the data set (DSORG) can also be used as the filtering criteria, as Figure 38 shows.

Storage Class Filtering: You can use ACS routines to filter on PDSEs and assign them to specific storage classes. For example, if a PDSE requires a storage class with GUARANTEED SYNCHRONOUS WRITE, then you can use the storage class ACS routine in Figure 39 on page 51.

```

PROC STORCLAS

/*****
/* STORAGE CLASS ACS ROUTINE EXTRACT TO ASSIGN A PDSE TO A STORAGE */
/* CLASS WITH GUARANTEED SYNCHRONOUS WRITE */
*****/

SELECT
    WHEN (&DSNTYPE = 'LIBRARY' && &DATACLAS = 'SPECIAL')
        SET &STORCLAS = 'SYNCWRIT'
    .
    .
    .
    .
END /* END OF SELECT */
.
.
.
.
END /* END OF STORCLAS PROC */

```

Figure 39. Selecting PDSE Candidates Using Storage Class ACS Routine

_____ End of General-Use Programming Interface _____

Converting to PDSEs

Before converting a PDS to a PDSE, ask the data set owner if I/O synchronization to the media is a requirement for the designated data set. (I/O synchronization is the process of writing data from its buffer space to DASD. This is beneficial because you do not lose the data in the buffer space even if new data writes over existing data.) If so, the program should continue to use the SYNCDEV macro, or the dataset should be assigned to a storage class with GUARANTEED SYNCHRONOUS WRITE.

We recommend using the SYNCDEV macro because it does not involve unnecessary serialization. It is especially important to use either SYNCDEV or the GUARANTEED SYNCHRONOUS WRITE option when a PDS requiring I/O synchronization to the media is converted to a PDSE. Otherwise, in certain situations, I/O synchronization to the media is not continued once the conversion takes place. For more information on converting PDSs to PDSEs, see *DFSMS/MVS V1R2 Using Data Sets*.

Many types of libraries are candidates for conversion to PDSE:

- PDSs that are updated often, and that require frequent and regular reorganization
- Large PDSs that require specific device types because of the size of allocation

We recommend using DFSMSdss to convert PDS-format data sets to PDSE-format data sets. However, you can also use IEBCOPY for conversion.

By using the DFSMSdss COPY function with the CONVERT and PDS keywords, you can convert a PDSE back to a PDS. This is especially useful if you need to prepare a PDSE for migration to a site that does not support PDSEs.

Figure 40 on page 52 shows the DFSMSdss COPY command being used to convert a PDS to a PDSE. For more information on DFSMSdss, see *DFSMS/MVS V1R2 DFSMSdss Storage Administration Reference* or *DFSMS/MVS V1R2 DFSMSdss Storage Administration Guide*.

```
//CONVERT EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
COPY DATASET( INCLUDE(DEPTM86.X15PROJ.** ) -
              EXCLUDE(DEPTM86.**.LOAD) -
              BY(DSORG,EQ,PDS) ) -
CONVERT(PDSE(**)) -
DELETE
/*
```

Figure 40. Converting PDS to PDSE Using DFSMSdss COPY

Allocating New PDSEs

You can allocate PDSEs only on a system-managed volume. Users can allocate a PDSE using the JCL keyword DSNTYPE, a data class, the ALLOCATE command, or by using IDCAMS ALLOCATE with the DSNTYPE keyword. The keyword specified is either DSNTYPE(LIBRARY) to allocate a PDSE, or DSNTYPE(PDS) to allocate a PDS. Figure 41 shows IDCAMS ALLOCATE used with the DSNTYPE(LIBRARY) keyword to allocate a PDSE:

```
//ALLOC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALLOCATE DSNAME('DEPTM86.X15PROJ.REL01.SCRIPT') -
        NEW -
        DSNTYPE(LIBRARY)
/*
```

Figure 41. Allocating PDSEs Using IDCAMS ALLOCATE

You cannot use the JCL keywords DSNTYPE and RECOGR together. A PDS and a PDSE can be concatenated in JCL DD statements, or by using dynamic allocation, such as the TSO ALLOCATE command.

A PDSE is allocated only if the following criteria are met:

- Directory space or DSORG=PO is specified on the JCL statement in a dynamic allocation request, in the TSO/E ALLOCATE command, or in the data class assigned to the data set.
- The keyword DSNTYPE(LIBRARY) is specified in the command, or provided either in the data class or in the SMS installation default.
- The data set to be allocated is assigned to system-managed storage through the ACS routines.

Identifying PDSEs

You can use ISMF to display information associated with data set name type (DSNTYPE). Figure 42 shows a sample data set list obtained through a catalog:

```

                                DATA SET LIST
COMMAND ==>                                SCROLL ==> HALF
                                           Entries 1-14 of 33
ENTER LINE OPERATORS BELOW:                Data Columns 27-29 of 34

LINE          DATA SET NAME          DATA SET  DATA SET  ENTRY
OPERATOR      DATA SET NAME          ENVIRONMENT NAME TYPE  TYPE
--(1)---     -----(2)-----          ----(27)--- --(28)--- --(29)---
              X15PROJ.REL01.PGM          MANAGED    LIBRARY    NONVSAM
              X15PROJ.REL01.LOAD          MANAGED    OTHERS     NONVSAM
              X15PROJ.REL01.CLIST          UNMANAGED  OTHERS     ALIAS
              X15PROJ.REL01.VSAM          UNMANAGED  -----    DATA
              X15PROJ.REL01.VSAM          UNMANAGED  -----    INDEX
              X15PROJ.REL01.GDS            MANAGED    -----    DEFERRED
              X15PROJ.REL01.DGTSLIB        UNMANAGED  OTHERS     NONVSAM
              X15PROJ.REL01.DGTTABL        UNMANAGED  OTHERS     NONVSAM
              X15PROJ.REL01.DUMP           UNMANAGED  -----    NONVSAM
              X15PROJ.REL01.DUMP.ABND50C4 UNMANAGED  -----    NONVSAM
              X15PROJ.REL01.INPUT          MANAGED    -----    NONVSAM
              X15PROJ.REL01.OUTLIST        UNMANAGED  -----    NONVSAM
              X15PROJ.REL01.ISPFILE        UNMANAGED  -----    NONVSAM
              X15PROJ.REL01.ISPPLIB        MANAGED    LIBRARY    NONVSAM

USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT;
CANCEL TO EXIT.

```

Figure 42. Listing Data Sets and Displaying DSNTYPE Information

Column 28 in Figure 42 lists the data set name type. “LIBRARY” indicates the data set is a PDSE, and “OTHERS” indicates that the data set is a PDS.

If you obtain the list through a catalog and column 28 contains nulls “-----” the data set name type is neither LIBRARY (PDSE) nor a PDS.

The valid values for data set name type in the ISMF Data Set Application are EXTENDED (extended sequential data sets), HFS (hierarchical file system data sets), LIBRARY (PDSEs), and OTHERS (data sets that are not allocated in the extended, HFS, or PDSE format). A saved data set list from a release prior to DFSMS/MVS can still be used and the values of data set name type matches those under DFSMS/MVS. For example, PDSs are indicated as OTHERS in column 28.

You can also display data set name type information through the ISMF Data Class List function. Figure 43 on page 54 shows an example of a data class list obtained using ISMF. Column 25 shows the DSNTYPE information:

```

COMMAND ==>>>                                DATA CLASS LIST
                                                SCROLL ==>>> PAGE
                                                Entries 1-12 of 33
                                                Data Columns 21-25 of 25

CDS NAME: ACTIVE

ENTER LINE OPERATORS BELOW:

LINE      DATACLAS  SHARE  LAST MOD  LAST DATE  LAST TIME  DATA SET
OPERATOR  NAME          XSYSTEM USERID    MODIFIED   MODIFIED   NAME TYPE
---(1)--- --(2)---  -(21)-- --(22)--  ---(23)--- --(24)--- --(25)---
          DATAF      -   STOADMIN  1988/06/20  10:05:19  -----
          DATAV      -   STOADMIN  1988/06/20  10:08:05  -----
          SRCFLIB     -   STOADMIN  1988/06/20  10:13:54  LIBRARY
          SRCVLIB     -   STOADMIN  1988/06/20  10:21:42  LIBRARY
          LOADLIB     -   STOADMIN  1988/06/20  10:29:01  OTHERS
          LISTING     -   STOADMIN  1988/06/20  10:37:17  -----
          ENTRY       -   STOADMIN  1988/06/24  14:47:31  -----
          KEYED       -   STOADMIN  1988/06/24  14:53:16  -----
          DIRECT      -   STOADMIN  1988/06/24  14:57:45  -----
          LINEAR      -   STOADMIN  1988/06/24  15:03:10  -----
          NORMAL      -   STOADMIN  1988/06/20  15:10:28  OTHERS

USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT;
CANCEL TO EXIT.

```

Figure 43. Listing Data Classes and Displaying DSNTYPE Information

In column 25, "LIBRARY" indicates the data set is a PDSE, and "OTHERS" indicates that the data set is a PDS. For more information on using ISMF, see *DFSMS/MVS V1R2 Using ISMF*.

Processing Generation Data Groups

SMS allows generation data groups (GDGs) that contain generation data sets (GDSs) on both system-managed and non-system-managed volumes. This section describes GDG and GDS processing in a DFSMS environment.

Defining and Allocating Generation Data Sets

To define a GDG base, use the IDCAMS DEFINE GENERATIONDATAGROUP command. The DEFINE GENERATIONDATAGROUP command does not assign SMS classes to the generation data sets in the generation data group. ACS processing assigns the SMS classes and storage groups to each generation data set as it does for any other type of data set.

To allocate a GDS on a system-managed volume, use the JCL DCB, LIKE, REFDD, or DATACLAS parameters. You can also let ACS routines assign a data class to allocate the data set. Figure 44 on page 55 is an example of using the DATACLAS parameter when allocating a generation data set on a system-managed volume. The data class named GDGV104 contains the required DCB and SPACE parameters.

```
//NEWGDS DD DSN=DEPTM86.DAILY.LOG(+1),DISP=(,CATLG),  
// DATACLAS=GDGV104
```

Figure 44. Using DATACLAS to Allocate GDSs on System-Managed Volumes

If your installation supports many GDSs with different space or DCB values, you can define data classes that satisfy most allocation requirements; the user can override specific values as necessary. Figure 45 shows an example of using the GDGV104 data class to satisfy an allocation requirement with LRECL=326.

```
//NEWGDS DD DSN=DEPTM86.DAILY.LOG(+1),DISP=(,CATLG),  
// DATACLAS=GDGV104,LRECL=326
```

Figure 45. Using Data Classes to Satisfy Allocation Requirements

A GDG can contain both system-managed and non-system-managed GDSs. To allocate a non-system-managed GDS to a GDG base that is defined in a system-managed catalog, use a data class or a model data set name that is different from the GDG base name.

Be aware that because SMS is choosing volumes for allocation of GDSs, all GDSs might not be accessible from all processors in a complex. As with all data sets, it is important to ensure correct DASD connectivity to the processors that require the data.

Because pattern data set control blocks (DSCBs) are not cataloged, you cannot use them to allocate generation data sets on system-managed volumes. If you create a GDG in system-managed storage, and its catalog is on a system-managed volume, you should remove any dependencies on pattern DSCBs. For information on identifying and removing GDG pattern DSCBs, see *MVS/ESA SML: Implementing System-Managed Storage*.

Cataloging Generation Data Sets

A system-managed DASD GDS is cataloged when it is allocated. Because cataloging of a new generation might cause the uncataloging of an old generation that is required by the job, the new generation is cataloged in a *deferred roll-in* state at the time of creation. This means that the data set is cataloged, but it is not included in the GDG sphere record until disposition processing occurs at the step that specifies DISP=(,CATLG). The values on the DISP parameter determine when the deferred generation is rolled in:

DISP=(NEW,CATLG)

Rolls in the GDS at the end of the job step.

DISP=(NEW,KEEP)

Leaves the data set in a deferred roll-in state at the end of the job step.

DISP=(NEW,PASS)

Leaves the data set in a deferred roll-in state. If PASS is specified on the last step of the job, or if the passed data set is not received by the end of the job, the generation data set is deleted.

DISP=(NEW,DELETE)

Deletes the data set at the end of the job step.

A GDS can be left in a deferred roll-in state because of a job or system failure, or if `DISP=(,KEEP)` was specified instead of `DISP=(,CATLG)` in the JCL. To roll in the GDS manually, use the IDCAMS `ALTER ROLLIN` command or the JCL `DISP=(NEW,CATLG)` parameter. Do not use `DISP=(OLD,CATLG)` to roll in a GDS, as it results in a JCL error.

Note: When a GDS is left in the deferred roll-in status, any subsequent job that creates a new GDS with the same GxxxxVyy number overwrites the deferred GDS and its associated space on the volume.

Allocating Compressed Data

To conserve the use of DASD space without compromising I/O performance, allocate sequential and key sequenced data sets in the compressed format. Under this format, more records can fit into a control interval thus reducing the space required for the data sets. In addition, the compression algorithm makes better use of all the systems and devices, consequently improves overall processing rates.

Compressing of sequential and key sequenced data sets does not require application modifications. However, it does require the use of DASD attached to a 9340 or 3990 control unit accessible via ESCON channels.

This section describes the data sets that are eligible for compression and how to request compression for these data sets. For detailed information data set compression, see *DFSMS/MVS V1R2 Using Data Sets*. For information about the hardware and software requirements, see *MVS/ESA SML: Implementing System-Managed Storage*.

Selecting Compression Candidates

Only the following system-managed data sets are eligible for compression:

- Extended sequential data sets
- VSAM key sequenced data sets, not including catalogs, other system data sets, and temporary data sets.

There are two major advantages of compressing sequential and key sequenced data sets. Compressing sequential data sets improves the batch window processing, and compressing key sequenced data sets provides CICS with transparent compression of the user data it manages.

Allocating and Compressing the Data Sets

To allocate and compress sequential or key sequenced data sets, you need to specify the following data class attributes:

- Data Set Name Type = EXTENDED REQUIRED
- Compaction = YES

For key sequenced data sets, you can specify `RECORG = KS` with `EXTENDED REQUIRED`. Specifying the extended format for compression does not stripe the data set across multiple volumes, and the data transfer rate is not affected. Also, you can use the data class `ADDITIONAL VOLUME AMOUNT` attribute to obtain a secondary allocation amount when extending the key sequenced data set to a new volume.

| You can use ISMF to allocate data sets in the compressed format. With ISMF, you
| can also display useful information about the compressed format data sets. In the
| ISMF Data Set Application, the Data Set List panel indicates whether a data set is
| in the compressed format. In addition, the % User Data Reduction column shows
| the percent of space saved by compressing the data set.

| For more information on how to specify the compression attributes and display
| information about compressed data sets, see *DFSMS/MVS V1R2 DFSMSdfp*
| *Storage Administration Reference*.

| You can also generate reports to help assess the space savings and monitor the
| amount of I/O processing performed. For information on how to collect the informa-
| tion and create the reports, see Chapter 9, “Monitoring and Reporting for Storage
| Management” on page 159.

| Allocating data sets in the compressed format requires the use of DASD attached
| to ESCON channels. Therefore, the storage groups you set up for these data sets
| must have ESCON connectivity. For information on planning and setting up
| storage groups, see *MVS/ESA SML: Managing Storage Groups*.

Chapter 4. Managing Space Use

To benefit most from centralized storage management, automate as many space management tasks as possible. In the DFSMS environment, you use data class and management class to automate the deletion and migration of data, and the release of allocated but unused space.

This chapter describes how to establish standards for managing space and to make more efficient use of your DASD. It describes the following tasks:

- Establishing space management standards
- Deleting data
- Releasing unused space
- Deallocating data sets at close time
- Migrating infrequently referenced data

Establishing Space Management Standards

On system-managed volumes, you use data class or management class to manage allocated DASD space. This section explains the roles of data class and management class in automating space management.

For sample classes, descriptions, and ACS routines, see *MVS/ESA SML: Implementing System-Managed Storage*.

Managing Space with Data Class

The two data class space management attributes that users can specify are retention period (RETPD) and expiration date (EXPDT), which tell SMS how long to keep the data before deletion. The values for these attributes are set at data set creation time, and are changeable by using the IDCAMS ALTER command.

Users can override the data class values with the JCL RETPD and EXPDT keywords for data sets, or they can use the OSREQ STORE RETPD macro for objects. These are also specified at data set creation time.

Users can help you manage DASD space by defining the data class space management attributes. You can use management class to override their values if they define unreasonable limits for data retention.

Managing Space with Management Class

Management class allows you to control storage management policies for system-managed DASD data at a data set level across the installation. You can set up an installation-defined, default management class to ensure that all system-managed data conforms to the established space management policies. For tape data sets, management class can override the JCL RETPD and EXPDT specifications.

Groups of data with common management requirements that reside on different volumes can share management classes. For example, a management class for LIST data sets could:

- Expire data sets that have not been referenced for five days or more
- Prevent DFSMShsm migration
- Prevent DFSMShsm backup
- Release allocated but unused space

With this management class, LIST data sets become candidates for deletion when they have not been accessed for five days.

You can change this management class definition later to expire data sets that have not been referenced for 10 days or more. This new definition would apply when you activate SMS with the new configuration.

You can also define management classes for groups of objects (object collections) with common management requirements. OAM uses management class to control the retention, backup, and class transition of objects in an object storage hierarchy.

Use management class to limit the retention period or expiration date that users can assign in data class definitions when they allocate data sets or store objects. Too, you can use management class to manage the expiration of aggregate backup versions.

Management Class Space Management Attributes

The values you assign to these attributes determine how DFSMShsm does its space management tasks. These include expiration, deletion, migration and partial release of allocated but unused space for system-managed DASD data sets, processing system-managed generation data groups, and whether it performs automatic migration. You can also select how OAM manages expiration and class transition of objects, and how DFSMShsm is to manage aggregate backup versions.

The management class space attributes are:

- Data Sets
 - Retention/Expiration
 - Release
 - Migration
 - Generation Data Group
- Objects
 - Expiration
 - Retention
 - Transition
- Aggregates
 - Number of versions
 - Retention of versions

For detailed information on these space management attributes, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

For information on management class backup attributes, see “Using Management Class Backup Attributes” on page 97.

Planning Management Classes

Use the service-level agreement for reference when you plan management classes for the installation. Because your goal is to make optimal use of DASD space, follow these guidelines when you plan management classes:

- Limit the total number of management classes that you define for data sets and for objects. This helps reduce maintenance.
- Do not combine data sets and objects into the same management classes.
- Identify groups of data that are good candidates for early deletion, migration, or object transition and that do not require backup.
- Identify candidate data for extra backup, delayed migration, or transition. Data sets or objects requiring backup need a management class that specifies AUTO BACKUP. Objects requiring **any** class transition require a management class that uses values for either the TIME SINCE CREATION, TIME SINCE LAST USE, or PERIODIC class transition attributes.
- Group objects that have the same management requirements. Define a management class for each of these groups of objects, because they can be stored together in a collection.
- Identify those objects that are moved down the storage hierarchy to slower and less expensive media when their use is infrequent, and promoted to DASD space in the hierarchy when their use is required.
- Define management classes for objects that will remain in a storage hierarchy, to allow OAM to move these objects to mountable optical or tape storage as early as possible. This makes optimal use of the DASD space in the storage hierarchy.

If you maintain data in a distributed environment, we recommend that you work with the users in that environment when planning management classes for this group of data.

Defining Management Classes

Use the ISMF Management Class Application to define management classes to SMS. When users allocate data sets, they can either explicitly assign a management class on their JCL, or use ACS routines to assign a management class. You use the ACS routines to determine if the user-specified management class, an ACS-derived management class, or the RACF default management class should be assigned. *MVS/ESA SML: Managing Storage Groups* discusses how to choose the appropriate method for assigning the SMS classes.

The management class name remains associated with the data set until it is deleted, and not the individual management class attributes. You can use ISMF to change management class definitions, and the new values will apply to the associated data sets at the next DFSMSHsm space management cycle.

You can use the ISMF Management Class Application to define object management classes to SMS. When users store objects, they can explicitly specify a management class on their OSREQ STORE macro. Use the default management class for the collection to which the objects belong, or use the ACS routines to determine if the user-specified management class should be assigned.

ACS routines are also called to assign a management class to an object as a result of a class transition during OAM storage management, or when users explicitly

select a management class change by specifying a new management class, using the OSREQ CHANGE macro in their application program. The name of the object and its associated object collection, and the object's management and storage class, are available to the ACS routines as input variables. Use the ACS routines to determine whether the original, or the new, SMS classes are to be assigned to the object.

The management class name remains associated with the object until it is changed or the object is deleted. You can use ISMF to change management class definitions, and the new values will apply to the associated objects at the next OAM storage management cycle.

Defining a Default Management Class: When DFSMSHsm processes system-managed data that does not have a management class, it uses the default management class defined in the active SMS configuration. You define this default management class using the ISMF Management Class Application and supply the name of the default management class when you define the SCDS base configuration.

DFSMSHsm determines the space management and availability requirements for system-managed data sets from the attributes in the assigned management class. If no management class is assigned, DFSMSHsm uses the default management class you defined. If you did not define a default management class, the system uses the DFSMSHsm management class defaults. These are the same as the ISMF-supplied defaults.

If you do not use a default management class, check the DFSMSHsm defaults to ensure that they can be used for data sets that do not have an associated management class.

If an object is stored and no management class is assigned on the OSREQ STORE macro, the management class of the object's associated collection is used. This management class is assigned to the object collection as a default by ACS routines when the first object is stored in that collection.

Deleting Management Classes: Before you delete a management class from the SMS configuration, ensure that the class is not assigned to data on system-managed volumes, or to migrated and backed up data. DFSMSHsm automatic processing bypasses a data set that has a management class not defined in the active SMS configuration. DFSMSHsm recall and recovery of the data set fails unless the old management class is overridden by the ACS routines.

If you delete a management class assigned to an object, the management class assigned by ACS routines during the storing of an object is determined by the collection to which the object will belong.

Deleting Data

Data in the DFSMS environment can be deleted automatically or manually. This section explains how you can specify automatic deletions or issue commands to delete data.

Automatic Deletion

Data can be deleted automatically if you use the data class or management class RETPD or EXPDT attributes. These indicate when DFSMSHsm is to expire and delete data. In the management class definition, you can set up the retention period as the number of days since the data set was created, the number of days since the data set was last referenced, or you can give an explicit date.

We recommend using the management class RETENTION LIMIT attribute to limit the user-specified or data class RETPD or EXPDT values. If you use zero for RETENTION LIMIT, values for RETPD and EXPDT are ignored.

Then, to enforce installation retention period and expiration date, use the EXPIRATION attributes. The values for the EXPIRE AFTER DAYS NON USAGE and EXPIRE AFTER DATE/DAYS are also used when RETPD or EXPDT is not selected by the user or data class. Figure 46 shows how to enforce the installation retention period and expiration date with the EXPIRATION attributes.

```
MANAGEMENT CLASS DEFINE                                     Page 1 of 5
COMMAND ==>
SCDS NAME:          STGADMIN.SCDS1
MANAGEMENT CLASS NAME: DELETE

TO DEFINE MANAGEMENT CLASS, SPECIFY:

DESCRIPTION ==> Delete data sets 45 days after create date,
               ==> or at least 40 days since last reference or use.

EXPIRATION ATTRIBUTES
EXPIRE AFTER DAYS NON-USAGE ==> 40          (1 to 9999 or NOLIMIT)
EXPIRE AFTER DATE/DAYS      ==> 45          (0 to 9999, yyyy/mm/dd
                                           or NOLIMIT)

RETENTION LIMIT           ==> 0           (0 to 9999 or NOLIMIT)

USE ENTER TO PERFORM VERIFICATION; USE DOWN
COMMAND TO VIEW NEXT PANEL; USE HELP COMMAND FOR HELP;
USE ENDCOMMAND TO SAVE AND EXIT; CANCEL TO EXIT.
```

Figure 46. Enforcing Installation RETPD and EXPDT Values

The management class RETENTION LIMIT and EXPIRATION attributes do not require the users to alter their JCL or IDCAMS commands. However, the values you place in the EXPIRATION attributes need to reflect the service levels you agree to when you negotiate the SLA. For example, you might agree to some expiration criteria for different types of data, as Figure 47 shows:

Figure 47. Determining Expiration Criteria

Data Type	Expire After
Listing	5 days
Batch	800 days
TSO	400 days
TSO logon/ISPF	400 days
DB2 table spaces and indexes	1200 days

Some data types are not good candidates for DFSMSHsm automatic deletion. For example, when DFSMSHsm automatically deletes database data (such as DB2 production databases, DB2 active logs, or IMS online log data sets), the database catalogs are not updated automatically. This results in inconsistencies in the database catalogs.

To avoid database catalog inconsistency, we recommend you use management classes that use NOLIMIT for the EXPIRATION and RETENTION LIMIT attributes. This allows the database managers to control the deletion instead of DFSMSHsm. See *MVS/ESA SML: Implementing System-Managed Storage* for database data management classes.

You can also use the MAXIMUM RETENTION PERIOD attribute to override the JCL EXPDT when redirecting tape data set allocations to system-managed DASD storage. This interception is part of the *tape mount management* technique that ensures efficient tape data management. Overriding the JCL EXPDT allows you to implement tape mount management without modifying existing JCL.

To control the expiration or retention of data sets on tape volumes, you can use management class attributes or the vital record specifications defined through DFSMSrmm (a functional component of DFSMS/MVS). DFSMSrmm uses these expiration or retention values to determine when it can release a volume. The vital record specification retention definition overrides the management class expiration or retention specifications.

For information about DFSMSrmm and how to define vital record specifications, see *DFSMS/MVS V1R2 DFSMSrmm Guide and Reference*.

Objects can be deleted automatically if you specify the management class expiration and retention limit attributes. These indicate when OAM is to expire and delete the objects. OAM deletes objects during a storage management cycle with the approval of the auto-delete installation-wide exit.

The management class expiration and retention limit attributes also control the expiration and retention of objects on tape volumes. The data class EXPDT and RETPD attributes can be used to control the expiration of the tape volumes. The values you specify for these attributes override the values set by your tape management system so that the object tape volumes do not expire prematurely. For information on how to specify space management attributes, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*. For detailed information on controlling the retention and expiration of objects, see *DFSMS/MVS V1R2 OAM Application Programmer's Reference*.

Expired Data

DFSMSHsm can delete expired system-managed data, including migrated data and old GDG generations. If RETPD or EXPDT attributes are not specified by the user, or derived from a data class, scratch and uncatalog action is determined by the retention criteria specified in the data's management class definition.

To have DFSMSHsm automatically delete system-managed data sets based on user-specified, data class retention periods and expiration dates, specify the SCRATCH parameter on the SETSYS EXPIREDDATASETS command. DFSMSHsm then automatically scratches data sets that have reached their expiration dates when it performs volume space management and migration cleanup.

The OAM Storage Management Component ensures the automatic deletion of objects that have reached their expiration dates. On expiration, both the primary copy and backup copy of the object are deleted.

The object deletion method applied depends on the type of storage media used. For objects residing on rewritable media, the objects can be physically deleted. However, because write-once-read-many (WORM) optical storage is not rewritable, objects residing on WORM optical storage are not physically deleted upon reaching their specified expiration dates. Only their entries in the object directory are deleted.

Temporary, List and Utility Data Sets

In the DFSMS environment, the deletion of temporary system-managed data sets occurs during automatic space and availability management. How the data sets are deleted depends on whether global resource serialization (GRS) is activated, and the number of processors that are sharing the data sets.

If GRS is activated, then deletion scratches temporary data sets that are not in use. If GRS is not activated or installed in a single processor environment, deletion scratches temporary data sets that are at least one day old. If GRS is not activated or installed in a multiple processor environment, deletion scratches temporary data sets that are at least two days old.

We recommend that you use GRS. Using GRS ensures that excess temporary data sets do not take up the space on your primary volumes. In addition, it ensures data set integrity in a multiple processor environment.

To use GRS, you need to specify the USERDATASETSERIALIZATION parameter on the SETSYS command. For more information on MVS GRS, see "Data Set Serialization" on page 73 and *MVS/ESA Global Resource Serialization*. For more information on controlling data set serialization, see *DFSMS/MVS V1R2 Using Advanced Services for Data Sets*.

One way to control the deletion of list type system-managed data sets is to use management class. For example, you can define a management class for data sets with names ending in LIST, LINKLIST, and OUTLIST.

To scratch list data sets on non-system-managed primary volumes, use the DFSMSHsm SETSYS SCRATCHFREQUENCY parameter. For information on using the SETSYS SCRATCHFREQUENCY parameter, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

You can also use management class to control the deletion of utility data sets. For example, you can assign the INTERIM management class to utility data sets, as discussed in *MVS/ESA SML: Implementing System-Managed Storage*. The INTERIM management class specifies that utility data sets be deleted three days after creation.

Another method of deleting utility data sets is to automatically invoke DFSMSdss using the DUMP function with DELETE and data set filtering. You can also use DFSMSdss to delete your own installation-defined temporary data sets. For more information on using DFSMSdss to delete data sets, see *DFSMS/MVS V1R2 DFSMSdss Storage Administration Guide*.

Manual Deletion

Management class is the recommended way to handle overriding expiration dates for deletions. However, you can delete data manually between space management cycles if you need to delete data that has not expired or free your volumes of obsolete and uncataloged data.

Note: Exercise caution when you delete data shared between remote sites.

Data That Has Not Expired

In certain environments, you may want to manually delete an unexpired data set. For example, you may want to do this in an environment using TMM, where DASD data sets have expiration dates because they were converted from tape. Two actions you can take to control the deletion of unexpired system-managed DASD data sets are:

- Specify OVRD_EXPDT(YES) in the IGDSMSxx member of SYS1.PARMLIB

OVRD_EXPDT(YES) should only be used when management class cannot be used. OVRD_EXPDT(YES) ensures that deletion requests through JCL DD statements are honored, regardless of any expiration date specified. By allowing the deletions via JCL DD statements, you can eliminate the extensive JCL changes required if you were to use IDCAMS or IEHPROGM to delete the tape data sets directed to DASD. Specifying OVRD_EXPDT(YES) enables you to take maximum advantage of the tape mount management strategy.

With OVRD_EXPDT(YES), you can also delete unexpired data sets using any of the following methods:

- IEHPROGM SCRATCH with or without the PURGE parameter specified
- SVC 99 with a disposition of DELETE

Notes:

- Exercise caution when you specify OVRD_EXPDT(YES), as it is a global override and accidental deletion of unexpired data may occur as a result.
- OVRD_EXPDT(YES) should not be used if you use explicit expiration dates with system-managed DASD data sets because you will affect these data sets also.

- Specify OVRD_EXPDT(NO) in the IGDSMSxx member of SYS1.PARMLIB

OVRD_EXPDT(NO) prevents unexpired system-managed DASD data sets from being accidentally deleted by making sure the expiration dates are never overridden unless specifically requested. This is the existing practice, where JCL DD deletion requests are not possible, and the use of IDCAMS and IEHPROGM with PURGE are still applicable.

To delete a migrated system-managed data set with an expiration date that has not passed, use the DFSMSShsm DELETE command with the PURGE keyword. This will delete the data set without recalling it before deletion. Your users can use the DFSMSShsm HDELETE command.

If you want to purge an unexpired backup copy of a system-managed data set, issue the DFSMSShsm BDELETE command. Your users can issue the DFSMSShsm HBDELETE command. For more information on the use of these commands, see *DFSMS/MVS V1R2 DFSMSShsm Managing Your Own Data*.

To remove an object from the object storage hierarchy or from the backup storage group, use the OSREQ DELETE macro. The deletion request removes an object's directory information and frees all reusable resources allocated to the object. For more information on the OSREQ DELETE macro, see *DFSMS/MVS V1R2 OAM Application Programmer's Reference*.

Obsolete and Unnecessary Data

Data is obsolete when it has exceeded its expiration dates and is no longer needed. Some examples are old masters, listings, and permanent work files. To select obsolete data for deletion using DFSMSdss, issue the DUMP command and the DELETE parameter, and force OUTDDNAME to DUMMY.

Excess backup data versions are examples of unnecessary data. DFSMSShsm normally deletes excess backup versions when it creates a new backup version and when the EXPIREBV command is processed. This is determined by the VERSIONS parameter of the SETSYS command or the BACKUP VERSIONS parameters on the management class definition. To explicitly delete versions at other times, use the storage administrator command, BDELETE. Users can delete backup data sets with the user command HBDELETE. The HBDELETE command can also be issued with ISMF.

Uncataloged Data on System-Managed Volumes

Under certain circumstances, uncataloged DASD data sets can end up on system-managed volumes. These are data sets that have no BCS or NVR in the VVDS, and are not duplicates of cataloged data.

We recommend deleting these uncataloged data sets using the IDCAMS DELETE command. Figure 48 on page 68 shows a sample job step calling IDCAMS to delete an uncataloged dataset in the DEPTM86.FINANCE.IFACE.COPY generation data group. This job step also updates the VTOC, VTOC index, and, if an NVR existed, the VVDS. If you want to delete an uncataloged VSAM data set, specify VVR instead of NVR.

For more information on using the IDCAMS DELETE command, see *DFSMS/MVS V1R2 Access Method Services for ICF*.

```

//DELUNCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DASD DD VOL=SER=PRM901,UNIT=3390,DISP=OLD
//SYSIN DD *
/* DELETE UNCATALOGED SYSTEM-MANAGED DATA SET */
DELETE 'DEPTM86.FINANCE.IFACE.COPY.G0202V00' NVR FILE(DASD)
/*

```

Figure 48. Deleting Uncataloged Data With IDCAMS

Another way to delete uncataloged data sets from system-managed volumes is to use the DELETE command in the ISMF LIST function. You can use volume serial numbers to list data sets on these volumes, then type the line command, DELETE, next to the uncataloged data set entries. For more information on using the ISMF LIST function, see *DFSMS/MVS V1R2 Using ISMF*.

Releasing Unused DASD Space

By releasing allocated but unused space, you can reclaim the wasted space caused by overallocation. This section explains how to specify automatic release during space management cycles and at data set close time, and how to manually release allocated but unused space between the cycles.

Automatic Release

Use management class to indicate that a data set is eligible for automatic release of allocated but unused space. You can specify the management class PARTIAL RELEASE attribute as follows:

- YI** Unconditionally release space when the data set is closed after writing.
- CI** Conditionally release whole, unused extents when the data set is closed, only if secondary allocation was specified when the data set was created.
- Y** Unconditionally release space during DFSMSHsm space management.
- C** Conditionally release whole, unused extents during DFSMSHsm space management only if secondary allocation was specified when the data set was created.

You can specify these options using ISMF. Figure 49 on page 69 shows a sample Management Class Define panel and the PARTIAL RELEASE attribute.

```

                                MANAGEMENT CLASS DEFINE                Page 2 of 5
COMMAND ===>

SCDS NAME:                      STGADMIN.SCDS1
MANAGEMENT CLASS NAME: PARTREL

TO DEFINE MANAGEMENT CLASS, SPECIFY:

PARTIAL RELEASE                  ===> CI      (Y, C, YI, CI or N)

MIGRATION ATTRIBUTES
PRIMARY DAYS NON-USAGE          ===> 15      (0 to 9999 or blank)
LEVEL 1 DAYS NON-USAGE         ===> 60      (0 to 9999, NOLIMIT or blank)
COMMAND OR AUTO MIGRATE        ===> BOTH    (BOTH, COMMAND or NONE)

GDG MANAGEMENT ATTRIBUTES
# GDG ELEMENTS ON PRIMARY       ===> 1      (0 to 255 or blank)
ROLLED-OFF GDS ACTION          ===> EXPIRE  (MIGRATE, EXPIRE or blank)

USE ENTER TO PERFORM VERIFICATION;
USE UP/DOWN COMMAND TO VIEW OTHER PANELS;
USE HELP COMMAND FOR HELP;
USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

```

Figure 49. Specifying Partial Release Conditions

The PARTIAL RELEASE attribute applies only to system-managed physical sequential, extended sequential, partitioned, and VSAM extended format data sets. PARTIAL RELEASE does not apply to data sets allocated with guaranteed space or during DFSMSHsm interval migration.

For non-VSAM data sets, specifying unconditional release (YI or Y) causes the release of unused space, even if it has no secondary allocation. This can create space problems for users who do not specify secondary extents in their non-VSAM space allocation. Once the primary extent is used, any attempt to extend the non-VSAM data set fails.

For example, a non-VSAM data set is created with a primary allocation of five cylinders, a secondary allocation of zero, and only three of those cylinders are used. If you specified unconditional partial release, then two of those cylinders are released because they are not used. Subsequently, when you need to extend beyond the three cylinders, the allocation would fail because there was no secondary allocation quantity specified at data set creation time.

To avoid this situation with non-VSAM data, we recommend that you specify CI for the PARTIAL RELEASE attribute, as Figure 49 shows. When you do this, the unused space is released only if secondary space is allocated.

With VSAM data sets, this situation is different. When you need to extend beyond three cylinders, the system remembers that five cylinders were initially allocated. So, it would allocate two additional cylinders to regain the former quantity.

With VSAM data sets, conditional release (CI or C) will operate differently than with non-VSAM. If you specify conditional release for VSAM data sets, then unused space is released even if no secondary allocation quantity was specified. In other words, for VSAM data sets, CI will operate as if YI was specified and C will operate as if Y was specified.

For more information on specifying automatic release and selecting a release condition, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Manual Release

For non-VSAM data sets, users can reduce wasted space by using the JCL parameter, RLSE. This frees unused space when the data set is closed after writing. Figure 50 shows an example of how to use the RLSE parameter.

```
//SYSUT2 DD DSN=DEPTM86.PARTS.DATA,  
//          DISP=(,CATLG,DELETE),RECFM=VB,LRECL=132,  
//          SPACE=(132,(5000,50),RLSE)
```

Figure 50. Coding RLSE in JCL

Do **not** specify RLSE for the following data sets:

- Data sets that are closed and then reopened for output.
Unused space is released during CLOSE, so there might not be enough space to write into after the second OPEN (unless sufficient secondary allocation is available).
- Sequential data sets that need to be preallocated for future updates (unless sufficient secondary allocation is available).
- Partitioned data sets that are not managed by DFSMSShsm MAXEXTENTS processing, which compresses the data set and consolidates extents. (For a discussion on MAXEXTENTS, see “Compressing Data Sets and Consolidating Extents” on page 80.) Significant expansion of these partitioned data sets might result in the data set using the maximum number of extents available.

Users can also reduce wasted space by using the TSO ALLOCATE parameter, RELEASE. This frees unused space when the data set is closed after writing. Figure 51 shows how to use the RELEASE parameter on a TSO ALLOCATE statement.

```
ALLOCATE DATASET('DEPTM86.PARTS.INPUT.DATA') SPACE(500 50) RELEASE
```

Figure 51. Coding RELEASE on a TSO ALLOC Statement

To release allocated but unused space from sequential, partitioned, and extended sequential data sets, you can use the DFSMSdss RELEASE command. When you release space from an extended sequential data set, issue RELEASE for all of the volumes on which the data set resides.

Deallocating Data Sets at Close Time

The JCL parameter `FREE=CLOSE` deallocates a non-VSAM data set when it is closed instead of at step termination, when deallocation would normally occur. No equivalent option exists in TSO ALLOC. The user should, however, use TSO FREE when the data set is no longer needed. Note that for VSAM data sets, there is no equivalent for `FREE=CLOSE`.

The advantage of using `FREE=CLOSE` is that data sets normally deleted at step termination (those with a normal termination disposition of `DELETE`, or temporary data sets) can be deleted earlier. If you have many long-running job steps with such data sets, or if the data sets are numerous and large, this option is especially useful. Do not deallocate a data set at close time if the data set is closed and then reopened within the same step.

Named data sets that are required only for the duration of the job but not needed after job termination need to be deleted in the last job step (by specifying the JCL DISP parameter `DELETE`).

Migrating Infrequently Referenced Data

One way to use your DASD volumes efficiently is to move infrequently referenced data to less expensive storage media. This process is called data migration. This section explains the data migration process, and how you can allow DFSMSHsm to consolidate data during migration.

The Data Migration Process

Data migration consists of moving cataloged data from user-accessible DASD volumes to DFSMSHsm-owned migration volumes. This takes place in a three-level hierarchy.

Migration Levels

The three-level migration hierarchy is as follows:

Primary Storage

DASD volumes containing currently referenced data that users can access without any intervention by DFSMSHsm.

Migration Level 1 Storage

DFSMSHsm-owned volumes containing data moved from primary volumes. Migration level 1 volumes are usually DASD volumes.

Migration Level 2 Storage

DFSMSHsm-owned volumes containing data moved from primary or migration level 1 storage. Migration level 2 volumes are usually magnetic tape volumes.

In this hierarchy, frequently accessed data is stored on primary volumes; data that is infrequently accessed is stored on DFSMSHsm-owned migration volumes. When a user references migrated data, DFSMSHsm automatically recalls the data to a primary volume. Figure 52 on page 72 illustrates DFSMSHsm's automatic migration and recall functions.

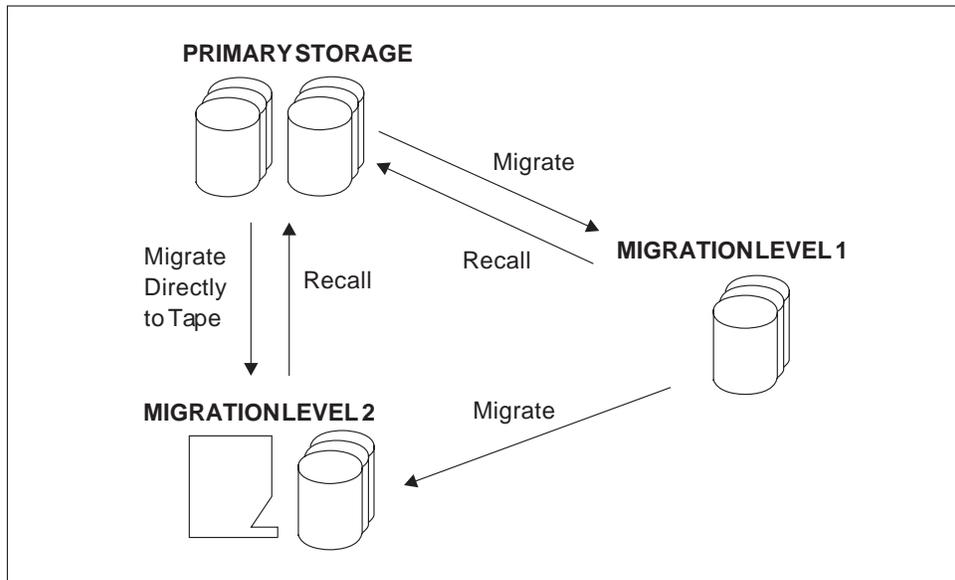


Figure 52. Migrating and Recalling Data using DFSMSHsm

An object moves to another level of storage hierarchy during an *object class transition*, which is performed by the object access method (OAM). Before the object is moved, it is assigned a new storage class and management class, which change the placement of the object in the hierarchy and determine the object's new transition date. For more information, see "Object Class Transition" on page 77.

Threshold Processing

Data set migration activity is determined by the high and low occupancy thresholds of storage group volumes. You specify threshold values in the storage group definition; these values apply to each of the volumes in the storage group. You need to document the expected amount of free space in each storage group in the service-level agreements. For information on service-level agreements, see *MVS/ESA SML: Leading a Storage Administration Group*.

The **low** threshold value is used during the DFSMSHsm automatic volume space management cycle. If the amount of allocated space exceeds the low threshold, DFSMSHsm first attempts to free space by deleting eligible data and releasing unused space from data sets. DFSMSHsm then migrates data sets until the threshold is met.

The **high** threshold value is used if you use the DFSMSHsm SETSYS command to request interval migration. In interval migration, DFSMSHsm periodically compares the percentage of allocated space on the primary volume with its high threshold. If the high threshold has been met or exceeded, DFSMSHsm deletes or migrates data sets until occupancy meets the low threshold. The high threshold is also used by SMS during new data set allocation to balance occupancy across a storage group. SMS prefers volumes below high threshold over volumes which exceeds the high threshold. This helps minimize the number of times DFSMSHsm has to perform interval migration.

Note: Data set allocation does not fail even if the high threshold is exceeded.

Space Management Cycles

You use the storage group AUTO MIGRATE attribute to specify whether DFSMSShsm is to perform automatic space management processing.

During automatic primary space management, eligible data sets are deleted or moved to migration level 1 or migration level 2 volumes, and unused space is released. During automatic secondary space management, CDSs, SDSPs, and migration volumes are cleaned up, and eligible data sets are moved from migration level 1 to migration level 2 volumes.

Daily space management of volumes is normally sufficient. However, you can request interval migration to free space on primary volumes between space management cycles. Interval migration is especially useful during periods of heavy allocation, because interval processing checks the volumes hourly.

If you want to suppress interval migration without turning off primary storage space management, you can set the AUTO MIGRATE attribute to P (Primary Storage Management). Interval migration will not be performed even if SETSYS INTERVALMIGRATION is specified.

You can define automatic secondary space management cycles independent of automatic primary space management cycles. This allows you to perform automatic primary space management at almost any time instead of having to wait for the automatic secondary space management cycle to complete.

DFSMSShsm also offers input channel load balancing. With this function, migration, backup, and dump loads are spread across channels to minimize channel contention for primary volumes during automatic cycles. For more information, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Data Set Serialization

To ensure that data sets are not altered during migration or backup, specify the DFHSM DATASET SERIALIZATION or the USER DATASET SERIALIZATION parameter. Both parameters provide data integrity.

Using DFHSM DATASET SERIALIZATION causes DFSMSShsm to prevent other processing units from accessing the volume used during migration. Because of possible deadlock situations, this serialization method does not support multivolume non-VSAM data sets.

Using USER DATASET SERIALIZATION makes sure that no other processing unit can update the data set while DFSMSShsm is performing its operations. Before you specify USER DATASET SERIALIZATION, you must have support for system-wide data set serialization. We recommend you use MVS Global Resource Serialization (GRS) to provide system-wide support, especially if data is shared across multiple processing units.

For more information on controlling data set serialization, see *DFSMS/MVS V1R2 Using Advanced Services for Data Sets*. For information on MVS GRS, see *MVS/ESA Global Resource Serialization*.

Automatic Migration

The management class migration and class transition attributes control automatic migration and class transition for system-managed data. The migration attributes determine when DFSMSHsm migrates data to migration level 1 volumes and from migration level 1 volumes to migration level 2. Figure 53 shows the migration attributes you can define for data sets and GDGs, using ISMF.

```
MANAGEMENT CLASS DEFINE                                     Page 3 of 5
COMMAND ==>
SCDS NAME:          STGADMIN.SCDS1
MANAGEMENT CLASS NAME: EARLYMIG

TO DEFINE MANAGEMENT CLASS, SPECIFY:

PARTIAL RELEASE          ==> CI      (Y, C, YI, CI or N)

MIGRATION ATTRIBUTES
PRIMARY DAYS NON-USAGE   ==> 2      (0 to 9999 or blank)
LEVEL 1 DAYS NON-USAGE   ==> 0      (0 to 9999, NOLIMIT or blank)
COMMAND OR AUTO MIGRATE  ==> BOTH   (BOTH, COMMAND or NONE)

GDG MANAGEMENT ATTRIBUTES
# GDG ELEMENTS ON PRIMARY ==> 1      (0 to 255 or blank)
ROLLED-OFF GDS ACTION    ==> EXPIRE (MIGRATE, EXPIRE or blank)

USE ENTER TO PERFORM VERIFICATION;
USE UP/DOWN COMMAND TO VIEW OTHER PANELS;
USE HELP COMMAND FOR HELP;
USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.
```

Figure 53. Specifying Management Class Migration Attributes

To allow automatic migration of system-managed data, the data must be assigned a storage group that specifies AUTOMIGRATE = Y, I, or P. The data must also be assigned a management class that specifies the conditions of the migration or class transition.

You can also use the ARCHMIG macro to automate migration from a user program. For more information, see *DFSMS/MVS V1R2 DFSMSHsm Managing Your Own Data*.

DFSMSHsm can automatically migrate eligible data on a daily basis. You can specify up to 15 volume migration tasks to be carried out concurrently. Allowing multiple migration tasks reduces the time required to perform automatic volume space management functions.

When you determine the number of tasks, consider the effect multiple migrations might have, such as resource contentions for tape drives and SDSPs. Use the MAXMIGRATIONTASKS parameter of the SETSYS command to control the maximum number of automatic volume migration tasks you want to run concurrently. For further information on multi-task volume migration, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Data Set Migration

For data sets on storage group volumes, migration is based on volume threshold and the number of days since a data set was last referenced (PRIMARY DAYS NON USAGE).

The PRIMARY DAYS NON USAGE values you specify in the management classes for your installation need to reflect the service levels you agree to when you negotiate the service-level agreement. For example, you could agree to set up the migration criteria for the different types of data used at your installation, as Figure 54 shows:

Figure 54. Determining Primary Storage Migration Criteria

Data Type	Migrate After
Batch	7 days
TSO	15 days
TSO logon/ISPF	30 days

You can also specify that DFSMSHsm never migrates certain data sets like high activity ISPF libraries, data shared between sites in a distributed environment, and database data requiring high availability. Figure 55 shows the database data that are not to be migrated.

Figure 55. Identifying Database Data That Does Not Need Automatic Migration

DB2	IMS	CICS
Production databases	Online log data sets	Data Sets in local shared resource pool
Active Logs		
Bootstrap data sets		
System libraries		
Temporary work files		
Catalogs		
Directories		

For data sets on migration level 1 volumes, movement to migration level 2 is based on the total days of inactivity for the data set. You can specify migration criteria for different types of data, based on the likelihood of the data set being recalled. Figure 56 shows sample migration level 1 migration criteria.

Figure 56. Determining Migration Level 1 Migration Criteria

Data Type	Migrate After
Batch	45 days
TSO	90 days
TSO logon/ISPF	180 days

You can also specify that data sets migrate directly to migration level 2 tapes by specifying a value of 0 for the LEVEL 1 DAYS NON-USAGE attribute. For example, you could use this attribute to migrate the following data types directly to migration level 2 volumes if they are not referenced for two consecutive days:

- Very large data sets
- Low-use batch data sets
- Generation data sets
- DB2 archive logs
- DB2 and IMS image copies

Figure 53 on page 74 shows how to implement this migration technique.

Note: You can only migrate the database data sets when they are closed.

If migration level 2 volumes are tape volumes, you might want to use management class to selectively stop migration of some system-managed data sets from migration level 1 to migration level 2. If you know that certain data sets are likely to be recalled, you might save some tape mounts by keeping them on migration level 1 DASD. For example, recalling HFS data sets from tape can decrease the rate of performance because it requires a physical mount. Therefore, if you must migrate HFS data sets, we recommend that you migrate them only to migration level 1 DASD.

Generation Data Group Migration

The Generation Data Group (GDG) attributes control the migration of older elements of a GDG from system-managed volumes. You specify the number of the most recent GDG elements that are to remain on primary volumes. When you run DFSMSHsm automatic primary space management or interval migration, other elements will be eligible for early migration.

To make the most effective use of space, consider retaining only the current generation of a GDG on the primary volume and migrating other versions directly to migration level 2 tape. Migrate the current version on the primary volume only if it is not referenced for two days. Figure 53 on page 74 shows you how to use ISMF to do this. DFSMSHsm prioritizes the migration of generation data sets (GDSs) that satisfy the value specified by the # GDG ELEMENTS ON PRIMARY attribute over other GDGs and non-GDGs.

You can also use management class to control migration or expiration of *rolled off* GDSs. These result when a GDS is cataloged to a GDG that is full, and the oldest generations become rolled off, therefore not associated with a GDG. If you want to delete these generations when they become rolled off, use the SCRATCH attribute when defining the GDG base. If you specify the NOSCRATCH attribute, GDG LIMIT processing results in old generations not being deleted.

The management class ROLLED-OFF GDG ACTION attribute allows you to specify whether DFSMSHsm migrates or expires the undeleted, rolled-off data sets during space management or interval processing. As an installation standard, consider deleting rolled-off generations.

Object Class Transition

Object class transition consists of moving objects within the object storage hierarchy:

- DASD volumes
- Optical volumes inside a library device
- Optical volumes outside a library device
- Tape volumes inside a library device
- Tape volumes outside a library device

In this storage hierarchy, frequently accessed objects reside on DASD volumes, while less frequently accessed objects reside on either optical or tape volumes.

Object class transitions occur during an OSMC storage management cycle. Values specified in the management class object class transition attributes determine when an object class transition occurs. During transition, the object can be assigned a new storage class and management class, which cause the object to move to another level in the storage hierarchy. The new storage class and management class also determine the object's new transition date and/or new expiration criteria.

The object class transition criteria is based on one of three specifications: TIME SINCE CREATION, TIME SINCE LAST USE, and PERIODIC. These attributes cannot be used together. Figure 57 shows the attributes you can specify for object class transition using ISMF.

```
MANAGEMENT CLASS DEFINE                                     Page 4 of 5
COMMAND ===>
SCDS NAME:          SMS.SCDS1.SCDS
MANAGEMENT CLASS NAME: QTRLY1

TO DEFINE MANAGEMENT CLASS, SPECIFY:

OBJECT CLASS TRANSITION CRITERIA
  TIME SINCE CREATION YEARS ===>      (0 to 9999 or blank)
                                MONTHS ===> (0 to 9999 or blank)
                                DAYS  ===> (0 to 9999 or blank)
  TIME SINCE LAST USE YEARS ===>      (0 to 9999 or blank)
                                MONTHS ===> (0 to 9999 or blank)
                                DAYS  ===> (0 to 9999 or blank)

PERIODIC
  MONTHLY  ON DAY  ===>      (1 to 31, FIRST, LAST or blank)
  QUARTERLY ON DAY  ===> 15  (1 to 92, FIRST, LAST or blank)
                                IN MONTH ===> 2  (1 to 3 or blank)
  YEARLY   ON DAY  ===>      (1 to 366, FIRST, LAST or blank)
                                IN MONTH ===>      (1 to 12 or blank)

USE ENTER TO PERFORM VERIFICATION;
USE UP/DOWN COMMAND TO VIEW OTHER PANELS;
USE HELP COMMAND FOR HELP;
USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.
```

Figure 57. Specifying Object Class Transition Attributes

TIME SINCE CREATION defines the time that must elapse since the object creation date before class transition. You specify the value in years, months, days, or a combination of the three. For example, if 7 is entered for YEARS, and 6 for DAYS, then class transition occurs 7 years and 6 days after the date the object was created.

TIME SINCE LAST USE defines the time that must elapse since the last reference date before class transition occurs. You specify the value in years, months, days, or a combination of the three. For example, if 8 is entered for MONTHS, and 12 is specified for DAYS, then class transition occurs 8 months and 12 days after the date the object was last referenced.

PERIODIC specifies that a class transition occurs at a regular time, based on the calendar. Figure 57 on page 77 shows the different time periods you can specify. You can use these attributes separately or combine them (except for MONTHLY ON DAY, which must be used separately). If you combine them, the values specified for the attributes are calculated to obtain an absolute value. For example, Figure 58 shows entries that cause a class transition to occur on the last calendar day of the third month in each quarter.

PERIODIC			
MONTHLY	ON DAY	====>	(1 to 31, FIRST, LAST or blank)
QUARTERLY	ON DAY	====> LAST	(1 to 92, FIRST, LAST or blank)
	IN MONTH	====> 3	(1 to 3 or blank)
YEARLY	ON DAY	====>	(1 to 366, FIRST, LAST or blank)
	IN MONTH	====>	(1 to 12 or blank)

Figure 58. Specifying a Combination of Periodic Attributes

Because class transitions might not be required, you can also specify that a class transition for certain objects should never occur from the DASD volumes in the storage hierarchy (for objects that have a very high frequency of reference, for example). To do this, enter a 0 for all the attributes.

Manual Migration

If you want to force migration of specific data sets between space management cycles, you can issue the HMIGRATE command in ISMF or in TSO. For more information on using the HMIGRATE command, see *DFSMS/MVS V1R2 DFSMSShsm Managing Your Own Data*.

You can cause an object to go through class transition by changing its storage class and management class assignments. An object's initial storage class and management class can be changed manually by executing an OSREQ CHANGE macro in an application to specifically request other class assignments for the object. ACS routines are invoked to validate the change request and assign new classes. For more information on using the OSREQ CHANGE macro, see *DFSMS/MVS V1R2 OAM Application Programmer's Reference*.

Space Management During Migration

DFSMSHsm helps consolidate data on storage volumes during the migration process using features that require less space to store the data that it moves from primary to migration volumes. You can use DFSMSHsm to ensure that data is available to users when they need it, without intervention and, at the same time, use less storage for the data.

DFSMSHsm will not perform the space-saving functions on sequential and key-sequenced data sets that are already compressed. It migrates and backs up the compressed data sets in their compressed formats.

This section describes the following DFSMSHsm features:

- Compacting data
- Compressing data sets and consolidating extents
- Blocking data sets on DFSMSHsm-owned volumes
- Reblocking during recall and recovery
- Storing small data sets

Compacting Data

There are two ways to compact data sets as they are moved to lower storage levels. You can use the DFSMSHsm compaction option, or enable DFSMSHsm to use the improved data recording capability (IDRC) feature of the 3490 cartridge-type device (you have the option of installing IDRC on your 3480 subsystems). You can also combine the two compaction methods to maximize space use on the storage volumes.

Using the compaction option, DFSMSHsm saves space on migration and backup volumes by encoding data sets in a compacted form as it moves data from primary volumes to migration or backup volumes. You can achieve an average byte compaction ratio of up to two to one during migration, even when excess space is removed before migration. This level of compaction makes the use of DASD for migration and backup more effective.

To use the DFSMSHsm data compaction option, specify the COMPACT parameter on the SETSYS command. When DFSMSHsm recalls or recovers a compacted data set, it automatically decodes and expands the data set.

You can compact the data even further using the IDRC feature of the 3490 cartridge-type device. With IDRC, you can compact data whenever it is migrated or backed up to tape, whether it is from a primary volume, DASD migration volume, or from another tape volume. With the 3490E 36-track recording mode, twice the amount of data can be stored on a cartridge. And with the Enhanced Capacity Cartridge System Tape, you can store four times as much data on a cartridge. When the data is recalled, it is automatically decompacted before DFSMSHsm receives the data.

There are two ways you can control IDRC compaction. If you want to use IDRC for a tape library device, specify the data class COMPACTION attribute to request the use of IDRC.

If the allocation is to a tape outside of the library, and you have the 3490 Magnetic Tape Subsystem control unit, you can automatically take advantage of the IDRC feature. DFSMSHsm always uses IDRC to write to 3490 tape devices.

If you have a 3480 tape device with the IDRC feature, use the DFSMSHsm SETSYS TAPEHARDWARECOMPACT parameter to enable DFSMSHsm to request IDRC compaction on eligible 3480 tape cartridges. Figure 59 on page 80 shows how to use the IDRC feature of a 3480 by specifying the SETSYS command parameters.

```
SETSYS TAPEHARDWARECOMPACT
SETSYS MIGUNITNAME(3480X)
```

Figure 59. Using the IDRC Feature of 3480 Tape Devices

The TAPEHARDWARECOMPACT parameter allows you to use IDRC for 3480, and the MIGUNITNAME(3480X) parameter specifies that IDRC is to be used for migrating 3480 tape volumes.

We recommend that you use IDRC to compact data sets for your incremental backup and DFSMSHsm migration level 2 volumes instead of using the DFSMSHsm data set compaction option alone. The compaction achieved with IDRC is greater than that with DFSMSHsm, and can further compact DFSMSHsm migrated data.

For objects that are moved to tape volumes, you can also take advantage of the IDRC feature of the tape cartridge devices to maximize space use. See *DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information on using IDRC for object tape volumes.

Compressing Data Sets and Consolidating Extents

DFSMSHsm generally moves only data, not the entire allocated space, when it migrates and recalls data sets. If a user allocates more space for a data set than the data requires, DFSMSHsm releases the unused space when it recalls the data set. The recall also attempts to reallocate the data set in a single extent. For a PDS, DFSMSHsm releases the unused space that is no longer available because of member updates when it recalls the data set, thus compressing the PDS.

The DFSMSHsm MAXEXTENTS function automatically causes a migrate-and-recall combination when a data set reaches the number of extents you specify on the MAXEXTENTS parameter of the SETSYS command. This combination performs extent consolidation and PDS compression. Therefore, these data sets will extend the next time output processing occurs. When the specified number of extents is again exceeded, DFSMSHsm selects the data set for MAXEXTENTS processing in the next DFSMSHsm space management cycle.

Blocking and Reblocking Data Sets

To make more efficient use of space on DFSMSHsm-owned DASD volumes, you can use DFSMSHsm-defined optimum block size whenever DFSMSHsm performs migration or backup. Using the SETSYS OPTIMUMDASDBLOCKING command causes DFSMSHsm to automatically block data sets in a multiple of 2KB that is determined by the track size of the output device.

You can also reblock data sets to make efficient use of available DASD space. The DFSMSHsm DASD conversion process provides options for reblocking during recall and recovery of blocked, physical sequential, extended sequential, and partitioned data sets.

You can also have DFSMShsm invoke DFSMSdss to do the reblocking. However, DFSMShsm does not request DFSMSdss to reblock a compressed sequential or key sequenced data set when performing recall and recovery.

To initiate the reblocking of data sets during recall and recovery, specify the SETSYS CONVERSION keyword. The parameters are:

- REBLOCKBASE** Reblocks to any target device when the target volume is different from the source volume. This is the SETSYS default and DFSMSdss ignores this parameter.
- REBLOCKTOANY** Reblocks to any DFSMShsm-supported target volume. This is the only parameter used by DFSMSdss.
- REBLOCKTOUNLIKE** Reblocks to any target device when the target volume is different from the source volume. DFSMSdss ignores this parameter.

Conversion processing automatically reblocks data sets to about 6KB, if the 6KB block size can store more data on a track than the original block size. This reblocking can provide more efficient track utilization and device independence for the data set.

During the conversion process, the DFSMShsm data set reblock exit (ARCCDEXT) is invoked for a data set if its VTOC entry indicates that it is reblockable. You can use this exit to change the block size to zero. This allows the system to select an optimum block size for the device to which the data set is recalled or recovered. The block size is recalculated if the data set is moved to a different device type.

Storing Small Data Sets

DFSMShsm SDSP helps conserve DASD space by writing small data sets as one or more records in a VSAM data set when it migrates them to migration level 1 volumes. This reduces the space requirements and fragmentation on migration level 1 volumes, and the DFSMShsm performance overhead.

Data sets eligible for SDSP are:

- Direct access data sets
- PDSs
- PDSEs
- Physical sequential data sets
- VSAM data sets
- Extended sequential data sets

When you use the SDSP option, specify the size of data sets (in kilobytes) for which you want to go into SDSPs. Data sets equal to, or less than, the specified size are directed to SDSPs.

You can use management class to set different age criteria for migrating SDSP-eligible data sets to migration level 2. You can use ACS routines to assign this management class to new data sets based on the maximum size of the data sets (using the &MAXSIZE variable).

Chapter 5. Improving I/O Performance

Managing the I/O performance in your installation is a complex task because of the various factors involved: processor time, the hardware configuration, the number of volumes in a storage group, the capacity of the volumes, where and how the data is placed on the volume. In the DFSMS environment, you can enable the system to control the placement of the data and automate many of the performance tuning tasks. You can take advantage of the latest hardware and software technology to better meet or exceed the performance requirements.

This chapter explains the many steps you can take to improve performance across the installation. They are:

- Establishing Performance Objectives
- Using Dynamic Cache Management
- Allocating extended sequential data sets
- Allocating temporary data sets
- Allocating data with lower performance demands
- Improving performance of shared data in a sysplex

Establishing Performance Objectives

This section describes how to use storage class to establish performance objectives for data. For sample storage classes, descriptions, and ACS routines, see *MVS/ESA SML: Implementing System-Managed Storage*.

Using Storage Class to Set Performance Goals

Use the storage class performance attributes to set your installation's performance objectives. They are:

- Direct bias
- Sequential bias
- Direct millisecond response
- Sequential millisecond response
- Initial access response seconds
- Sustained data rate

The BIAS attribute is always used for volume selection. Use it to specify the prevalent access for data sets assigned to this storage class. For data sets that have dominating read access, set BIAS to READ. For data sets that have dominating write access, set BIAS to WRITE. If the data set has equal read and write access, or if you do not know what the dominating access is, do not set a bias.

Through the MILLISECOND RESPONSE attributes, SMS provides dynamic data set caching for data sets residing on volumes that are attached through a cache-capable storage control. Unless 999 (never cache) is given for MILLISECOND RESPONSE, data sets are cache candidates if they reside on volumes on which caching is active, and which are attached to a cache-capable storage control.

The INITIAL ACCESS RESPONSE SECONDS attribute determines device selection for a particular data set allocation or object storage. For example, a value of 0 places the data on fixed DASD. Any non-zero value gives preference to placing a data set on mountable DASD, or an object on a mountable optical disk or tape volume. Note that because SMS is able to make the distinction during allocation, you can combine all 3390 look-alike devices (DASD and optical) in the same storage group.

The SUSTAINED DATA RATE attribute specifies the data transfer rate goal that should be maintained during a period of typical I/O activity for an extended sequential data set. The cache and fast write attributes are ignored for extended sequential data sets.

The SUSTAINED DATA RATE attribute, used with the INITIAL ACCESS RESPONSE SECONDS attribute, determines the removable media on which an object is to reside. For example, a non-zero value for INITIAL ACCESS RESPONSE SECONDS and a value of 3 or greater for SUSTAINED DATA RATE places the object on tape. If SUSTAINED DATA RATE has a value less than 3, the object is stored on an optical volume.

For more information on specifying storage class attributes, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Planning and Defining Storage Classes

When you plan your storage classes, identify data sets requiring high performance. Consider the capabilities of your hardware to meet these requirements. Balance the requirements of your key data sets against the cost of installing the devices that provide faster response times.

When you determine the response time and bias values for storage classes, consider the performance capabilities of devices that can be selected for new data set allocations. SMS attempts to select devices that meet the response time given in the storage class for the data set. However, if SMS cannot find one, the device selected may be slightly faster or slower than requested.

To select devices that meet the performance objective, SMS might allocate data sets to cached devices, even though cache is not required. For example, if a data set with an associated MILLISECOND RESPONSE of 15 milliseconds is placed on a cached device whose native millisecond response is slower than 15 milliseconds, then the data set is considered a must-cache candidate at open time.

When planning storage classes, you also need to identify data sets that are to be allocated to tape. These data sets must be assigned a storage class to be directed to a tape storage group. We recommend that you set up a separate storage class for tape data sets to ensure efficiency.

When determining storage class values, you need to consider:

- The various performance requirements of the data that will reside in each storage group
- How the devices in each storage group can meet these requirements

You use the ISMF Storage Class Application to define storage classes. Using the storage class performance attributes, you can choose the DASD I/O response time required by a data set and its read or write bias. The response time value is based on the time required to transfer a 4 KB block of data between the device and the processor, under optimal conditions. The value provides relative response time guidelines only; it does not guarantee performance. The bias indicates whether the predominant access mode for the data set is read or write.

When you define the storage class values to specify high performance objectives, make sure the hardware devices in the storage group definitions can satisfy these objectives. For information on defining storage groups, see *MVS/ESA SML: Managing Storage Groups*. For more information about planning and defining storage classes, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Storage class is also used for volume selection and to help provide availability of your data. For more information, see *MVS/ESA SML: Managing Storage Groups*, and Chapter 6, “Backing Up Data” on page 93.

Using Dynamic Cache Management

You can use dynamic cache management to achieve overall performance objectives of the performance-critical, non-extended sequential data sets. Dynamic cache management is a service that automatically determines which data sets are cached, based on the storage control's load, the characteristics of the data set, and the performance requirements that you define.

With DFSMS/MVS, if you are using a cache-capable storage control with the Extended Platform, you can take advantage of the enhancement to dynamic cache management. It optimizes the selection of data sets that are cache candidates. If the cache-capable storage control is not updated with the Extended Platform, the system uses a simpler algorithm.

Selecting Dynamic Cache Candidates

Only system-managed data sets residing on volumes attached through a cache-capable storage control can be cache candidates. Data sets that are accessed directly can benefit most from dynamic cache management.

Performance-critical database data can also benefit from a cache-capable 3990 storage control with DASD fast write, or dynamic cache management, as Figure 60 shows.

DB2	CICS	IMS
<ul style="list-style-type: none"> • DB2 directory (read/write, direct) • DB2 catalog (read/write, direct) • Application databases (read/write, direct or seq) • Active logs (write, sequential) • Archive logs (write, sequential) • Image copies (write, sequential) 	<ul style="list-style-type: none"> • System log (write, sequential) • Monitor log (write, sequential) • User log (write, sequential) • Restart data set (write, direct) • CICS availability manager (CAVM): <ul style="list-style-type: none"> – CAVM control data set (read/write, sequential) – CAVM message queue (read/write, sequential) • Auxiliary trace data set (write, sequential) • VSAM user data sets (read/write, direct) • BDAM user data sets (read/write, direct) 	<ul style="list-style-type: none"> • Data entry database—(DEDB) (read/write, direct) • Database—(DL/I) (read/write, direct) • Online data set—(OLDS) (write, sequential) • Write ahead data set—(WADS) (write, sequential) • System log data set—(SLDS) (write, sequential) • Restart data set—(RLDS) (write, sequential) • Recovery control data set—(RECON) (write, direct)

Figure 60. Improving I/O Performance for Database Data. These database data sets can benefit from dynamic cache management.

Note that IMS OSAM data sets cannot use the enhanced dynamic cache management, but can continue to use the previous algorithm. OSAM data sets can co-exist with data sets that are using the enhancement to dynamic cache management.

Implementing Dynamic Cache Management

To use dynamic cache management, you need to define the storage class performance attributes. To use storage class to specify performance objectives, see “Establishing Performance Objectives” on page 83.

For dynamic cache management, SMS recognizes data sets as falling in one of four categories, based on the bias specification and the millisecond response (MSR) specification in their respective storage classes:

Must-Cache Data sets with MSR specifications that cannot be met without use of cache, regardless of bias specification.

Never-cache Data sets with MSR values of 999, regardless of their read or write bias.

May-cache Data sets with no MSR, or with MSR specifications that can be met without the use of cache.

Fast Write/NVS Data sets with MSR specifications that can not be met without the use of non-volatile storage and BIAS is specified as WRITE.

Must-cache candidates are forced to always use cache. Only high priority data sets should be must-cache candidates. If a high priority data set does not cache well, make it a may-cache candidate and allow dynamic cache management to control its cache use.

Never-cache data sets are inhibited from using cache. Only use MSR = 999 if you know that the data set will never benefit from cache or non-volatile storage.

Dynamic cache management controls the cache use of the data sets in the may-cache category. May-cache candidates are eligible to use cache facilities only if the must-cache candidates are not fully occupying cache storage.

To achieve efficient I/O workload management, we recommend that you limit the number of must-cache data sets and allow the system to manage the caching of the may-cache data sets. With the new algorithm, the enhancement to dynamic cache management optimizes the selection of cache data sets and prevents over-commitment of non-volatile storage.

The enhancement to dynamic cache management tracks a data set's hit ratio. It monitors the data set's read and write hit ratio when the data set is initially opened. After some period of time, the ratio is compared to predetermined thresholds.

- If the data set read hit ratio is less than the read threshold, the data set is inhibited from the use of cache and cannot return for a period of time.
- If the write hit ratio is less than the write threshold, DASD fast write is inhibited for the data set.
- If the hit ratio indicates that the data set is a good cache candidate, it remains in cache for a period of time before the next hit ratio evaluation.

The system can also distinguish between different applications of the same data set. An application with a bad hit ratio is inhibited from using cache, while another application with a good hit ratio is allowed to use cache.

Data sets that are inhibited from the use of cache can return to cache if, at the next hit ratio evaluation, its recent hit ratio makes it a good cache candidate.

With DFSMS/MVS, you can collect I/O statistics to monitor the caching activities for all system-managed data sets that reside on volumes attached through cache-capable 3990 storage controls. For information on collecting the I/O statistics, see Chapter 9, "Monitoring and Reporting for Storage Management" on page 159.

Allocating Extended Sequential Data Sets

You can allocate large physical sequential data sets as extended sequential data sets to reduce the I/O service time normally required. This technique stripes a sequential data set over several volumes, on different channels and control units. The advantage is that it allows I/O operations to execute in parallel, thus providing significant improvement in the rate of data transfer between DASD and main storage.

Extended sequential data sets are written across the volumes selected for the allocation so that different volumes contain the consecutive tracks of a data set. The data blocks are written as fixed-length full blocks (maximum size), each with a 32-byte suffix that is not apparent to the applications. The suffix carries important information about the extended data set, such as the length of the data and the relative block number. To account for the space required for the suffix and to avoid space allocation problems, we recommend that your application programmers use system-determined block size.

Allocating extended sequential data sets requires the use of DASD attached to serial (ESCON) channels. Therefore, the storage groups you set up for these data sets must have ESCON connectivity. For information on planning and setting up effective storage groups, see *MVS/ESA SML: Managing Storage Groups*.

To ensure efficient striping of the data sets, audit the allocation requests. Check that the size of the data set is large enough to benefit from the technique, and that system-determined block size is used.

Determining Extended Sequential Data Set Candidates

Only system-managed data sets that have physical sequential organizations can be allocated as extended sequential data sets. Large, physical sequential data sets with high rates of I/O activity can benefit most from this allocation technique.

The following data sets should **not** be allocated as extended sequential:

- System data sets
- DFSMSHsm journal and log data sets
- DFSORT work (SORTWK) data sets
- VIO data sets
- Hiperbatch* data sets
- DFSMSdss DUMP data sets
- Data sets accessed with EXCP

Extended sequential data sets can only be accessed using BSAM or QSAM.

Allocating Sequential Data Sets in Extended Format

To allocate extended sequential data sets, you need to specify the data class and storage class extended sequential data set attributes. The data class attribute indicates whether a data set is to be allocated as extended sequential. The storage class attribute determines the target number of stripes for the data set.

To allocate a data set in the extended format, you need to specify EXTENDED for the data class DSNTYPE attribute. If DSORG=PS is not used with the data class ACS routine or in the JCL DD statement, then DSNTYPE of EXTENDED is ignored.

Figure 61 on page 89 shows an ISMF Data Class Define panel using a DSNTYPE of EXTENDED.

```

                                DATA CLASS DEFINE                                Page 1 of 2
COMMAND ===>

SCDS NAME:      ACTIVE
DATA CLASS NAME: STRIPE

TO DEFINE DATA CLASS, SPECIFY:
DESCRIPTION ===>
                ===>
RECORD          ===>      (KS, ES, RR, LS or blank)
RECFM           ===> VB   (any valid RECFM combination or blank)
LRECL           ===>      (1 to 32761 or blank)
KEYLEN          ===>      (0 to 255 or blank)
KEYOFF          ===>      (0 to 32760 or blank)
SPACE AVGREC    ===> U    (U, K, M or blank)
  AVG VALUE     ===> 255   (0 to 65535 or blank)
  PRIMARY       ===> 5000  (0 to 999999 or blank)
  SECONDARY     ===> 5000  (0 to 999999 or blank)
  DIRECTORY     ===>      (0 to 999999 or blank)
DATA SET NAME TYPE ===> EXT (EXT, HFS, LIB, PDS or blank)
  If EXT        ===> P    (P=Preferred, R=Required, or blank)

USE ENTER TO PERFORM VERIFICATION;
USE HELP COMMAND FOR HELP; USE END COMMAND
TO SAVE AND EXIT; CANCEL TO EXIT.

```

Figure 61. Specifying Extended Format Allocation

You can indicate whether the extended allocation is required or preferred. If you enter extended preferred, the data set is allocated in the extended format, if possible. If no ESCON-attached volumes are available, or if there is not enough space on available volumes, the data is allocated as non-extended.

If you specify extended required, the data set must be allocated in the extended sequential format. If no ESCON-attached volumes are available, or if there is not enough space on the available volumes, the allocation fails.

In addition to placing EXTENDED DSNTYPE in data class, you can use the storage class attribute, SUSTAINED DATA RATE, to set a goal for the data transfer rate, in MBs per second. The system uses this goal, in combination with the device transfer rate, to determine the number of stripes, or volumes, required to match the performance objective of your data set. The maximum number of stripes, or volumes, that can be allocated for an extended sequential data set is 16. Each extended sequential data set can have up to 123 extents on each volume.

Figure 62 on page 90 shows an ISMF Storage Class Define panel using a SUSTAINED DATA RATE. This attribute is ignored if you do not use EXTENDED DSNTYPE in the data class.

```

                                STORAGE CLASS DEFINE
COMMAND ===>

SCDS NAME:          STGADMIN.SCDS
STORAGE CLASS NAME: STRIPE

TO DEFINE STORAGE CLASS, SPECIFY:
  DESCRIPTION ===>
                ===>

PERFORMANCE OBJECTIVES
  DIRECT MILLISECOND RESPONSE    ===>          (1 to 999 or blank)
  DIRECT BIAS                    ===>          (R, W or blank)
  SEQUENTIAL MILLISECOND RESPONSE ===>        (1 to 999 or blank)
  SEQUENTIAL BIAS                ===>          (R, W or blank)
  INITIAL ACCESS RESPONSE SECONDS ===>        (0 to 9999 or blank)
  SUSTAINED DATA RATE (MB/SEC)   ===> 3.5    (0 to 999 or blank)
  AVAILABILITY                   ===> STANDARD (STANDARD or CONTINUOUS)
  ACCESSIBILITY                  ===> P       (C, P, S or N)
  GUARANTEED SPACE               ===> N       (Y or N)
  GUARANTEED SYNCHRONOUS WRITE   ===> N       (Y or N)

USE ENTER TO PERFORM VERIFICATION;
USE HELP COMMAND FOR HELP; USE END
COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

```

Figure 62. Specifying SUSTAINED DATA RATE for Extended Data Sets

Valid entries for SUSTAINED DATA RATE are 0 to 999, and blank. If you specify 0 or leave the field blank, the target number of stripes is 1. A single striped data set can experience improved I/O rate if the QSAM access mode is used.

Using SUSTAINED DATA RATE enables the system to determine the number of stripes to allocate for your data set. To find out how many stripes are allocated to an extended sequential data set, use the ISMF Data Set Application to display information about the data set. The NUMBER OF STRIPES attribute shows the number of stripes allocated to extended sequential data sets. For more information on using the ISMF Data Set Application, see *DFSMS/MVS V1R2 Using ISMF*.

Sometimes you might want to manually specify the number of stripes to allocate. If you state GUARANTEED SPACE = YES, you can decide the number of stripes to be allocated using VOLCNT (volume count), and SUSTAINED DATA RATE is ignored.

If you choose GUARANTEED SPACE = YES, the primary space is for each stripe of the extended data set. The secondary space is divided among the stripes of the extended data set.

When GUARANTEED SPACE = NO, both the primary and secondary space are divided among the stripes of an extended data set.

Allocating Temporary Data Sets

Temporary data sets normally exist only for the duration of a job, job step, or terminal session. To ensure efficient use of space, make sure these data sets are system-managed. This allows DFSMSHsm to automatically delete these system-identified temporary data sets during primary space management.

You can allocate most temporary data sets as VIO data sets to improve performance. However, VIO does not support temporary VSAM data sets or temporary PDSEs.

Although VIO supports DFSORT data sets, we do not recommend that you allocate DFSORT output and temporary work data sets as VIO data sets. VIO reduces elapsed time at the cost of increased CPU time for DFSORT applications.

You can still allocate DFSORT temporary data sets as VIO data sets, if:

- The system supports VIO in expanded storage.
- Elapsed time is of greater concern than CPU time.
- The temporary data sets are passed to and from other job steps whose performance can be improved by the use of VIO.

To ensure that all temporary data sets are system-managed, you need to design your storage class and storage group ACS routines effectively. For information on writing the ACS routines, see *MVS/ESA SML: Implementing System-Managed Storage*.

Allocating Data with Lower Performance Demands

You might have data that has lower performance requirements than traditional DASD applications, but still requires faster access than tape or optical storage can provide. If so, you can allocate or move the data to IBM 3390 Model 9 devices.

The 3390 Model 9 is a DASD that has three times the capacity of a 3390 model 3, but its performance level might be as low as one-third of the other 3390 models. Despite its slower seek time and longer latency, the 3390 Model 9 maintains the 3390 track format and data rate. You can create a new storage group to ensure that applications needing traditional high-performance DASD are not allocated to the 3390 Model 9.

This section discusses the types of data that can benefit from the use of 3390 Model 9 and how to maintain acceptable performance. You can find detailed information on the 3390 Model 9 in *Using 3390 in an MVS Environment*.

Selecting Lower Performance Candidates

Before allocating data to 3390 Model 9 devices, consider the performance requirements of the data, as agreed to in the service-level agreement with your users.

Some candidates for the 3390 Model 9 include:

- New applications that are currently not on DASD because their capacity requirements or low use characteristics are not cost effective on high performance DASD. The 3390 Model 9 provides a combination of performance capabilities and low cost-per-MB characteristics that can justify implementing new applications.
- Data with low access characteristics. Certain types of data that are currently on DASD or tape can be moved to the 3390 Model 9 and sustain acceptable performance. They are:
 - DFSMShsm migration level 1 data
 - Some tape mount management buffers
 - Certain large, sequentially processed data sets
 - Infrequently accessed history or archival files
- Information currently stored on microfiche
- Image data and indexes
- Data whose retrieval requirements fall below that of data stored on high performance DASD, but whose demands are still above the level provided by tape
- Backup copies of databases
- Infrequently accessed database information
- Test copies of databases

Improving Data Access Rate

Uncached performance might be acceptable for much of the data allocated or migrated to the 3390 Model 9. However, configurations with a cached 3990 storage control can provide improved performance and offer availability functions, such as concurrent copy and fast dual copy.

Improving the Performance of Shared Data in a Sysplex

For data shared across MVS systems, you can achieve performance improvements with the sysplex configuration. The new sysplex is an extension of the traditional MVS sysplex, combining up to 32 systems in one system complex. This increases the throughput and decreases the response time significantly. Software in the sysplex uses the coupling facility, a PR/SM feature on certain ES/9000 models, to enable applications to share data across systems with integrity, and maintain performance approaching that available in a single system.

Online transaction processing can benefit from the sysplex configuration. Under MVS, the CICS and IMS transaction managers can fully use the new sysplex capabilities. The IMS database, DB2, and VSAM can take advantage of the sysplex to improve their data sharing abilities. For more information on the enhanced sysplex, see *System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism*, and *System/390 MVS Sysplex Hardware and Software Migration*.

Chapter 6. Backing Up Data

This chapter presents the issues to address when developing a backup plan and describes different backup tasks, including:

- Developing a backup plan
- Establishing availability standards
- Backing up data
- Backing up data critical to the installation
- Storing vital records and archiving data
- Performing aggregate backups

For information on backup and recovery of DASD volumes, see *MVS/ESA SML: Managing Storage Groups*.

Developing a Backup Plan

The purpose of a backup plan is to ensure the prompt and complete recovery of data. A well-documented plan identifies data that requires backup, the levels required, responsibilities for backing up the data, and methods to be used.

Preparing for General Backups

Major issues you should include in your plan are:

- **Identify critical applications.**

Determine priorities for your applications, assigning the highest priority to those that impact your business the most. Include upper-level management in this process.

Be sure to identify the critical applications and document their backup and retention requirements in the service level agreement.

- **Identify data associated with the application.**

Once you identify the critical applications, identify the data associated with them. This includes all data sets directly related to the application (such as master files and update files), as well as data sets required to support the application (such as JCL procedures and CLISTs). You might have data sets with instructional information that explains how to recover and run the application.

Each application should have its own self-contained *aggregate* of data sets. If certain data sets are shared by two or more applications, you might want to ensure application independence for disaster recovery by backing up each application that shares the data. This is especially important for shared data in a distributed environment.

We recommend that you appoint an owner for the shared data set to avoid inconsistency between backup copies. The owner is someone who runs the critical application in the installation and controls the aggregate backup process.

- **Determine backup frequency, the number of versions, retention period, and backup technique.**

Determine the backup frequency, the number of backup versions, and the retention period by consulting user group representatives. Be sure to consider whether certain data backups need to be synchronized. For example, if the output data from application A is used as input for application B, you must coordinate the backups of both applications to prevent logical errors in the data when they are recovered.

Identify the different types of backup techniques required by the different types of data. Consult your user group representatives for this task. The techniques are discussed later in this chapter.

- **Minimize installation and device dependencies.**

Ensure that esoterics, average record allocation, and generic profiles are used whenever possible to avoid device dependencies. You will not be able to avoid some dependencies; for example, if your site uses JES3 and the recovery site uses JES2, you will need to modify JCL during the recovery process. Also, be aware of other specialized installation requirements, such as user exits.

- **Provide documentation for recovery.**

Because personnel from your site might not be available during the recovery process, provide thorough documentation to communicate what needs to be done. Create data sets with instructions for recovery, and keep them with the application's package of data sets. You can provide some hard copy general instructions that address the overall recovery process and explain how to use the instruction data sets.

We recommend using naming conventions for the instruction data sets. For example, use the LLQ INSTRUCT. This makes it easier for the recovery team to identify them.

- **Keep an inventory of the backup data.**

Establish a method for tracking which data is backed up, when it is backed up, how many versions are made, and which backups are transported to the recovery site.

You can store backup data sets on DASD or tape (this does not apply to objects). Your choice depends on how fast the data needs to be recovered, media cost, operator cost, floor space, power requirements, air conditioning, the size of the data sets, and whether you want the data sets to be portable.

For some users, placing backup data sets (especially point-in-time backups) on tape can be a hindrance to productivity because of the amount of time, labor, and cost involved in managing tape.

With DFSMS/MVS, you can:

- Use the volume mount analyzer to help you implement tape mount management techniques in your installation.

Using these techniques, you can automatically fill tapes to capacity, and reduce tape mounts and the labor involved in the mounts, thus reducing the expense of tape mount activity.

- Use the software facilities of DFSMS/MVS, along with the IBM 3495 or 3494 Tape Library Dataserver.

You can extend your system-managed storage environment to include tape cartridges and automate tape handling operations to reduce the time delays in tape mounts.

- Use DFSMSrmm, a functional component of DFSMS/MVS, to help you manage your tape volumes.

DFSMSrmm keeps an inventory of all your shelf space and demountable volume resources. It tracks the location of the volumes, the data sets that are written to tape, and volume ownership. DFSMSrmm also helps automate tasks such as tape labeling, erasing, expiration processing, and vital record processing. By centralizing volume information and automating most inventory-related tasks for tape volumes, you can manage your resources in a more productive and cost-effective manner.

For more information on implementing tape mount management techniques, see *MVS/ESA SML: Implementing System-Managed Storage*. For information on planning and defining system-managed tape storage groups, see *MVS/ESA SML: Managing Storage Groups*. For information on using DFSMSrmm, see *DFSMS/MVS V1R2 DFSMSrmm Guide and Reference*.

Note: You can only store object backups on optical and tape volumes. However, if the object is stored in a DB2 table space on DASD, you might also back up the object using DB2 image copy utilities.

Preparing for Disaster Recovery Backups

Backups for disaster recovery require special considerations that normally do not apply to other types of backups.

One basic difference between regular backup data and that used for disaster recovery is that disaster recovery backups might need to be transported to a different site. Recovery is generally done at the remote site. However, some installations might use the remote site strictly for storage and, once damages are repaired, return the applications for recovery to the site where the disaster occurred.

For information on disaster recovery, see Chapter 7, “Recovering Data” on page 129.

Remote Site Backups

You can transport backup data to a remote site, either physically on portable media, or directly over the network.

For either method, consider the differences between your operating environment and that of the recovery site. Set up methods to resolve compatibility problems. Also, consider how the recovery site will handle the extra workload that it receives. Plan for recovery to another existing system or recovery of the original system, and add this to your plan.

If you recover to an existing system, you might need to merge into this system, which means that you might need to use that system’s DFSMSHsm, RACF, and catalog environment. In this case, maintain the following in off-site storage:

- Backups of the DFSMSHsm and SMS control data sets, and the data set that contains the source code for the ACS routines. Merge the information in these data sets with the control data sets on the recovery site’s system.

If you have a DFSMSrmm control data set, and tape and optical configuration databases, keep backup copies of these at the off-site storage.

- A backup of the member of SYS1.PARMLIB that contains the DFSMSHsm start-up parameters. This member will need to be merged with the PARMLIB member on the recovery system.

If you have DFSMSrmm installed, have a backup copy of the DFSMSrmm SYS1.PARMLIB member.

- Lists of definition requirements for catalog aliases, GDG bases, and RACF profiles.

You can recover the original system to a recovery site, either by replacing a system or by creating a new system. To do either, keep backups of the following data sets:

- System master catalog
- Dumps of system packs to use in stand-alone restores
- DFSMSHsm and SMS control data sets, and the data set that contains the source code for the ACS routines
- RACF data set

Disaster Backup Methods

You can perform disaster recovery backup either by volume (using DFSMSdss full volume dumps) or at the application level (using DFSMSHsm ABARS logical aggregate group backups or DFSMSdss logical dumps). We recommend you perform your disaster recovery backups at the application level.

Backup by application is preferable because it ensures device independence. This is extremely important because the target volumes at the remote site might not be of like device types.

Backup by application can also be beneficial for certain data sets, such as multi-volume or extended sequential data sets, because you will not lose any part of the data set.

Using ABARS, you can run up to 15 backup tasks simultaneously, and create up to 15 copies of aggregate backup files in parallel. ABARS can provide backup processing for your critical tape data sets. For information on using DFSMSHsm ABARS logical aggregate group backups, see “Performing Aggregate Backups” on page 117.

We do not recommend physical volume dumping for disaster backup unless you can ensure that the volume contents are not changed during the dump process. If you choose to use full volume dumps for special situations, we offer these recommendations:

- Full volume dumps run more quickly than logical dumps of all the data sets on a volume, but by dumping only the data sets critical to an application you considerably reduce the amount of data dumped and the time it takes to do it. This can reduce the number of output tapes, tape mount activity, and requirements for off-site storage facilities. Also, using concurrent copy for your logical or full volume dumps almost eliminates any outage time required to do the dumps. If you use concurrent copy, make sure that the contents of the

volumes do not change while the concurrent copy session is being initialized. For more information, see “Concurrent Copy” on page 104.

- You can use the DFSMSHsm invocation of DFSMSdss to automate the dump process.

Establishing Availability Standards

You can use both management class and storage class to provide automatic daily backup of system and user data at an installation level. By using SMS management classes and storage classes, you can also automatically back up data base management system (DBMS) libraries, test systems, and their associated data sets, CICS user databases, and DB2 user databases.

Using Management Class Backup Attributes

The management class backup attributes determine backup activity for system-managed data. The values you specify in the management class definition control:

- Backup frequency
- Number of backup versions
- Who can perform command backup
- How long to keep the backup versions
- Whether or not automatic backup occurs
- Whether or not to back up data sets while they are open
- Whether or not to use concurrent copy as the copy technique

The *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference* describes the management class backup attributes in detail.

For information on planning and defining management classes, see Chapter 4, “Managing Space Use” on page 59.

Using the Storage Class Availability Attributes

You can use two attributes, AVAILABILITY and ACCESSIBILITY, to manage storage class availability.

The AVAILABILITY attribute provides hardware dual copy for critical data sets. Dual copy keeps a duplicate copy of the data set on a different DASD volume. Both copies are updated each time the data set is updated, so if one device fails, the data is still available on the other. Dual copy requires dual copy-capable 3990 storage controls. If you use CONTINUOUS AVAILABILITY for a new data set allocation, and a dual copy volume is not available, the allocation fails.

The ACCESSIBILITY attribute lets you allocate new data sets to devices that support the concurrent copy function. Data sets on these volumes are then available for the concurrent copy operation.

You can use the storage class GUARANTEED SYNCHRONOUS WRITE attribute for data sets requiring I/O synchronization to the media. This provides data integrity for critical application data and is recommended for PDSEs when necessary.

Backing Up Data With Standard Availability Requirements

DFSMSHsm can create backup versions of data sets either automatically or by command, placing them on specially assigned tape or DASD volumes called *daily backup volumes*. When more space is needed on the daily DASD backup volumes, DFSMSHsm provides a *spill process* whereby older backup data sets are moved to *spill volumes*. You can consolidate data on tape backup volumes and on migration level 2 volumes by using the DFSMSHsm *recycle* process.

You can also compact the data using the IDRC feature of the 3490 cartridge-type device (you have the option of installing IDRC on your 3480 subsystems). With IDRC, you can compact data when it is backed up to tape. With the 3490E 36-track recording mode, twice the amount of data can be stored on a cartridge. And with the Enhanced Capacity Cartridge System Tape, you can store four times as much data on a cartridge.

There are two ways you can control IDRC compaction. Use the data class COMPACTION attribute to request IDRC for a tape library device.

If the allocation is to a tape outside of the library, and you have the 3490 Magnetic Tape Subsystem control unit, you can automatically take advantage of the IDRC feature. DFSMSHsm always uses IDRC to write to 3490 tape devices.

If you have a 3480 tape device with IDRC, use the DFSMSHsm SETSYS TAPEHARDWARECOMPACT parameter to enable DFSMSHsm to request IDRC compaction on eligible 3480 tape cartridges. See "Compacting Data" on page 79 for an example of specifying the TAPEHARDWARECOMPACT parameter to request IDRC.

If you use 3480 tape devices with the IDRC feature for your DFSMSHsm backup volumes, we recommend that you use the IDRC feature of the 3480 instead of the DFSMSHsm data set compaction option. You can save CPU cycles by using IDRC instead of DFSMSHsm data set compaction.

Besides its ability to back up data sets automatically, an important advantage that DFSMSHsm has over other backup tools is that it automatically creates and maintains an inventory of all backup data sets. It keeps this inventory in the backup control data set (BCDS). Also, DFSMSHsm automatically uses DFSMSdss for migration and backup processes.

This section discusses ways to back up data sets, database data, and objects. Note that you use OAM to back up objects to either optical or tape volumes. However, if the object is stored in a DB2 table space on DASD, you might also back up the object with DB2 image copy.

Determining Backup Frequency and Number of Versions

How often you back up data and the number of backup versions you maintain is determined by the user's recovery requirements. Therefore, you need to negotiate these issues with the user groups.

When you back up an object, only one backup copy is made because the data contained in an object cannot be changed. The backup copy of an object is retained until the object itself is deleted; then the backup copy is deleted also.

The factors that determine the frequency of data set backups are usually some combination of the following:

- Time and resource required to resynchronize or update the backup copy after recovery.
- Rate at which the data changes. For example, if a data set has its contents changed every day, a backup frequency of once a week is not sufficient.
- Relative ease of rebuilding a data set. For example, it is easier to rebuild a program load library than a source library.
- Requirements for backing up multiple versions at a high frequency.
- Confidence of the data owner in the reliability of the system or the backup process.
- Timeliness of backup relative to completion of job.
- Size of backup window.

The factors governing the number of data set backup versions are usually:

- Probability of logical errors occurring in the source data

Logical errors are seldom detected during backup and, as a result, errors are recorded in the data set's backup copy. For example, if a user accidentally deletes a required member of a PDS and does not notice the mistake before the data set is backed up, the user will need an older level of backup from which to recover. Another example is when an application data set changes daily, but an error might go undetected for a week.

- Importance of the data

In any data set backup procedure, there is a risk that the backup copy will become damaged or lost at the same time as the original. Some data is so vital or expensive to recreate that you must keep several duplicate backup copies to reduce the risk of total loss.

- Cost of backup storage

The extra cost of maintaining multiple backup copies is mostly confined to the cost of storing the extra data (the cost of producing the multiple backup copies is relatively small). The amount of storage you can afford to reserve for this extra data can affect the number of copies you can maintain.

Using Backup Frequency and Number of Versions

By using DFSMSHsm with SMS management classes, you can specify backup frequency and version requirements at the data set level.

To control the frequency of backup and the number of backup versions for system-managed data, use the management class `BACKUP FREQUENCY` and `NUMBER OF BACKUP VERSIONS` attributes.

`BACKUP FREQUENCY` is the minimum number of days that must elapse between automatic backups of a data set. If you choose 0, the data set is automatically backed up if it changes. If you select 1, it is not backed up more than once on the same day.

NUMBER OF VERSIONS is the number of backup versions DFSMSHsm retains for data sets on primary and migration volumes, and for deleted data sets. If you choose a version value of 0, it is not to be backed up at all.

NUMBER OF VERSIONS also influences the number of days DFSMSHsm retains backup versions. For a data set that is changed daily, if you want 5 for NUMBER OF VERSIONS, and 1 for BACKUP FREQUENCY, then on the sixth day, the oldest backup is deleted because only five versions can be kept.

The frequency of backup and the number of backup versions you want are based on recovery requirements for the different types of data sets you use. For example, program source data sets might need more frequent backup than program object data sets. You might also have data types that you do not want to include in the DFSMSHsm automatic backup cycle. For example, you might want to exclude DB2 table spaces and index spaces. The DB2 utilities provide backup, duplication and recovery of DB2 table spaces and index spaces.

Figure 63 shows an example of data set backup criteria for selected data types:

Figure 63. Determining Backup Criteria

Data Type	Frequency	Versions
TSO logon/ISPF	1	2
Program Source	0	5
Program Object	3	1
Load	1	2

Choosing Guaranteed Backup Frequency

To ensure that all unchanged data requiring backup within a storage group is regularly backed up by DFSMSHsm, you can use the storage group attribute GUARANTEED BACKUP FREQUENCY. Within the applicable storage group, this attribute determines the number of days within the last backup period in which DFSMSHsm should back up each data set. The data sets allocated to the storage group, even if unchanged, are backed up automatically if not already backed up during the number of days you specify.

Depending on the backup process used, the management class attribute AUTO BACKUP=N overrides the storage group request for automatic backup. The GUARANTEED BACKUP FREQUENCY attribute is valid only for POOL storage groups that say AUTO BACKUP=Y.

We recommend that you use GUARANTEED BACKUP FREQUENCY with DFSMSHsm incremental backup instead of full volume dumps (AUTO DUMP=Y), when the dump window is too short and the number of DASD volumes is too large to do DFSMSHsm full volume dumps. With this method, all unchanged data sets on volumes in the same storage group are backed up within the period stated in the storage group GUARANTEED BACKUP FREQUENCY attribute. The changed data sets are backed up by incremental backup.

The advantages of using GUARANTEED BACKUP FREQUENCY with incremental backup are:

- Tape mount reduction

You can reduce the number of full volume dumps, which results in the reduction of tape mounts.

- Ease of recovery

It places unchanged data sets on recent backup tapes. This allows you to recover volumes more easily, especially if there is no restore dump to work with as a starting point.

Using GUARANTEED BACKUP FREQUENCY with incremental backup might result in frequent recycling of DFSMSShsm backup tapes. You can use the DFSMSShsm recycle process to consolidate data on fewer tapes. For more information, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

For more information on GUARANTEED BACKUP FREQUENCY, see *MVS/ESA SML: Managing Storage Groups*.

Automatic Data Backup

DFSMSShsm automatically backs up data on system-managed volumes if the storage group definition for the volumes is AUTO BACKUP and the management class definitions for the individual data is AUTO BACKUP. Defining SMS storage groups is discussed in *MVS/ESA SML: Managing Storage Groups*.

The backup of extended format data sets is successful only if the target system supports the extended format. To back up extended format data sets to a system that does not support the format, use the access method services EXPORT command.

Database data sets not backed up by the DBMS backup and recovery functions can benefit from this automatic backup function.

Objects can also benefit from automatic backups. If the object has a management class that uses AUTO BACKUP, the backup is made during the first OAM storage management cycle after the object is stored, or during the storage management cycle that assigns the management class with AUTO BACKUP for the first time. Only one backup copy of an object is made because the data contained in an object cannot be changed. It is retained until the object itself is deleted; the backup copy is then also deleted. No archive copy is retained.

The location of the backup copy is determined by the specifications in the object backup storage group. For information on setting up the object backup storage group, see *MVS/ESA SML: Managing Storage Groups*.

Check SYSLOG for backup failures, because OAM reports by exception only. For information about a backup of a specific object, you should use Structured Query Language (SQL) to get information from the object directory.

For additional information on backing up and recovering objects, see the *DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Object Support*.

For data sets that are accessed with Distributed FileManager and have associated DDM attributes, use IEBCOPY and IDCAMS EXPORT to back up these data sets. You can also use DFSMSHsm to back up the data sets if it uses DFSMSdss as the data mover. IEBCOPY, IDCAMS EXPORT, and DFSMSdss automatically copy the attributes that need to be retained to allow remote access. For more information on the backup and recovery of these data sets, see *DFSMS/MVS V1R2 DFM/MVS Guide and Reference*.

For HFS data sets, the entire data set is backed up. To back up individual files, you must manually issue specific OpenEdition shell commands. For more information on using shell commands, see *OpenEdition MVS Command Reference*.

Incremental Backups

During automatic backup, done on specific days in a backup cycle beginning at a particular time, DFSMSHsm performs incremental backup. Data sets on volumes with storage groups that do not have GUARANTEED BACKUP FREQUENCY are backed up by DFSMSHsm according to the options you use on the INCREMENTALBACKUP parameter of the SETSYS command. The options are:

CHANGEDONLY

Backs up a data set if it has changed since the last time DFSMSHsm backed it up and a specified number of days have passed since DFSMSHsm last backed it up (the **frequency** of backup).

ORIGINAL

Backs up a data set on the same basis as CHANGEDONLY, and backs up a data set if a backup version does not already exist.

An advantage of incremental backup over full volume dump is that only data sets that have changed since the last time they were backed up, or data sets that do not have backups, are backed up. If you keep multiple versions, you will know that each new version is logically different from the previous one, saving you time comparing versions if you have to recover an older version.

Full Volume Dumps

You should use the DFSMSHsm invocation of DFSMSdss full volume dumps to complement DFSMSHsm incremental backups. Full volume dumps used together with incremental backups can provide a more accurate reconstruction of a volume's contents than volume dumps alone. Using full volume dumps with incremental backups can also provide faster volume recovery than using incremental backups alone.

The full volume dump function provides automatic record keeping in soft copy format, eliminating the need to maintain DFSMSdss output information. Full volume dump also can replace JCL or application programs written to automatically invoke DFSMSdss.

Full volume dump for system-managed volumes is controlled by the AUTO DUMP attributes in the storage group definition. For more information on DFSMSHsm full volume dumping, see *MVS/ESA SML: Managing Storage Groups* and *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Backup-While-Open

There are two ways you can back up a data set while it is being used: you can use the TOLERATE(ENQFAILURE) keyword with the DFSMSdss COPY or DUMP command to permit backups while a data set is open, or you can use the backup-while-open function of DFSMSdfp.

Specifying the DFSMSdss TOLERATE(ENQFAILURE) Keyword: You can use the TOLERATE(ENQFAILURE) keyword with the DFSMSdss COPY or DUMP command to allow the backup of data sets when they are in use. For more information on using TOLERATE(ENQFAILURE), see *DFSMS/MVS V1R2 DFSMSdss Storage Administration Reference*.

Using the Backup-While-Open Function: The DFSMSdfp backup-while-open function allows backups of system-managed VSAM data sets while the data sets are being accessed by an application.

Backup-while-open is only useful for applications, such as database manager systems, that have a recovery log facility. The logs contain information needed to recover the data sets to the version before the failure. This backup function is not used for online applications that can close and deallocate relevant database data sets to take a backup copy in a consistent state.

To enable a data set to use backup-while-open, you need to write an application program to call the DFSMSdfp Callable System Service, IGWABWO. The macro marks data sets as eligible for backup-while-open, and allows the database backup utility to do a backup while the database is open for updates.

CICS automatically makes use of backup-while-open when you specify BACKPTYPE(DYNAMIC) in the CICS CEDA (Resource Definition Online Transaction) define file. CICS backups are taken automatically through the DFSMS automatic backup support.

Whenever you invoke backup-while-open through IGWABWO or CICS, an authorization check is done to ensure data integrity. Without sufficient update authorization, the backup request is denied.

Enhancements to the backup-while-open function also address other integrity issues, including:

- Backup-while-open processes do not overlap.
- A backup-while-open dump is invalidated if a batch program updates the data set at the same time that DFSMSdss is dumping it.

Once data sets are enabled for backup-while-open, we recommend that you take the backups during low-activity time. Heavy I/O activity can affect application response time, and cause CI or CA splits for key sequenced data sets, thus invalidating the backups.

For more information on the backup-while-open function, see *DFSMS/MVS V1R2 Using Advanced Services for Data Sets*.

Concurrent Copy

Concurrent copy is available with DFSMS/MVS, and requires 3990 storage controls with the extended platform. It enables dumping or copying of your data while it is in use, significantly reducing application outages caused by backup processing.

Database data, such as DB2, can benefit from concurrent copy because it allows copying of the data during normal database operation.

We recommend that you also use concurrent copy for DFSMSHsm control data sets. During its backup cycle, DFSMSHsm normally copies its own control data sets and journal. This process can be time-consuming at larger installations and can stop other DFSMSHsm activity during this period. In a JES3 environment, it prevents the system from scheduling additional work.

To use concurrent copy, follow these steps:

1. Identify data that can benefit from concurrent copy.
2. Place the data on devices that support concurrent copy. Use the storage class ACCESSIBILITY attribute to direct new allocations to these devices. Figure 64 is a sample ISMF Storage Class Define panel with an ACCESSIBILITY parameter.

```

                                     STORAGE CLASS DEFINE
COMMAND ==>

SCDS NAME:          STGADMIN.SCDS
STORAGE CLASS NAME: CRITICAL

TO DEFINE STORAGE CLASS, SPECIFY:
  DESCRIPTION ==>
              ==>

PERFORMANCE OBJECTIVES
DIRECT MILLISECOND RESPONSE   ==> 10      (1 to 999 or blank)
DIRECT BIAS                    ==>         (R, W or blank)
SEQUENTIAL MILLISECOND RESPONSE ==> 10      (1 to 999 or blank)
SEQUENTIAL BIAS                ==>         (R, W or blank)
INITIAL ACCESS RESPONSE SECONDS ==>         (0 to 9999 or blank)
SUSTAINED DATA RATE (MB/SEC)  ==>         (0 to 999 or blank)
AVAILABILITY                   ==> STANDARD (STANDARD or CONTINUOUS)
ACCESSIBILITY                   ==> C       (C, N, P or S)
GUARANTEED SPACE               ==> N       (Y or N)
GUARANTEED SYNCHRONOUS WRITE   ==> N       (Y or N)

USE ENTER TO PERFORM VERIFICATION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.
```

Figure 64. Specifying ACCESSIBILITY for Concurrent Copy Data Sets

These are the ACCESSIBILITY selections:

CONTINUOUS

Allocation to a concurrent copy-capable volume is required. If the data cannot be allocated to such a volume, the allocation fails.

NO PREFERENCE

Allocation may be first attempted on either a concurrent copy or non-concurrent copy-capable volume. Both are primary volume candidates.

CONTINUOUS PREFERRED

Allocation to a concurrent copy-capable volume is attempted. If a concurrent copy-capable volume is not available, the data set is allocated to a non-concurrent copy volume. Primary volume selection candidates do not include non-concurrent copy volumes.

STANDARD

Allocation to a non-concurrent copy-capable volume is attempted. If the attempt is unsuccessful, the data set is allocated to other volumes from secondary volume candidates. This might result in the allocation to a concurrent copy-capable volume. Concurrent copy volumes are not considered primary volume candidates.

3. Select the management class BACKUP COPY TECHNIQUE attribute to indicate concurrent copy is to be used with AUTO BACKUP, or command data set backup. You can also use concurrent copy with aggregate backup by indicating the management class ABACKUP COPY TECHNIQUE.

Figure 65 shows an ISMF Management Class Define panel with the BACKUP COPY TECHNIQUE attribute. See “Managing ABARS Output Files” on page 126 for an example of the ABACKUP COPY TECHNIQUE attribute.

```
MANAGEMENT CLASS DEFINE                                     Page 3 of 5
COMMAND ===>
SCDS NAME:          STGADMIN.SCDS1
MANAGEMENT CLASS NAME: DBSTAN

TO DEFINE MANAGEMENT CLASS, SPECIFY:

BACKUP ATTRIBUTES
BACKUP FREQUENCY          ===> 0    (0 to 999 or blank)
NUMBER OF BACKUP VERS     ===> 3    (1 to 13 or blank)
  (DATA SET EXISTS)
NUMBER OF BACKUP VERS     ===> 1    (0 to 13 or blank)
  (DATA SET DELETED)
RETAIN DAYS ONLY BACKUP VER ===> 400 (1 to 9999, NOLIMIT or blank)
  (DATA SET DELETED)
RETAIN DAYS EXTRA BACKUP VERS ===> 100 (1 to 9999, NOLIMIT or blank)
ADMIN OR USER COMMAND BACKUP ===> BOTH (BOTH, ADMIN OR NONE)
AUTO BACKUP               ===> Y    (Y or N)
BACKUP COPY TECHNIQUE     ===> R    (P=Conc Preferred,
                                     R=Conc Required or S=Standard)

USE ENTER TO PERFORM VERIFICATION; USE UP/DOWN
COMMAND TO VIEW OTHER PANELS;
USE HELP COMMAND FOR HELP; USE END
COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.
```

Figure 65. Defining Aggregate Backup Attributes Using Management Class

You can specify one of the following parameters with BACKUP or ABACKUP COPY TECHNIQUE:

CONCURRENT REQUIRED

Concurrent copy is required. If the data set is not eligible for concurrent copy, the process terminates.

CONCURRENT PREFERRED

Concurrent copy is only used if the data set is eligible for concurrent copy. Otherwise, normal backup or aggregate backup processing is used.

STANDARD

Concurrent copy is not used, even if the data set is eligible for concurrent copy. Normal backup or aggregate backup processing is used.

If you specify R (Concurrent Required) for ABACKUP COPY TECHNIQUE, all primary level DASD data sets in the aggregate group must be on concurrent copy-capable volumes or ABACKUP process fails.

Concurrent copy can also be used in conjunction with the backup-while-open function of DFSMSdftp. If you specify concurrent copy as the copy technique for data eligible for backup-while-open, the application has complete and unrestricted access to the data. The backup copies will never be invalidated because of CI or CA splits.

To use concurrent copy for DFSMSShsm control data sets and the journal, use the DATAMOVER parameter on the SETSYS CDSVERSIONBACKUP command. This lets you indicate whether you want to use DFSMSShsm or DFSMSdss (concurrent copy). For more information on using concurrent copy for DFSMSShsm control data sets and the journal, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

To manually call concurrent copy, use the CONCURRENT keyword with the DFSMSdss COPY or DUMP command. CONCURRENT calls concurrent copy only if the data set is eligible.

You cannot use CONCURRENT with the DELETE, UNCATALOG, or RESET keywords.

Figure 66 shows a sample of JCL using the CONCURRENT keyword.

```
//DUMPSTEP EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE DD DSN=USER.BACKUP,UNIT=TAPE,LABEL=(1,SL),
// VOL=SER=(TAPE01,TAPE02,TAPE03),DISP=(,KEEP)
//SYSIN DD *
        DUMP DATASET(INCLUDE(USER.LOG,USER.TABLE,USER.XREF)) -
        OUTDDNAME(TAPE) OPTIMIZE(4) CONCURRENT
/*
```

Figure 66. Invoking Concurrent Copy in JCL

The JCL does a DFSMSdss logical data set dump of three fully qualified data sets using concurrent copy. However, if the data sets are not eligible for concurrent copy, a normal dump is carried out for the data sets.

You can restrict the use of concurrent copy by defining RACF Facility Class profiles with these names:

STGADMIN.ADR.COPY.CNCURRNT

Restricts the use of concurrent copy during DFSMSdss COPY. If the user is not authorized, the request fails.

STGADMIN.ADR.DUMP.CNCURRNT

Restricts the use of concurrent copy during DFSMSdss DUMP.

For information on defining RACF Facility Classes, see Chapter 8, “Protecting Data” on page 145.

Manual Data Backup

You can use DFSMSShsm backup commands to back up data between automatic backup cycles. For system-managed data, the authority to do command backup is determined by the management class definition for the data. The management class command backup attribute allows you to authorize DFSMSShsm command backup for the data associated with the management class. With it, you can use one of these:

- Command backup by only the storage administrator
- Command backup by both the administrator and the user
- No command backup

The storage administrator command for data set backup is BACKDS. The user command for backup is HBACKDS.

The BACKDS and HBACKDS commands back up single data sets to migration level 1 volumes. When either command is issued, DFSMSShsm does not check whether the data set has been changed or met the frequency requirement.

The BACKVOL INCREMENTAL storage administrator command backs up all eligible data sets on a volume. This command can be used to back up data sets on a volume not normally backed up by automatic DFSMSShsm incremental backup. You might want to use this command for a volume when it is first added to a storage group. This lets you control the run time of DFSMSShsm backup when it first processes this volume. After the original backup, DFSMSShsm can then do automatic incremental backup.

We do not recommend using DFSMSdss DUMP to back up data sets because of the amount of tracking required. However, you can use DFSMSdss DUMP to back up by volume. For more information, see *MVS/ESA SML: Managing Storage Groups*.

You can use ISMF to perform DFSMSShsm and DFSMSdss backup and recovery tasks for data. We recommend that you use ISMF as a supplement to automatic data set and volume backup. The commands available are:

- DFSMSdss
 - DUMP
 - RESTORE
- DFSMSShsm
 - HBACKDS
 - HRECOVER
 - HALTERDS (for non-system-managed data sets)

Using ISMF panels, you can construct a filtered list of data and perform tasks against its entries. If you use ISMF to issue the DFSMSShsm command HBACKDS, you gain the advantages of both filtering (with ISMF) and automatic inventory (with DFSMSShsm).

Inline Backup

Inline backup lets you create DFSMSHsm backups from a batch environment. This enables you to overcome the problems of executing the Terminal Monitor Program (TMP) in the background to issue HBACKDS commands. For example, with TMP, the HBACKDS command requires the use of a fully-qualified data set name so that a relative generation number cannot be used to identify a particular GDS. Inline backups also help overcome backup failures caused by job streams that normally serialize data sets.

Inline backup enables you to back up data sets during processing in a batch job. You add an extra step to a batch job, and name the inline backup module, ARCINBAK, as the program. The data sets you want backed up are identified by specially named JCL DD statements. Inline backup extracts the data set names associated with these ddnames, and then performs the backup. The results of each data set backup operation can be written to an output data set.

You cannot use inline backup with:

- Uncataloged data sets
- VSAM data sets currently opened for processing by another task at the time of inline backup processing

Backing Up Critical Data

This section presents methods to ensure the availability of different critical system data.

For extremely critical data sets, you might consider physically isolating them in a separate storage group. *MVS/ESA SML: Managing Storage Groups* discusses criteria for establishing storage groups.

You can also use the sysplex configuration to provide a high level of system availability. Instead of the typical paired processor backup configuration, the new sysplex configuration supports a large number of system images, with each processor serving as a backup for all the others. Therefore, the sysplex can significantly reduce the effects of an outage. In addition, features of the new sysplex enable failed systems and subsystems from shared resources to be contained. For detailed information on the enhanced sysplex configuration, see *System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism* and *System/390 MVS Sysplex Hardware and Software Migration*.

Automatic Critical Data Backup

You specify automatic backup of critical system data in different ways. If you use management class, be sure to use the AUTO BACKUP attribute in both the management class definition and the storage group definition. If you want to use the dual copy feature of the 3990 control units, use the AVAILABILITY=CONTINUOUS attribute in storage class.

Catalogs

Data accessibility in the DFSMS environment depends mainly on the integrity of the catalog because:

- All system-managed DASD data sets must be cataloged
- The names of the SMS classes associated with system-managed data are stored in the catalog

You want to avoid catalog outages and to be able to recover quickly should one occur.

Use the management class AUTOBACKUP attribute to select automatic DFSMSShsm backup of catalogs on system-managed volumes. If you have devices that support the dual copy feature of the 3990 storage control, you can also use the storage class AVAILABILITY=CONTINUOUS attribute to ensure catalog availability.

Well-planned catalog configurations can also improve catalog availability. For example, because certain applications or user groups have higher availability needs than others, you can reduce the effect of a single catalog outage by placing the data in separate catalogs. You can also make sure that critical online applications do not share a catalog with testing, and catalogs that have high availability objectives should be placed on units having alternate path access.

The catalog configuration should include an alternate master catalog, which provides an easy way to recover the master catalog if it is damaged. You can then IPL the system without first having to restore the catalog.

DFSMSShsm Control Data Sets

For DFSMSShsm control data sets, you can improve availability by following the recommendations below:

- Place the journal data set and the control data sets on different head disc assemblies (HDAs) to decrease the chances of a single hardware failure destroying both versions of the data.
- Do not place the MCDS and BCDS on the same volume.
- Use DFSMSShsm journaling to record transactions up to the point of failure. You then recover the control data sets by reapplying the saved records to rebuild the control data set to the point before failure.
- Use a 3990 cached storage control for the control data sets and journal to provide a significant performance benefit. This is especially beneficial if you are running DFSMSShsm in a multihost environment where buffers are always refreshed.

Using dual copy might cause a performance overhead because when DFSMSShsm performs an I/O on a CDS, the record has to be destaged to DASD before the next I/O operation. If you use DASD fast write on journal data sets without the use of dual copy, DFSMSShsm does not have to wait for a destage before the next I/O operation begins.

- Maintain multiple versions of the control data sets and journal data set. Using the CDSVERSIONBACKUP parameter on the SETSYS command, you can decide the number of versions to maintain. The control data sets and journal data set are automatically backed up as part of the DFSMSShsm backup cycle.

Under DFSMS/MVS, when you use SETSYS CDSVERSIONBACKUP, DFSMSdss backs up the control data sets for improved performance of control data set version backup. For more information on backing up the control data sets, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

- Back up the control data sets and journal data set to DASD to provide the fastest recovery.
- Use concurrent copy to back up the control data sets. Using concurrent copy significantly improves DFSMSShsm function availability during CDS version backup. For more information on concurrent copy, see “Concurrent Copy” on page 104.

You can also place DFSMSShsm control data sets on a device that supports the dual copy feature of the 3990 control unit, and use the storage class AVAILABILITY=CONTINUOUS attribute. Note that duplexing is different from journaling, and you should not replace journaling with duplexing. Duplexing records the same data in two places to provide protection against hardware failure. Journaling provides protection against accidental loss of critical data by allowing you to reconstruct it using log contents.

We recommend placing the backup data on a different volume than the primary volume for both duplexing and journaling to decrease the chances of a single hardware failure destroying both versions of the data. Both duplexing and journaling can provide the most complete form of backup but at an increased overhead for the application or service, as previously mentioned.

DFSMSrmm Control Data Sets

The DFSMSrmm control data set contains the complete inventory of your removable media library. It also records changes made to the inventory, such as additions or deletions of volumes.

The DFSMSrmm journal data set records individual updates your users make to the control data set. Like the DFSMSShsm journal data set, the DFSMSrmm journal data set is used to apply to the backup copy of the control data set for recovery.

We recommend that you back up the DFSMSrmm control data set and journal data set daily. You can apply the rules for placing and backing up DFSMSShsm control data sets to DFSMSrmm control data sets. See “DFSMSShsm Control Data Sets” on page 109 for the recommendations.

SMS Control Data Sets

For SMS control data sets, use DFSMSShsm to provide automatic backup of source control data sets (SCDS) and the ACS source data sets. If these data sets do not reside on volumes processed by DFSMSShsm automatic processing, you need to issue commands to manually create backup versions.

You should allocate spare copies of the ACDS and the COMMDS when you create the originals. Having spare copies provides an easy and quick method of recovering these data sets if there are I/O errors.

For more information on providing backup and recovery for the SMS control data sets, see *DFSMS/MVS V1R2 DFSMSdftp Storage Administration Reference*.

RACF Data Sets

Use the RACF duplex feature to ensure RACF data set availability. With duplexing, RACF maintains two functionally identical, active copies of the critical data by repeating each primary update operation to the backup. Any failure of the original copy is resolved by a dynamic switch to the backup, using the TSO RVARY command. To ensure availability, the backup data set must not reside on the same control unit as the original data set.

If it is important to your installation to have valid statistics at all times, you can resynchronize the duplexed backup data set with the original. Otherwise, you do not need to resynchronize the data sets.

An alternative to duplexing is to place the RACF data set on a dual copy device. To use dual copy for system-managed volumes, you specify the storage class AVAILABILITY=CONTINUOUS attribute.

In addition to duplexing and dual copy, back up the RACF data set weekly. This is a safety precaution because the RACF data set is essential to the security of your installation. If, for example, a logical error within the original data set were inadvertently recorded on the duplexed backup, both would become unusable. To recover from this situation, provide a secondary backup mechanism.

Because updates to the original RACF data set are not permitted during backup, perform the backup during a period of low activity.

Another way you can improve data availability is by using multiple RACF data sets to spread access across devices. This reduces the number of resources made unavailable by the loss of one data set or device.

Switchable RACF data sets also improve availability because the facility provides a backup data set that you can switch to, without an IPL, in the event of a failure on an original RACF data set.

RACF allocates the backup data sets at the same time as the original data sets; therefore, the backup data sets must be online during RACF initialization.

Critical Database Data

Certain critical database data can benefit from the dual copy feature of SMS. Figure 67 lists the types of database data that would benefit from the use of storage class with AVAILABILITY=CONTINUOUS, and the dual copy feature of 3990 storage control:

DB2	IMS	CICS
Directory DB2 catalogs	Restart control data sets (RECON)	CAVM control and message queue data sets

Figure 67. Using Hardware Dual Copy for Critical Database Data

Because some data sets in a database system are critical, they require instantaneous recovery facilities. Such facilities provide recovery from a logical error rather than a hardware failure. To recover, the database system continuously builds a duplicate copy by means of a software duplexing mechanism. DB2 and IMS provide such facilities for database management system data sets.

DB2 performs software duplexing on the following database data sets:

- Active logs
- Archive logs
- Bootstrap data sets (BSDS)

For more information, see *DB2 Administration Guide*.

IMS performs software duplexing on the following database data sets:

- Message Queues (3 OSAM Data Sets)
- Restart Data Set
- RECON Data Set (3 VSAM KSDSs)
- Online Data Set—3 to 5 pairs
- Write Ahead Data Set

For more information, see *IMS/ESA Database Administration Guide*.

Using SMS storage class with the 3990 storage control unit provides hardware duplexing. For critical data, we recommend that you use hardware duplexing in addition to the software duplexing capabilities available to provide increased protection.

A critical DB2 database that requires a different backup method is the optical configuration database. This database contains all the optical storage hardware definitions.

To ensure the availability of the optical configuration database, make an image copy when creating each table space in the configuration database and all the table spaces in each of the object storage databases. For more information on how to back up and recover optical configuration databases, see *DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Manual Critical Data Backup

This section presents some recommendations and considerations for backing up data manually.

Catalogs

To manually back up catalogs, use the following commands:

- DFSMSHsm BACKDS
- DFSMSdss DUMP
- IDCAMS EXPORT

We recommend using DFSMSHsm (which invokes IDCAMS) and IDCAMS to manually backup catalogs. These products enqueue on the catalog to prevent it from being updated during backup or recovery.

Which command you use for backing up catalogs depends on the following considerations for recovery:

- If you use DFSMSdss or DFSMSHsm, you must recover the catalog to a volume that has the same volume serial number and device type as the volume from which the catalog was dumped or backed up. If you want to recover a catalog to a volume with a different volume serial number, use IDCAMS directly.

- Through the use of IDCAMS, you can provide catalog locking and unlocking with the IDCAMS ALTER command and the LOCK or UNLOCK parameters. The locking mechanism restricts access to the catalog and allows online systems to continue operations while a catalog is being recovered. You can also use locking to prevent activity against a catalog by another processor in a multihost environment during backup or recovery of the catalog. The person responsible for recovery of catalogs needs RACF ALTER access to the catalogs.

When DFSMSdss, DFSMSHsm, or IDCAMS are used to back up and recover a user catalog, the aliases defined to that catalog are maintained. Specifying the keyword ALIAS on the IDCAMS IMPORT command causes IDCAMS to define the catalog with aliases.

Control Data Sets

If you do not want to wait until the DFSMSHsm backup cycle runs, you can issue the DFSMSHsm storage administrator command BACKVOL to back up DFSMSHsm control data sets (for example, BACKVOL CONTROLDATASETS).

The journal data set contains updates to the control data sets. If a control data set is damaged, you can recover it to its current status by using programs and procedures documented in *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*. These programs help rebuild a CDS by merging a CDS backup and an ordered journal.

If you have DFSMS/MVS installed and you need to back up DFSMSrmm control data sets manually, you can specify the BACKUP parameter when running the inventory management utility, EDGHSKP. You can also invoke the EDGBKUP utility to back up the control data set independently of the DFSMSrmm subsystem. For more information on using these utilities to back up DFSMSrmm control data sets, see *DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide*.

For SMS control data sets, you can use IDCAMS EXPORT, DFSMSdss DUMP, or DFSMSHsm BACKDS commands to provide backup for SCDS. Use DFSMSdss or DFSMSHsm for the ACS source data sets.

For more information on providing backup and recovery for the SMS control data sets, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Archiving Data and Storing Vital Records

This section describes ways you can archive data and store vital records. It also explains how you can use DFSMSrmm, a functional component of DFSMS/MVS, to manage vital records.

Archives

Archiving is the process of copying an active data set to produce an inactive version of it (usually on tape) and then deleting the original, active version. The main purpose of archiving data sets is to provide a long-term storage location for data unlikely to be referenced.

Which data sets can be archived, when they can be archived, and who will archive them (you or the users) are subjects for negotiation with your user groups. Which data sets are archived and how long they can go unreferenced before becoming archive candidates are concerns that can vary according to how the data is used and how critical it is. For example, you might delete data sets containing program object code instead of archiving them. You should consider setting up naming conventions for low-level qualifiers that map to archiving requirements. "Developing Naming Conventions" on page 24 discusses naming techniques.

Using DFSMSdss to Archive Data Sets

You can use DFSMSdss to archive user data sets from any storage group. Schedule DFSMSdss jobs based on the date of last reference and a combination of the following DFSMSdss filtering criteria:

- Data set name
- Creation date, absolute or relative
- Expiration date, absolute or relative
- Last-referenced date, absolute or relative
- Data set organization (if appropriate)

To archive data sets using DFSMSdss, use the DUMP command with the following parameters:

- DATASET** specifies a data set DUMP operation.
- OUTDDNAME** specifies the name of the DD statement that identifies the output data set.
- INCLUDE** specifies the partially- or fully-qualified data set name of the data sets you want to archive.
- EXCLUDE** excludes specific data sets by name.
- BY** specifies additional filtering criteria.
- DELETE** deletes the original data sets.

If you specify the INDDname or INDYname parameter, you must specify the volumes on which the data sets reside. If you omit these parameters, the data sets will be located through the catalog, which is the preferred method for system-managed storage.

For example, to archive all data sets with a prefix of A.B.C, code the following:

```
DUMP OUTDDNAME (TAPE) -  
  DATASET (INCLUDE(A.B.C.**)) -  
  COMPRESS DELETE
```

Figure 68. Using DFSMSdss to Archive Data Sets

DFSMSdss, however, does not select migrated data sets for the DUMP process. You must recall these data sets before the DUMP.

If DFSMSdss is the only method of archiving you use, retain the DFSMSdss dump output as a record of the dumped data set names, the date of the dump, and the output volume identification. You need this information to recover archived data sets. If your storage administration group is responsible for archiving in the installation, users need to contact you to retrieve data sets archived by DFSMSdss.

Using DFSMSHsm to Archive Data Sets

DFSMSHsm does not support an explicit archive function. However, if you need to retain a copy of a data set to serve as an archive copy, you can use the RETAIN DAYS ONLY BACKUP VERSIONS attribute in the data set's management class. This allows you to keep the most recent DFSMSHsm backup copy of a deleted data set for as long as you specify. This attribute only applies when the data set no longer exists on primary or DFSMSHsm migration level 1 or migration level 2 volumes.

Archiving Object Data

You can use the Item Access Facility (IAFC) archival function to store large amounts of object and coded data on less expensive optical storage media, tape storage, or DASD. IAFC is an economical alternative to computer output microfiche (COM) and paper microfiche, and is more efficient for data retrieval.

Two types of data that can benefit from IAFC are billing data and account transaction data. These data types are usually archived on COM and often require access for customer service or inquiries.

Because you define data in IAFC as OAM objects, treat the IAFC-processed data as objects in the DFSMS environment. To archive the data, you need to assign an appropriate storage class to determine whether the data resides on DASD, optical, tape storage.

For more information on using IAFC, see *Item Access Facility: User's Guide*.

Vital Records

Vital records are copies of data sets that must be maintained to meet an externally-imposed retention requirement (such as a legal requirement). For example, most installations are required to keep tax records for a certain period of time.

Vital records are typically stored for a long time (seven years or longer), and therefore should be stored on tape.

Vital records usually have versions. For example, if you need to keep tax records for 15 years, you would add a new version each year. Once you have a total of 15 versions, you delete the oldest one each time you add a new one.

You need to find out from each user group what its requirements are for vital records, including the names of the data sets that need to be copied, how many versions need to be kept, what the expiration dates for the versions are, and the recovery requirements.

If possible, use a naming convention for the data sets so that you can use a partially-qualified data set name as a filter when you copy them. This makes it easier for you to copy and, if necessary, to recover the datasets.

Using DFSMSdss to Store Vital Records

DFSMSdss and ISMF invocation of DFSMSdss are good tools for storing vital records. We do not recommend using DFSMSHsm to store vital records. When it performs automatic backup, it replaces your vital records with more current backups.

To store vital records using DFSMSdss, dump the active data sets to tape, using the same parameters discussed in “Using DFSMSdss to Archive Data Sets” on page 114, except do not specify DELETE.

For example, suppose the Research and Development department of a corporation is required by law to keep design data for its underwater propulsion jets for seven years. Each time a particular model of jet is enhanced, a new version of design data must be maintained.

The design data for the Model A12 is kept in a group of data sets beginning with RDA12.DESIGN. This model was enhanced. Using the DFSMSdss control input in Figure 69, the storage administrator stores vital records for the new version:

```
DUMP OUTDDNAME(TAPE) -  
      DATASET(INCLUDE(RDA12.DESIGN.**)) -  
      COMPRESS
```

Figure 69. Using DFSMSdss to Store Vital Records

This method locates all of the DASD data sets meeting the filter criteria and copies them from any online volume to tape.

If you use this method, you might want to retain the output from the DFSMSdss dump in a data set inventory containing the dumped data set name, the time of dump, and the output volume identification.

Managing Vital Records with DFSMSrmm

If you have DFSMS/MVS installed, you can use the DFSMSrmm functional component to manage data sets and volumes that are considered vital records. Using vital record specification definitions, you establish policies such as the retention period of the data set or volume, and where the data set or volume should reside.

While the vital record specification is active, the vital record specification management criteria overrides the original class retention and location information defined for the data set or volume. The management criteria in the vital record specification is active until the vital record specification itself expires.

For more information on using DFSMSrmm to manage vital records, see *DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide* and *DFSMS/MVS V1R2 DFSMSrmm Guide and Reference*.

Performing Aggregate Backups

To simplify the control of backup and recovery of critical datasets and applications in the event of a disaster, use DFSMSShsm aggregate backup and recovery support (ABARS) to back up data. ABARS enables you to back up logically related groups of data.

Although we recommend using DFSMSShsm daily incremental backup for backing up data for local recovery because it is both automatic and selective, we also recommend ABARS as the method for providing backups at an application level. For user applications, the backup copy can be used either for local recovery or recovery at a remote site in the event of a disaster.

Aggregate backup is a command- or program-driven function that creates DFSMSShsm ABARS backup copies to tape volumes. These volumes are DFSMSShsm-supported and are normally not mounted or online.

To use aggregate backup, SMS must be active on your system or systems where the backups are created.

This section presents some recommendations for preparing and performing aggregate backup. For more information on ABARS, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Preparing for Aggregate Backup

To prepare for aggregate backups, you need to do the following:

- Define aggregate groups.
- Define SETSYS parameters for aggregate backups.
- Define aggregate backup attributes in management class.

Defining An Aggregate Group

To back up data using ABARS, you must define an aggregate group. An aggregate group is a logical group of data belonging to an application. It consists of the following:

Selection Data Sets

Lists names of data to be processed during aggregate backup.

Instruction Data Set

Contains instructions and other information for recovering the application.

Control Information

Describes what data is backed up and what tape is used for the backup. This information is mainly used during the recovery.

You can define up to five selection data sets and an optional instruction data set to an aggregate group. When identifying data to be backed up, you can list them all in a single selection data set, or distribute them among as many as five selection data sets.

The instruction data set and those data sets listed in selection data sets are included in the aggregate backup copy on tape.

Work with your users to identify data sets that are critical to the recovery of their applications, and to determine whether the data sets need to be grouped in any particular way for recovery. After the application's requirements have been established, you can define an aggregate group for its data sets. Remember to activate the SMS control data set each time you define or change an aggregate group. If only the contents of the selection data sets have been modified, you do not need to activate the SMS configuration.

You use using the ISMF Aggregate Group Application when defining an aggregate group. The ISMF Aggregate Group Application enables you to display, define, alter, or delete aggregate groups. Figure 70 shows an example of the Aggregate Group Application panel.

```

                                AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===>

TO PERFORM AGGREGATE GROUP OPERATIONS, SPECIFY:

CDS NAME          ===> 'STGADMIN.SCDS1'
                   (1 to 44 character data set name or 'ACTIVE'
AGGREGATE GROUP NAME ===> PAYROLL1 (for Aggregate Group List, fully or
                                   partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS  ===> 3

1  LIST    - Generate a list of Aggregate Groups
2  DISPLAY - Display an Aggregate Group
3  DEFINE  - Define an Aggregate Group
4  ALTER   - Alter an Aggregate Group
5  ABACKUP - Backup an Aggregate Group
6  ARECOVER - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY VIEW CRITERIA      ===> N   (Y OR N)
RESPECIFY SORT CRITERIA     ===> N   (Y OR N)

USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 70. Defining an Aggregate Group

When using ISMF to define aggregate groups, you can specify the number of copies of the ABARS output files to create. You can also refer to the management class whose attributes and values are to apply to the output files. Figure 71 on page 119 shows the Aggregate Group Define panel with sample values specified.

```

                                AGGREGATE GROUP DEFINE                                Page 1 of 2
COMMAND ===>

SCDS NAME:                STGADMIN.SCDs1
AGGREGATE GROUP NAME: PAYROLL1

TO DEFINE AGGREGATE GROUP, SPECIFY:

DESCRIPTION ===> PAYROLL1 Aggregate Group backup attributes
                ===>

BACKUP ATTRIBUTES

  NUMBER OF COPIES        ===> 2          (1 TO 15)
  MANAGEMENT CLASS NAME   ===> PAYROLL1   (1 to 8 characters, to be
                                                defined in current SCDS)

OUTPUT DATA SET NAME PREFIX ===>
                                                (1 to 33 characters)

USE ENTER TO PERFORM VERIFICATION; USE DOWN COMMAND TO VIEW NEXT PANEL
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

```

Figure 71. Defining Aggregate Backup Attributes in DFSMS/MVS

For information on how to manage the output data, see “Managing ABARS Output Files” on page 126.

Creating an Instruction Data Set: The instruction data set is important if you want to ensure that the application is recovered to an environment similar to the one in which it normally runs and is backed up. Creating an instruction data set is optional, but we recommend creating one for each aggregate group you define. The information you provide in the instruction data set would be especially useful if recovery must be performed by people who did not perform the backup.

Include the following type of information in the instruction data set:

- A description of the SMS attributes.

Include the names of the SMS data, management, and storage classes and storage groups used by the application’s data sets. You do not need to include the SMS attributes in the instruction data set. With DFSMS/MVS, the SMS attributes associated with the aggregates are kept in the aggregate recovery activity log.
- A comprehensive description of the application.

Include a list of the data set names that should be present on completion of recovery, for the application can run successfully. List all user catalogs and associated aliases that are used by the data sets.
- A description of the RACF requirements for the application data sets.

Include a list of the RACF profile names and the attributes associated with each.
- The software requirements for the environment in which the application runs.

List the program products and their release levels. Include information about the SMS configuration, and names of the control data sets used by SMS, DFSMSHsm, and RACF.

- The hardware requirements for the environment in which the application runs. Include details of any dependencies on storage controls, and the device types and models required.
- Any unique requirements or instructions for running the application once recovery is complete.

When you plan the backup and recovery requirements for the application, consider the following points and include the appropriate information, requirements, and instructions in the instruction data set:

- SMS class names and attributes should be listed at the recovery site during the verification step of aggregate recovery.

This enables the storage administrator at the recovery site to determine which SMS class names are required to recover system-managed data to the same DFSMS environment as the backup site. However, to duplicate the DFSMS environment entirely, SMS classes must be defined at the recovery site with the same attributes as at the backup site.

- System-managed data is recovered as non-system-managed data if SMS is not active on the systems at the recovery site.

If SMS is active and an SMS class exists that matches the data set to be recovered, the data set is assigned the appropriate SMS class through the ACS routines. If storage class is not assigned to the data set, it is then recovered as non-system-managed data set.

- The application might have data set names that are the same as existing data set names at the recovery site.

Although this situation is unlikely, if it occurs, you will have problems with duplicate names during aggregate recovery. Therefore, as part of your disaster recovery planning, ensure that duplicate data set names do not exist at the remote site. Include a request in the instruction data set to have the recovery site run a duplicate data set name check before performing an aggregate recovery. This check is done by issuing the ARECOVER command with the VERIFY parameter, which shows any naming conflicts that might occur during the actual recovery. For more information on resolving data set name conflicts during an aggregate recovery, see “Resolving Data Set Name Conflicts” on page 143.

Creating Selection Data Sets: When you create a selection data set, you can specify different types of data set lists. The type you specify determines which data sets are to be backed up, and how they are to be recovered. The data set list types are as follows:

INCLUDE

Lists the data sets to be copied by aggregate backup to a tape data file that is used by aggregate recovery.

EXCLUDE

Lists the data sets to be specifically excluded from being backed up.

ACCOMPANY

Lists the tape data sets that are to physically accompany the aggregate backup tape data file to the recovery site, and are cataloged as part of an aggregate recovery. These data sets are **not** written to the aggregate backup data file, and the data sets have to be provided either by you or the users to the recovery site.

You can use the DFSMSrmm inventory to identify the tape volume names associated with the accompany data sets. You can also use the DFSMSrmm vital record specifications to manage and track the movement of the accompany data sets and volumes. For information on using DFSMSrmm to manage tape volumes, see *DFSMS/MVS V1R2 DFSMSrmm Guide and Reference*.

ALLOCATE

Lists the DASD data sets to be processed by aggregate backup, and to be pre-allocated at the recovery site. These data sets are specified on the control file to be allocated and cataloged at the recovery site. The data sets are **not** written to the aggregate backup data file.

You can also specify integrated user catalogs in the ALLOCATE list.

ACCOMPANYEXCLUDE

Lists the tape data sets to be excluded from the ACCOMPANY list.

ALLOCATEEXCLUDE

Lists the DASD data sets to be excluded from the ALLOCATE list.

Use ACCOMPANY to list the critical tape data sets to include in the backup processing. Tape data set are processed as long as they are cataloged and match the ACCOMPANY list.

For more information on selection data sets, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

We recommend that you create the selection data set as a PDS or PDSE. Each member in the data set contains one type of data set list, and the member name needs to correspond to the type of list it contains. For example, a member named INCLUDE contains the INCLUDE list. The names help recovery personnel identify the lists easily.

Defining SETSYS Parameters for Aggregate Backup

Use the SETSYS command to define aggregate backup processing options that you want externally specified.

To specify how many ABACKUP commands you want to run concurrently, use the MAXABARSADDRESSSPACE parameter. This determines the number of concurrent DFSMSHsm secondary address spaces that can be supported; the maximum is 15. A secondary address space is used every time you issue the ABACKUP command. When you specify this value, consider such factors as system load, resource contention, aggregate backup and recovery activity, and hardware availability.

The ABARSPROCNAME parameter specifies the name of the procedure used to start the DFSMSHsm secondary address space each time you issue an ABACKUP command.

Each time you issue an ABACKUP command, an activity log is created. You use the ACTLOGTYPE parameter to specify whether you want the activity log automatically written to DASD or SYSOUT.

You can use the ABARSACTLOGTYPE parameter to include the activity log as a separate output file. If ABARSACTLOGTYPE is not specified, it defaults to the values specified in the ACTLOGTYPE parameter.

The ABARSACTLOGMSGLVL parameter allows you to specify whether all DFSMSdss messages or just warning and error messages are written to the ABARS activity log.

The ABARSUNITNAME parameter allows you to supply the default unit to make the UNIT parameter optional when using the aggregate backup and recovery commands.

The EXITON or EXITOFF parameters can be used to indicate whether desired user exits are to be activated for aggregate backup or recovery.

You can specify the number of additional I/O buffers to use during aggregate backup with the ABARSBUFFERS(n) parameter. The number of buffers, n, is multiplied by 2 to arrive at the number of buffers to use for I/O processing.

Figure 72 shows an example of the SETSYS command parameters used for performing aggregate backup processing:

```
SETSYS MAXABARSADDRESSSPACE(1) ABARSPROCNAME(ABARS)
SETSYS ABARSACTLOGTYPE(DASD)
SETSYS ABARSACTLOGMSGLVL(REDUCED)
SETSYS ABARSUNITNAME(3490)
SETSYS EXITON(M2,ED,BE)
SETSYS ABARSBUFFERS(4)
```

Figure 72. Using SETSYS to Perform Aggregate Backup

For a full list and a description of the SETSYS parameters used for aggregate backups, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Issuing the ABACKUP Command

This section describes how you can verify backup contents, perform the aggregate backup, use the ISMF Aggregate Group Application for aggregate backup, and minimize data set contention.

When you perform aggregate backups, ABACKUP produces output files used for recovery. *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment* discusses these files and their contents in detail.

Verifying the Backup Contents

You can verify the content of an aggregate backup before you actually perform the backup. Issue the DFSMSShsm ABACKUP command with the VERIFY parameter. This optional test run allows you to identify errors and make sure that the data sets listed are intended for the backup. Figure 73 shows an example of issuing the ABACKUP command with the VERIFY parameter. The aggregate backup verification is performed for the PAYROLL1 aggregate group.

```
ABACKUP PAYROLL1 VERIFY
```

Figure 73. Verifying an Aggregate Backup

The VERIFY parameter also indicates the following:

- Amount of storage (in kilo-, mega-, giga-, or terabytes) needed at the aggregate backup site

VERIFY displays the total storage, and individual totals for primary, migration level 1, and migration level 2 volumes.

- List of required backup tapes

The list is sorted and written to the activity log, to allow the backup site to pull the required tapes prior to issuing the ABACKUP EXECUTE command. The tape list includes non-DFSMSShsm tapes as well as migration level 2 tapes.

For more information on using the ABACKUP VERIFY parameter, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Performing the Aggregate Backup

When you are ready to perform the backup, issue the ABACKUP command with the EXECUTE parameter. The example in Figure 74 shows an ABACKUP command for backing up an aggregate group (PAYROLL1).

```
ABACKUP PAYROLL1 EXECUTE
```

Figure 74. Executing an Aggregate Backup

You can also issue the following parameters with ABACKUP:

- MOVE

Deletes the aggregate backup data sets if the backup completes successfully.

- FILTEROUTPUTDATASET

Directs the list of fully qualified data set names filtered from the INCLUDE, EXCLUDE, ALLOCATE, ALLOCATEEXCLUDE, ACCOMPANY, and ACCOMPANYEXCLUDE data sets to a separate output data set. Also displays a count of the total number of data sets to be backed up.

The DFSMSShsm ABACKUP UNIT parameter is optional and you can specify a default unit when invoking aggregate backup. To do this, use the SETSYS ABARSUNITNAME parameter.

Upon completion of the ABACKUP processing, an ABARS activity record is created and written to the DFSMSHsm BCDS at the backup site. The activity record contains an inventory of aggregate backups and recoveries issued for a particular aggregate group. The record also allows the syntax of the ARECOVER command to be simplified so the VOLUME and UNIT parameters need not be entered each time an aggregate recovery is performed.

For information on using the ABACKUP MOVE, ABACKUP FILTEROUTPUTDATASET, or the SETSYS ABARSUNITNAME parameters, or for a description of the ABARS activity record, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

To automate the aggregate backup process, you can add a job step at the end of an application stream to send an ABACKUP request to DFSMSHsm. For more information on this method of backup, see “Inline Backup” on page 108.

Using the ISMF Aggregate Group Application for Backups

You can use the ISMF Aggregate Group Application to group and back up data sets according to application backup and recovery requirements. The ISMF Aggregate Group Application enables you to display, define, alter, or delete aggregate group attributes.

ISMF provides options on the Aggregate Group Application Selection panel for initiating ABARS commands. Figure 75 shows an example of this panel.

```

                                AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===>

TO PERFORM AGGREGATE GROUP OPERATIONS, SPECIFY:

CDS NAME          ===> 'STGADMIN.SCDS1'
                   (1 to 44 character data set name or 'ACTIVE')
AGGREGATE GROUP NAME ===> PAYROLL1 (for Aggregate Group List, fully or
                                   partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 5

 1 LIST   - Generate a list of Aggregate Groups
 2 DISPLAY - Display an Aggregate Group
 3 DEFINE - Define an Aggregate Group
 4 ALTER  - Alter an Aggregate Group
 5 ABACKUP - Backup an Aggregate Group
 6 ARECOVER - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY VIEW CRITERIA      ===> N   (Y OR N)
RESPECIFY SORT CRITERIA     ===> N   (Y OR N)

USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 75. Performing an Aggregate Backup Using ISMF

When you specify ABACKUP in ISMF, you can specify the type of device to use for the aggregate backup, whether you want to verify or execute the backup, and as a check for errors during the verification process. Under DFSMS/MVS, you can optionally specify the unit type. You can also specify whether to delete data sets after the backup is complete, and provide the name of the data set to contain the

list of data sets filtered from the INCLUDE, EXCLUDE, ALLOCATE, and ACCOMPANY lists. Figure 76 on page 125 shows the options and sample values you can specify in ABACKUP.

```
AGGREGATE GROUP BACKUP                                Page 1 of 2
COMMAND ===>
CDS NAME: STGADMIN.SCDS1
AGGREGATE GROUP NAME ===> PAYROLL1
UNIT NAME      ===>
PROCESSING OPTION ===> 1                (1=VERIFY,2=EXECUTE)
WAIT FOR COMPLETION ===> Y            (Y or N)
DELETE DATA SETS AFTER ABACKUP ===> (Y or N)
FILTER OUTPUT DATA SET NAME (1 to 44 characters)
===>

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 76. Specifying Options During Aggregate Group Backup

The description of the specifications follows:

UNIT NAME

A blank defaults to the device defined in SETSYS ABARSUNITNAME.

PROCESSING OPTION

1 specifies running the backup in VERIFY mode.

WAIT FOR COMPLETION

Y enables you to check for errors encountered during the verification process.

DELETE DATA SETS AFTER ABACKUP

A blank defaults to N, indicating the data sets for the aggregate group will not be deleted following the verify process.

FILTER OUTPUT DATA SET NAME

A blank indicates the data set names filtered from the INCLUDE, EXCLUDE, ALLOCATE, and ACCOMPANY data sets are not written to a separate output data set.

You can also enter the ABACKUP line operator command through the ISMF Aggregate Group Application. For more information on how to use the ISMF Aggregate Group Application, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

For details on all the ABACKUP parameters and the values you can specify, see *DFSMS/MVS V1R2 DFSMSShsm Storage Administration Guide*.

Minimizing Data Set Contention

To minimize data set contention during aggregate backup, follow these recommendations:

- Do **not** issue the ABACKUP command while AUTOMIGRATION, AUTOBACKUP, or AUTODUMP is active.
- Hold MIGRATION, RECALL, and BACKUP functions.
- Do **not** use aggregate backup while the application is in use.

Aggregate backup processing might fail if you do not follow these recommendations. Data sets that were indicated as being migrated at the time the ABACKUP command was issued are prevented from being recalled before aggregate backup processes the data set.

Managing ABARS Output Files

With DFSMS/MVS, you can use management class and DFSMSrmm to manage the output files.

Using Management Class

Figure 77 shows the aggregate backup attributes you can specify in management class. You can specify these attributes when you define management classes.

```
MANAGEMENT CLASS DEFINE                                     Page 5 of 5
COMMAND ==>
SCDS NAME:          STGADMIN.SCDS1
MANAGEMENT CLASS NAME: PAYROLL1

TO DEFINE MANAGEMENT CLASS, SPECIFY:

AGGREGATE BACKUP ATTRIBUTES:

# VERSIONS          ==> 2      (1 to 9999, NOLIMIT or blank)
RETAIN ONLY VERSION ==> 2      (1 to 9999, NOLIMIT or blank)
  UNIT              ==> W      (D=days, W=weeks, M=months, Y=years
                             or blank)
RETAIN EXTRA VERSIONS ==> 12   (1 to 9999, NOLIMIT or blank)
  UNIT              ==> M      (D=days, W=weeks, M=months, Y=years
                             or blank)
COPY SERIALIZATION  ==> C      (C=continue, F=fail or blank)
ABACKUP COPY TECHNIQUE ==> S    (P=Conc Preferred,
                             R=Conc Required or S=Standard)

USE ENTER TO PERFORM VERIFICATION;
USE UP COMMAND TO VIEW PREVIOUS PANEL
USE HELP COMMAND FOR HELP;
USE END COMMAND TO SAVE AND EXIT;
CANCEL TO EXIT.
```

Figure 77. Defining Aggregate Backup Attributes Using Management Class

VERSIONS

Specifies the number of versions to maintain.

RETAIN ONLY VERSION

Specifies how long to retain the only version.

RETAIN EXTRA VERSION

Specifies how long to retain the additional aggregate backup copies.

COPY SERIALIZATION

Indicates whether aggregate backup should continue if an enqueue failure is encountered.

ABACKUP COPY TECHNIQUE

Indicates whether concurrent copy is used with aggregate backup.

If you specify Concurrent Required for ABACKUP COPY TECHNIQUE, all of the data sets in the aggregate group must reside on concurrent copy-capable volumes or the process fails.

The values you assign to aggregate backup attributes should vary by application. We recommend that you retain the backup copies for at least the length of an application's process cycle. For example, for payroll applications that process data twice each month, you might retain the backup copies for two weeks.

In general, we recommend that you use your service-level agreements (SLAs) when you determine the aggregate group backup attribute values. For information on SLAs, see *MVS/ESA SML: Leading a Storage Administration Group*.

Using DFSMSrmm

You can use DFSMSrmm to help you manage the ABARS output volumes. DFSMSrmm can manage expiration processing and track the movement of the tape volumes to different locations. These management features are especially useful if you transport your disaster recovery ABARS tapes to different sites. For more information on using DFSMSrmm to manage your ABARS tapes, see *DFSMS/MVS V1R2 DFSMSrmm Guide and Reference*.

Chapter 7. Recovering Data

Data recovery is the process of rebuilding data after it has been lost, damaged, or corrupted. The recovery process usually consists of copying a backup version of the data and restoring the data from the latest backup. For database data, the recovery process includes reapplying transactions listed in a log to the backup copy.

Many methods are available for recovering data. The method you use depends on how the backup copy was created.

This chapter describes methods you can use to recover data and objectives to address in a recovery plan. It discusses the following tasks:

- Developing a recovery plan
- Recovering data
- Recovering data critical to the installation
- Retrieving vital records and archived data
- Performing aggregate recovery

Developing a Recovery Plan

The purpose of developing a recovery plan is to ensure a smooth recovery process. The following are some general and disaster recovery objectives you need to consider in a recovery plan:

- **Avoid modifications to the JCL.**

By maintaining consistent data set naming conventions and data classes, you should not need to modify JCL to run the applications at the recovery site.

- **Prevent duplication of data names and tape and DASD volume serial numbers.**

Set up the naming conventions so that data set names and tape and DASD volume serial numbers are not duplicated.

- **Ensure the security of recovered data.**

If possible, define RACF profiles at the recovery site ahead of time to provide security for your data sets.

- **Define catalog aliases at the recovery site ahead of time.**

Define the catalog aliases for your installation's data sets at the recovery site ahead of time. Under DFSMS/MVS, you can use DFSMSHsm aggregate backup and recovery support (ABARS) to define required integrated catalogs and aliases.

Define any generation data groups (GDGs) in the catalogs. When using ABARS, you do not need to pre-define GDG bases because aggregate recovery automatically defines them if they do not exist.

- **Make sure the recovery site has enough extra DASD and related hardware.**

The recovery site must have sufficient storage devices to handle your applications. You might also need to check that the performance capability of the

storage subsystem can meet the storage class requirements of your installation's applications.

- **Be prepared to recover the DFSMS environment.**

Either ensure that the SMS classes and storage groups are consistent across sites, or plan to add your classes and storage groups to those at the recovery site. To prepare for recovering the DFSMS environment, ensure that the class and storage group names you use are unique. You also need to modify the ACS routines at the recovery site.

- **Identify the noncritical applications.**

If any of the recovery site's applications need to be rescheduled or cancelled to handle the extra workload, those that are not critical should be identified.

- **Test the recovery plan.**

Test your recovery plan to see if it works. Do not wait until a real disaster to do that. Hold practice drills on a regular basis, at least twice a year. If your installation has a high personnel turnover rate, you need to hold more frequent practice drills.

Recovering User Data

The method you use to recover data depends on the method used when you made backup copies of the data. This section presents the different ways to recover data sets, objects, and database data.

Data Set Recovery

An advantage of using DFSMSHsm for backup and recovery is that data sets can be recovered by users, who usually know first about any problem with the data sets and are most affected by a lengthy recovery process. In addition, users do not need to know where the data set is located.

Note that DFSMSHsm only recovers an extended format data set on a system that supports the extended format. If you need to recover an extended format data set to a system that does not support the format, use the access method services IMPORT command.

Recovery Commands

The user command for recovering a single data set or a list of data sets is HRECOVER. DFSMSHsm selects either an incremental backup version or a dump copy, whichever is more recent. Recovery of a dump copy, however, is done only if the dump was taken as part of a dump class which permits data set recovery. Figure 78 shows the HRECOVER command being specified for the FER1234.ANALYSIS.TEXT data set.

```
HRECOVER 'FER1234.ANALYSIS.TEXT'
```

Figure 78. Using the DFSMSHsm HRECOVER User Command

The storage administrator command for recovering a data set is RECOVER. As with the HRECOVER command, DFSMSHsm recovers the most recent backup copy. For example, Figure 79 on page 131 shows the RECOVER command recovering the most recent copy of the data set named FER1234.ANALYSIS.TEXT.

```
RECOVER FER1234.ANALYSIS.TEXT
```

Figure 79. Using the DFSMSHsm RECOVER Command

With RECOVER, you can optionally request a recovery from a dump copy by specifying the FROMDUMP parameter as shown in Figure 80:

```
RECOVER FER1234.ANALYSIS.TEXT FROMDUMP(DUMPCLASS(WEEKLY))
```

Figure 80. Using the DFSMSHsm RECOVER Command for Physical Data Set Restore

For more information on using the RECOVER command, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Special Considerations For Data Set Recovery

This section describes DFSMSHsm recovery processing for data sets in a DFSMS environment. Processing is described for cases in which DFSMSHsm recovers a data set using a backup copy or a physical dump. DFSMSHsm will delete all existing data sets, including VSAM, non-VSAM, system-managed, or non-system-managed data sets prior to the recovery.

Using an DFSMSHsm Backup Version: If you recover an **existing** data set (currently cataloged), the data set is recovered to the volume (system-managed or non-system-managed) where the existing data set resides.

If you recover a data set that **no longer exists**, ACS routines are invoked to determine the SMS classes and target storage group for the data set. The following are available to the ACS routines:

- The volume serial number of the volume from which the data set was backed up, or the volume serial number specified on the RECOVER command
- An allocation environment indicator of recover

If the data set was system-managed when it was backed up, the SMS classes originally associated with the data set are also available to the ACS routines.

In the ACS routines, you can determine whether the original SMS classes or new classes should be assigned to the data set. For example, you could use the code shown in Figure 81 in a management class routine to determine if a management class is already associated with a data set, and if so, to reassign it:

```
WHEN (&ACSENVIR = 'RECOVER' && &MGMTCLAS ->= ' ')  
SET &MGMTCLAS = &MGMTCLAS
```

Figure 81. Determining if Management Class is Assigned to Recovered Data

If you recover an existing or non-existing system-managed data set using a DFSMSHsm incremental backup version, DFSMSHsm will allow the system to select the target volume.

Using a Dump Copy: If an **existing** data set (currently cataloged) is being recovered, DFSMSHsm invokes DFSMSDss physical data set restore with the REPLACE parameter. DFSMSDss attempts to restore the backup data on the volume that contained the preallocated data set.

If a data set **no longer exists**, the same restore procedure is used but REPLACE is not specified.

When DFSMSHsm invokes a DFSMSdss physical data set restore for a VSAM data set to a system-managed volume, the data set must exist on the volume. To ensure this, you can preallocate space for the data set. If the VSAM data set no longer exists and a DFSMSHsm backup version of the data set exists, DFSMSHsm recovers the data set itself using the backup version. The recovery process is then the same as that used for an existing system-managed data set, except that the recovery fails if the existing space on the volume is too small.

Recovery to a Non-System-Managed Volume: If SMS is not active, or if you need to recover a data set to non-system-managed storage, you can use the FORCENONSMS parameter on the RECOVER command. Also specify a volume that is not managed by SMS on the TOVOLUME parameter.

Volume Recovery: You should use DFSMSHsm full volume restore (which invokes DFSMSdss RESTORE) or DFSMSHsm incremental volume recovery with the FROMDUMP APPLYINCREMENTAL option for volume recovery. By using a combination of the two, you can provide a more accurate reconstruction of volume contents. For more information on this method of volume recovery, see *MVS/ESA SML: Managing Storage Groups*.

Database Data Recovery

To recover database data backed up by DFSMSHsm automatic backup, issue the HRECOVER command. When you recover DB2 data, you also need to reconcile the DB2 catalog to maintain catalog consistency. You may also want to use the DB2 RECOVER utility to update that copy to a more current state, using change data from the DB2 log.

To recover data that was backed up while open, you need to use DFSMSdss RESTORE to retrieve the latest version. The database utilities are then used to apply the recovery logs to bring the backup copy up-to-date. During recovery, the Database Manager prevents access to the data.

For information on using backup-while-open to back up database data, see “Backup-While-Open” on page 103. For more information on backup-while-open, see *DFSMS/MVS V1R2 Using Advanced Services for Data Sets*.

Object Data Recovery

If an object becomes unreadable, you can recover the object from its backup copy. Figure 82 shows the OAM operator command to use for recovery:

```
F OAM,START,OBJRECV,collection_name,object_name
```

Figure 82. Recovering Object Data

You can also perform an Optical Disk recovery to recover objects on both sides of an optical disk from the backup copies. Alternatively, you can recover an entire backup optical disk by creating a new backup volume using the primary copies.

Figure 83 on page 133 shows the command to use if you want to create a new backup from primary copies.

```
F OAM,START,RECOVERY,'VOL_SER'
```

Figure 83. Creating New Object Backup from Primary Copies

You can also invoke Optical Disk Recovery through ISMF. For information on using ISMF, see *DFSMS/MVS V1R2 Using ISMF*.

For more information on backing up and recovering objects, see *DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Recovering Data Critical to the Installation

This section presents tasks you perform when you need to manually recover critical data.

Catalog Recovery

If you recover a catalog to a different volume, check the catalog for VSAM data sets that have discrete RACF profiles. The RACF profile for these data sets contains the serial number of the volume containing the catalog. If the profile is not changed, RACF fails any attempts to access the data set. You can avoid this problem by using RACF generic profiles. For information on RACF profiles, see Chapter 8, “Protecting Data” on page 145.

To recover a catalog to a specific system-managed volume, use an SMS storage class with the GUARANTEED SPACE attribute. With GUARANTEED SPACE, SMS honors the specific volume allocation, provided the volume is defined to the storage group (or groups) selected by ACS routines.

When you use IDCAMS to recover a catalog, you can define the catalog with aliases by specifying the ALIAS keyword on the IDCAMS IMPORT command. When you use DFSMSdss or DFSMSHsm to recover a user catalog, the aliases defined to that catalog are maintained.

If you use IDCAMS to recover to an empty target catalog, you need to specify the INTOEMPTY parameter on the IMPORT command. If you use DFSMSHsm for the recovery, DFSMSHsm determines if the target catalog is empty as part of its processing. With both IDCAMS and DFSMSHsm, any existing aliases for the catalog are maintained. Any aliases that were backed up, but not defined for the catalog, are redefined.

For more information on the recovery of integrated catalogs, see *DFSMS/MVS V1R2 Managing Catalogs*.

Reconciling Catalog Entries

After the recovery of a catalog, you need to reconcile the entries in the catalog with the current data set status.

On system-managed volumes, the VVDS includes entries for VSAM permanent and temporary data sets and for permanent non-VSAM data sets. When you recover catalogs that contain entries for system-managed data sets, you need to reconcile the catalog entries with VVDS entries. You can use the DIAGNOSE command to check for inconsistencies.

Combined with the diagnosis, you can use the Integrated Catalog Forward Recovery Utility (ICFRU, program number 5785-DXQ) to do a forward recovery of the catalog. This utility uses a backup copy of the catalog and SMF records of catalog changes to restore the catalog to its status at the time of failure. For more information, see *ICFRU Program Description/Operations Manual*.

Analyzing Effects of Catalog Outage

To determine which users or applications would be affected and how long recovery might take for a damaged catalog, we recommend you perform an outage analysis for critical catalogs. If a damaged catalog contains non-VSAM entries for many volumes, processing the VTOCs or SMF data to determine appropriate actions might require longer than the tolerable recovery time. The same is true for VSAM data sets in terms of the time required to use the DIAGNOSE command and to perform the indicated recovery actions.

Control Data Set Recovery

Control data sets under different subsystems require different recovery methods.

DFSMSHsm Control Data Set

To recover DFSMSHsm control data sets, you can use the DFSMSHsm system programmer command UPDATEC. UPDATEC combines the entries in the journal data set with the restored backup version of the control data set to recover the data sets to the point of failure.

UPDATEC can require extensive run time. As the process holds an exclusive enqueue and reserve on all the control data sets, no other DFSMSHsm functions (such as RECALL) can run. When this happens, a complex will not be able to do any work until UPDATEC is completed. In a JES3 environment, the entire complex might be unavailable until UPDATEC is finished.

To improve the availability of your complex and provide a faster recovery of the control data sets, you can use the backup version made by AMS EXPORT, and the ARCBJRN and ARCIMPRT utilities. The ARCBJRN utility makes a copy of the journal data set, which can be sorted. The ARCIMPRT utility merges the sorted journal data set with the backup copy of the control data set.

For detailed information on recovering DFSMSHsm control data sets using the utilities or the UPDATEC command, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

If you need to recover a DFSMSHsm control data set with a broken index (because of hardware defects), extract records from a VSAM data component, then use DFSORT to repair and reload the control data. For more information on this

process, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

DFSMSrmm Control Data Sets

Use the DFSMSrmm utility, EDGBKUP, with the RESTORE parameter to recover the DFSMSrmm control data set. During the recovery process, the control data set is restored from a backup copy, and the current journal is applied to recover the data set to the point of failure.

For detailed information on recovering DFSMSrmm control data sets, see *DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide*.

SMS Control Data Sets

For SMS control data sets, use IDCAMS IMPORT, DFSMSdss RESTORE, or DFSMSShsm RECOVER to provide recovery for the SCDS. Use DFSMSdss RESTORE or DFSMSShsm RECOVER only for the ACS source data sets.

RACF Data Set Recovery

An error condition in the data set is usually a sign that the RACF data set needs to be repaired or recovered. If an error occurs, you can issue the RVARY command to deactivate the RACF data set and use the RACF data set verification utility program (ICHUT200) and the block update utility command (BLKUPD) for repair.

Critical Database Data Recovery

We recommend that you place most critical database management system data sets under the control of SMS to benefit from hardware dual copy and automatic recovery. For more information using the dual copy feature of the 3990 control units, see Chapter 6, “Backing Up Data” on page 93.

For optical configuration databases, you can use the DB2 recovery log for recovery. Each change made to the database is recorded in the recovery log. For recovery, DB2 can merge incremental image copies of the table space with the full image. It then applies the recovery log to recover the table space to its last point of committed change, thus ensuring data integrity.

For more information about recovering DB2 table spaces, see *IBM DATABASE 2 Version 3 Administration Guide*.

Retrieving Vital Records and Archived Data

To retrieve vital records or archived data stored with DFSMSdss, use the DFSMSdss RESTORE command. For example, to retrieve all data sets with a prefix of A.B.C, use the code in Figure 84:

```
RESTORE INDDNAME (TAPE) -  
        DATASET (INCLUDE(A.B.C.**))
```

Figure 84. Using DFSMSdss to Retrieve Data Sets

For more information on using DFSMSdss RESTORE, see *DFSMS/MVS V1R2 DFSMSdss Storage Administration Reference*.

To retrieve data archived with IAFC, use the IAFC retrieval function. The IAFC retrieval function enables you to build search criteria, retrieve the data, and display the record on your screen. For more information on using IAFC, see *Item Access Facility: User's Guide*.

Performing Aggregate Recovery

We recommend that you use the DFSMSHsm ABARS function for both the backup and recovery of data sets that are critical to your users' applications. You can only use aggregate recovery if the data sets were backed up by the aggregate backup process. SMS does not need to be active for aggregate recovery unless you need to recover PDSEs, striped sequential data sets, or if you specified VSAM data sets in the ALLOCATE list during the aggregate backup.

This section explains how to prepare for aggregate recovery and how to perform the recovery. For detailed information, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Preparing for Aggregate Recovery

Before performing aggregate recovery, you need to:

1. Verify receipt of the aggregate backup tapes containing the control, data, and instruction/activity log files.
2. Retrieve the instruction file for information and instructions for the recovery site.
3. Retrieve the ABACKUP activity log.
4. Determine the recovery site constraints.
5. Define target DASD volumes for the aggregate recovery.
6. Determine the required SETSYS parameters.
7. Define the model entity to define discreet RACF profiles (optional).
8. Link-edit the aggregate recovery user exits (optional).

The following are recommendations for when you prepare for aggregate recovery.

Retrieve Recovery Instructions and Activity Log

The instruction file contains recovery instructions. It is optionally created for an aggregate backup. Retrieve the instruction file before you begin aggregate recovery.

You can issue the ARECOVER command, with the PREPARE and INSTRUCTION keywords, to retrieve the instruction data set. You can also retrieve the aggregate backup activity log using the ARECOVER command with the PREPARE and ACTIVITY keywords. Figure 85 on page 137 shows an example of issuing the ARECOVER command with the PREPARE, INSTRUMENT, and ACTIVITY keywords.

```
ARECOVER DATASETNAME(PAYROLL1.C.C01V0001) UNIT(3480)
VOLUME(TAPE10,TAPE11) PREPARE INSTRUCT ACTIVITY
```

Figure 85. Retrieving the Instruction Data Set and the Activity Log

The PREPARE keyword causes the following to occur:

- Creates or updates the ABARS activity record. The ABARS activity record must exist at the recovery site to maintain an inventory of aggregate recovery activity and to support the simplified ARECOVER command syntax.
- Catalogs the control, data, and instruction/activity log files.
- Defines the ABARS tapes to RACF at the recovery site. Volumes are added to RACF HSMABR TAPEVOL tape volume set if required.
- Writes storage requirement messages to the ARECOVER activity log.
- Recovers the instruction data set and aggregate backup activity log (if INSTRUCT and ACTIVITY are issued).

For more information on recovering the instruction file and the activity log, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Determine Recovery Site Constraints

When you determine the recovery site constraints, find out the DASD and tape devices needed to recover the control and application data sets, and the users' tape data sets. Establish this information as part of your disaster backup and recovery planning, and include the information in the instruction file at the backup site, which is backed up as part of the aggregate backup process. This information also exists in the ABACKUP activity log. If the activity log is written to DASD, it is also backed up as part of the ABACKUP output.

How much DASD space you require for recovery should also have been determined as part of the disaster backup and recovery planning. However, the amount of DASD space will be determined by the disposition of the data sets during the aggregate backup process, and by the DASD device types installed at the recovery site. You can obtain the space requirements information from the ABACKUP activity log, or from the DFSMSshm LIST command if the ARECOVER command is run with the PREPARE keyword.

The tape devices used for recovery of the control and data files must be compatible with the tape devices used for aggregate backup.

Define Target DASD Volumes

To specify a target pool of volumes to be used for the data sets to be recovered by the aggregate recovery process, you use the DEFINE ARPOOL command. This enables you to determine the target DASD primary and migration level 1 volumes on which the data sets will be recovered. You can include the DEFINE command in the DFSMSshm PARMLIB member ARCCMDxx to ensure that the settings are defined to DFSMSshm at startup initialization.

If you are an authorized DFSMSShsm user, you can issue the DEFINE command before an aggregate recovery to modify existing pool configurations or add new pools. You can define pools of DASD volumes that are used for the recovery of both non-system-managed and DFSMSShsm-migrated data sets.

Figure 86 shows a DEFINE ARPOOL command defining a target pool to be used for aggregate recovery:

```
DEFINE ARPOOL (PAYROLL1 ML1VOLS(MIG102) L0VOLS(PRM002))
```

Figure 86. Defining Target Pools for Aggregate Recovery

In Figure 86, the command specifies that the migration volume, MIG102, and the primary volume, PRM002, are to be used for the recovery of data sets from the aggregate group PAYROLL1.

Non-system-managed data sets originally residing on primary volumes are targeted to a volume defined with the L0VOLS parameter. System-managed data sets originally residing on primary volumes are allocated to system-managed volumes by ACS routines if SMS is active on the systems at the recovery site.

System-managed volumes are managed separately from the pools defined using the DEFINE ARPOOL command. If SMS is not active at the recovery site, or if SMS is active but no ACS routines exist to direct the data sets to system-managed volumes, the data sets are recovered to one of the primary volumes specified in the DFSMSShsm DEFINE ARPOOL command.

Data sets that were backed up from migration volumes are targeted to volumes defined with the ML1VOLS parameter, unless the installation directs migrated data to migration level 2 tape using the MIGRATEDDATA parameter of the ARECOVER command.

If an ARPOOL has not been defined for an aggregate, then ARECOVER uses currently ADDVOLed primary volumes and migration level 1 volumes as target volumes for non-system-managed and migrated data sets.

Defining the SETSYS Parameters for Aggregate Recovery

If you are an authorized DFSMSShsm user, you can issue the SETSYS command before aggregate recovery to modify the SETSYS parameters that control the recovery processing.

To specify how many ARECOVER commands can run concurrently, use the MAXABARSADDRESSSPACE parameter. This determines the maximum number of secondary address spaces DFSMSShsm requires to support the number of ARECOVER commands you issue. Consider factors such as system load, resource contention, aggregate recovery activity, and hardware availability when you specify the MAXABARSADDRESSSPACE value.

The ABARSPROCNAME parameter specifies the procedure name that is used to start the DFSMSShsm secondary address space each time you issue the ARECOVER command.

To specify the default tape unit name for user tape data sets being recovered to a tape device during recovery processing, use the ARECOVERUNITNAME parameter. If the TARGETUNIT parameter is specified on the ARECOVER command, the value specified takes precedence over any value you specify for the ARECOVERUNITNAME parameter. If neither parameter is specified, the default is 3400-6.

Each time you issue the ARECOVER command, an entry is made in an activity log. Specify whether you want the log written to SYSOUT or DASD by using the SETSYS ACTLOGTYPE parameter.

ARECOVERUNITNAME specifies the default tape unit name for tape data sets being recovered to a tape unit during recovery processing.

The ARECOVERPERCENTUTILIZED parameter establishes the percentage full DFSMSHsm will fill primary volumes when recovering data sets in a non-system-managed environment.

You can use the ABARSACTLOGTYPE parameter to include the activity log as a separate output file. If ABARSACTLOGTYPE is not specified, it defaults to the values specified in the ACTLOGTYPE parameter.

The ARECOVERML2UNIT parameter specifies the unit name to use when recovering migrated data sets to non-specific migration level 2 tape volumes.

The OPTIMUMDASDBLOCKING and NOOPTIMUMDASDBLOCKING parameters specify whether DFSMSHsm should use its defined optimum block size when restoring data sets to migration level 1 DASD.

The SETSYS parameters described in “Performing Aggregate Backups” on page 117 can also be used to control aggregate recovery.

Figure 87 shows some sample SETSYS commands you can enter before performing aggregate recovery:

```
SETSYS MAXABARSADDRESSSPACE(5)
SETSYS ABARSPROCNAME(ABARS)
SETSYS ABARSACTLOGTYPE(DASD)
SETSYS ARECOVERUNITNAME(3490)
SETSYS OPTIMUMDASDBLOCKING
SETSYS ABARSACTLOGMSGLVL(REduced)
SETSYS ABARSBUFFERS(2)
SETSYS ABARSUNITNAME(3490)
```

Figure 87. Using SETSYS to Perform Aggregate Recovery

For a full list and a description of the SETSYS parameters used for aggregate recovery, see *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Issuing the ARECOVER Command

This section describes how you can verify recovery contents, perform the aggregate recovery, and use the ISMF Aggregate Group Application.

When you perform aggregate recovery, ARECOVER produces a set of output files. *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment* discusses these files in detail.

Verifying the Recovery Contents

You might want to perform a test run of the aggregate recovery before actually recovering the data. Use the DFSMSShsm ARECOVER command with the VERIFY parameter. Figure 88 shows an example of issuing the ARECOVER command with the VERIFY parameter. The verification is performed for the PAYROLL1 aggregate group.

```
ARECOVER AGGREGATE(PAYROLL1) VERIFY
```

Figure 88. Verifying Aggregate Recovery

The VERIFY parameter also indicates the amount of storage (in kilo-, mega-, giga, or terabytes) required at the recovery site. VERIFY displays the total storage, and individual totals for migration level 1 and migration level 2 volumes.

Note: The AGGREGATE parameter on the ARECOVER command can only be used if an ARECOVER command with the PREPARE option was previously issued for the aggregate. The PREPARE option catalogs the control data and the instruction/activity log files, and creates an ABR record in the DFSMSShsm control data set.

Performing the Aggregate Recovery

When you are ready to perform the recovery, issue the ARECOVER command with the EXECUTE parameter. Figure 89 illustrates the command issued for the recovery.

```
ARECOVER AGGREGATE(PAYROLL1) EXECUTE
```

Figure 89. Executing Aggregate Recovery

The command in Figure 89 invokes DFSMSShsm aggregate recovery processing for the PAYROLL1 aggregate group. In this example, user tape data sets are recovered to the tape unit type specified on the SETSYS ARECOVERUNITNAME parameter, because the optional parameter TARGETUNIT is not specified.

A number of additional parameters are available for performing aggregate recoveries, including:

MIGRATEDDATA

Recovers data sets to the specified level instead of always recovering migrated data sets to migration level 1.

NOBACKUPMIGRATED

Specifies that recovered migrated data sets do not need backing up during automatic backup.

DATASETCONFLICT

Handles data set conflict during recovery. You can choose subparameters to specify what action to take to resolve the conflict.

PERCENTUTILIZED

Allows you to determine a high threshold to which ARECOVER fills primary volumes when recovering non-system-managed, non-migrated data sets.

RECOVERNEWNAMEALL

Requests ARECOVER to rename all data sets on the ABARS files to the highest-level qualifier specified with this parameter.

RECOVERNEWNAMELEVEL

Requests ARECOVER to rename data sets with a specified highest-level qualifier to a new highest-level qualifier name, as specified under this parameter.

The DFSMSHsm ARECOVER UNIT parameter is optional and you can specify a default unit when invoking aggregate recovery. To do this, use the SETSYS ABARSUNITNAME parameter.

The DFSMSHsm ARECOVER command also supports the DFSMSHsm SETSYS NOOPTIMUMDASDBLOCKING parameter. If you use this parameter, ARECOVER does **not** use the optimum block size for DASD during recovery.

Using the ISMF Aggregate Group Application for Recovery

You can enter the ARECOVER command through the ISMF Aggregate Group Application. You can also enter specific ARECOVER parameters that apply to the application you want to recover. Figure 90 on page 142 shows the Aggregate Group Application Selection panel.

```

                                AGGREGATE GROUP APPLICATION SELECTION
COMMAND ==>

TO PERFORM AGGREGATE GROUP OPERATIONS, SPECIFY:

CDS NAME           ==> 'STGADMIN.SCD51'
                    (1 to 44 character data set name or 'ACTIVE')
AGGREGATE GROUP NAME ==> PAYROLL1 (for Aggregate Group List, fully or
                                partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ==> 6

1  LIST      - Generate a list of Aggregate Groups
2  DISPLAY   - Display an Aggregate Group
3  DEFINE    - Define an Aggregate Group
4  ALTER     - Alter an Aggregate Group
5  ABACKUP   - Backup an Aggregate Group
6  ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY VIEW CRITERIA      ==> N   (Y or N)
RESPECIFY SORT CRITERIA     ==> N   (Y or N)

USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 90. Performing Aggregate Recovery Using ISMF

When you specify ARECOVER in ISMF, you can specify the type of device to use for the aggregate recovery, whether you want to verify or execute the recovery, and whether you want to check for errors during the verification process. Figure 91 shows panel 1 of the ISMF Aggregate Group Recover application.

```

                                AGGREGATE GROUP RECOVER                                Page 1 of 7
COMMAND ==>

ABACKUP CONTROL DATASET ==>
                    (1 to 44 characters)
XMIT                ==> N   (Y or N)

AGGREGATE GROUP NAME ==> 'PAYROLL1'
DATE                ==>    (yyyy/mm/dd)
VERSION             ==> 0300 (1 to 9999)

PROCESSING OPTION   ==> 1   (1=PREPARE, 2=VERIFY, 3=EXECUTE)
WAIT FOR COMPLETION ==> N   (Y or N)

RECOVER INSTRUCTION DATA SET ==> N   (Y or N)
RECOVER ACTIVITY LOG      ==> N   (Y or N)

USE ENTER TO CONTINUE; USE DOWN TO VIEW ADDITIONAL OPTIONS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 91. Specifying Options During Aggregate Group Recovery

The description of the values specified in the panel follows:

ABACKUP CONTROL DATA SET

A blank defaults the value to AGGREGATE GROUP NAME. This parameter and the AGGREGATE GROUP NAME parameter are mutually exclusive.

XMIT

N indicates the aggregate group output tapes have not been sent to the recovery site.

AGGREGATE GROUP NAME

PAYROLL1 is the name of the aggregate group to be verified for aggregate recovery.

DATE

A blank defaults the value to VERSION. The DATE and VERSION parameters are mutually exclusive.

VERSION

0300 indicates the version number of the aggregate group. This value corresponds to an ABARS activity record.

PROCESSING OPTION

1 indicates the recovery process is in VERIFY mode.

RECOVER INSTRUCTION DATA SET

N indicates the instruction data set is not to be recovered.

RECOVER ACTIVITY LOG

N indicates the activity log is not to be recovered.

|
| For full descriptions of all the ISMF Aggregate Group Application panels for aggregate recovery, see *DFSMS/MVS V1R2 Using ISMF*.

|
| **Resolving Data Set Name Conflicts**

| You can use the ARECOVER RECOVERNEWNAMEALL,
| RECOVERNEWNAMELEVEL, or the DATASETCONFLICT parameters on the
| DFSMSHsm SETSYS command, or the installation-wide exit ARCCREXT to resolve
| data set name conflicts during aggregate recovery. If the conflicts are still not
| resolved, then ARECOVER fails and produces a conflict resolution data set. You
| can modify this data set with specific data set name conflict resolution options.
| When ARECOVER is restarted, it uses this data set to resolve remaining data set
| name conflicts.

|
| For detailed information on the use of the ARECOVER parameters and the conflict
| resolution data set, see *DFSMS/MVS V1R2 Managing Data Availability in a
| System-Managed Storage Environment*.

Chapter 8. Protecting Data

Resource Access Control Facility (RACF) is the recommended product for managing security in the DFSMS environment. RACF is used to define and verify users' authority to access data and to use system facilities. With RACF, you can control the following:

- Access to DASD and tape data sets
- Access to DASD and tape volumes
- Ability to execute programs (load modules)
- Ability to execute certain functions, keywords, and commands at other than the load module level
- Ability to define and access fields in RACF profiles

You can also log and report unauthorized attempts to access protected data sets. In the DFSMS environment, you can use RACF to provide default SMS classes and to control the use of specific management classes and storage classes.

This chapter presents the features of RACF and offers recommendations on using RACF in a DFSMS environment. It describes the following tasks:

- Implementing a protection policy
- Controlling access with RACF
- Defining access profile for the storage administration group.
- Protecting data in a shared environment

If a separate security administration group controls the use of RACF at your installation, you need to work closely with the security administrator to define SMS security requirements.

Implementing a Protection Policy

With the advent of larger systems and with ever-increasing numbers of users, password protection can no longer provide the security users require. Therefore, password protection is not available in the DFSMS environment.

To protect system-managed data sets, you need to use an external security product, such as RACF. Invocations to the product should be made through the System Authorization Facility (SAF).

Because object data is stored in DB2 table spaces kept on DASD volumes, DB2 authorization are used to protect object data.

When RACF is applied to a password-protected data set, password protection is bypassed and access is controlled solely through RACF. For information on identifying OS and VSAM password-protected data, see "Protecting User Data" on page 156.

Warning: Protecting data sets by passwords, rather than by a security subsystem is not recommended. A security subsystem provides far greater protection.

To benefit from RACF, you need to set up a protection policy. This section presents guidelines for designing and implementing a RACF protection policy for your installation.

Determining Security Categories

Implementing a RACF protection policy involves developing a set of security categories that both describe your installation's data and define the level of RACF protection necessary for each category. For example, you might use the following categories and definitions:

Public Use

No restrictions apply to the access of these data sets.

Internal Use

Data sets containing information restricted to use within your company because of its personal, technical, or business nature. Examples include organization charts, procedure manuals, and internal phone directories.

Confidential

Unauthorized disclosure of the information in these data sets might be damaging to your company. Examples include revenue, cost, profit, and financial results for part of your business; maintenance, performance, and service characteristics of one of your company's products; and employee personal data.

Highly Confidential

Unauthorized disclosure of the information in these data sets would be harmful to your company's business interests. Examples include operational plans showing future strategies; consolidated cost, profit, and other financial results; market analysis reports; and product plans.

Confidential Restricted

Unauthorized disclosure of the information in these data sets would cause immediate and severe damage to your company. Examples include long-term operating and marketing plans and strategies; drawings and documentation regarding new products; descriptions of unique parts and materials; and new technological studies.

Once you determine the categories, the next step is to inform your users about them. Publish guidelines (perhaps in your user's guide) that describe how to determine the security categories of data sets and what level of RACF protection is necessary for each category. Using the previous example, you might establish the guidelines shown in Figure 92.

Figure 92. Developing Security Guidelines

Security Category	Description
Public Use	Data is open to the public.
Internal Use	Data is limited to only the employees of the company.
Confidential	Data is limited to the set of employees with a business need to know the information, as determined by the data set owner.
Highly Confidential	Data is limited to the set of employees with a business need to know, as determined by the management of the company. Accesses by these users are logged and tracked, and periodic reports are generated for management review.
Confidential Restricted	Data is subject to the same controls as highly confidential data. These are registered with the security department and encrypted when not in use.

To enforce your policy, periodically audit selected users or user groups, working with your security department to check that they have correctly categorized and

RACF-protected their data. If you use this technique, you also need to RACF-protect tapes containing backup, archive, and vital record data with sensitive information. Tapes containing many data sets should have the same level of RACF controls as the most sensitive data on the tape. For example, if you dump a DASD volume containing a mixture of internal use and confidential data, protect the dump tape using the confidential category guidelines.

For extremely confidential data, consider using the RACF option DASD *erase-on-scratch*. This option causes the physical erasure of data at the time the data set is scratched (deleted) or released for reuse, thus preventing access to the data when the space is reallocated for other use. Be aware that this feature affects performance overhead.

DFSMSHsm supports erase-on-scratch when it deletes data during migration processing. If you want erase-on-scratch used when data is deleted from migration and backup volumes, then specify SETSYS ERASEONSCRATCH. DFSMSHsm then communicates with RACF to determine whether or not to do erase-on-scratch. SETSYS ERASEONSCRATCH only applies to DASD migration and backup volumes.

Another measure you can take to ensure erasure of confidential data is to use the Classified Data Erasure Program. Classified Data Erasure Program is a program that overwrites every area on count-key-data DASD that may have contained data at any time. This ensures that none of the original data exists anywhere on the volume.

You execute the Classified Data Erasure Program by issuing the CLEANUP command through JCL. The command is entered as part of the SYSIN data. You can also issue multiple CLEANUP commands for multiple devices as part of the input stream. For more information on using the Classified Data Erasure Program, see *Classified Data Erasure Program: User's Guide and Reference*.

Defining Data Security Profiles

RACF maintains security information about data in data profiles. A data profile describes the access authorities given to individual users and user groups.

Types of Security Profile

RACF provides two types of data set profiles: discrete and generic. Discrete profiles have a one-for-one relationship with a data set—one profile for each data set. Generic profiles have a one-for-many relationship—one profile for many data sets. A generic profile describes data that conforms to a particular naming convention. For example, all data sets with a high-level qualifier of DEPTM86 can be controlled with one generic profile, (DEPTM86.**).

We recommend you use the generic profile. Generic profiles have a number of advantages over discrete profiles. They:

- Reduce I/O activity against the RACF control data set
- Reduce the administrative effort in controlling access to large numbers of data sets
- Simplify storage management and DASD migration by reducing dependency on volume serial numbers

Authority Levels within Security Profiles

You can define authority levels for each security profile. Figure 93 shows the authority levels you can use.

Figure 93. Access Authority Levels

Authority Level	Description
ALTER	Specifies that the user or group has full control over the data set, including the ability to delete it.
CONTROL	Used for VSAM data sets and specifies that the user or group has access authority equivalent to the VSAM control password.
UPDATE	Grants read or write access to the user or group.
EXECUTE	Allows execution of program load modules but prevents users from reading or copying them.
READ	Specifies that the user or group can only read the data set.
NONE	Disallows the user or group to access the data set.

For example, the profile for data set DEPT29.LIB.CNTL can specify the following:

User T030056 has ALTER authority.
Group DEPT29 has UPDATE authority.
Group DEPT62 has READ authority.
All other users have a default authority of NONE.

You can use the RACF EXECUTE access authority to protect private program libraries (which are partitioned data sets that contain load modules). EXECUTE access authority allows users to fetch and execute programs (load modules), but also prevents them from reading or copying the programs. For example, a user can be prevented from using a utility such as IEBCOPY to copy a program from a library, but can use MVS system macros (such as LINK, LOAD, XCTL, or ATTACH), the PGM parameter on a JCL EXEC statement, or the TSO CALL command to execute programs in the library.

Note: If you specify EXECUTE for any other type of data set (such as a CLIST or EXEC), RACF treats the access authority as NONE.

To prevent creation of, and access to, data sets that do not have a discrete profile or a matching generic profile, use the protect-all option. For more information on using the access authority levels, see *RACF Security Administrator's Guide*.

Using a Default Authorization Table

Use *global access checking* to improve performance by eliminating unnecessary I/O to the RACF data set. Global access checking allows you to establish a system-wide, in-storage table of default authorization levels for selected resources, including data sets. RACF checks this table to determine whether access to a resource is permitted. If it is, RACF bypasses further security checking. If it is not, RACF continues to check other means of protection (such as profiles). That is, global access checking by itself can grant access, but it cannot deny access.

Global access checking also does not log permitted access and gathers no statistics. Therefore, data sets with generalized access rules, accessed frequently enough that no logging or statistics are desired, are good candidates for global access checking.

Controlling Access With RACF

In the DFSMS environment, you use RACF to control access to:

- RACF profile fields
- SMS control data sets
- SMS functions
- SMS classes and groups
- System-managed data

You can also use RACF to control the use of ISMF, including individual ISMF applications, line operators, and commands.

The following sections explain the use of RACF commands to control access in the DFSMS environment. For more information on RACF, see *RACF Command Language Reference*. For information on RACF auditing, see *RACF Auditor's Guide*.

If a separate security administration group controls the use of RACF at your installation, you need to work closely with the security administrator to define SMS security requirements.

Using RACF Profile Fields

Use the RACF *field resource class* to control access to the fields in a RACF profile. With this facility, you can control a user's ability to specify or update the following fields:

- Resource owner (RESOWNER) in data set profiles
- Default SMS data class, storage class, and management class values in user or group profiles
- Application identifier (DATAAPPL) in user or group profiles

Figure 94 shows how you can use RACF commands to:

- Activate the field resource class
- Define field resource classes for user and group profiles that contain default SMS classes, and allow a universal access of READ to these fields
- Allow UPDATE access to this field for the storage administration group

```
SETROPTS CLASSACT(FIELD) RACLIST(FIELD)
RDEFINE FIELD USER.DFP.DATACLAS UACC(READ)
RDEFINE FIELD USER.DFP.MGMTCLAS UACC(READ)
RDEFINE FIELD GROUP.DFP.DATACLAS UACC(READ)
RDEFINE FIELD GROUP.DFP.MGMTCLAS UACC(READ)
PERMIT USER.DFP.DATACLAS CLASS(FIELD) -
ID(STGADMIN) ACCESS(UPDATE)
PERMIT USER.DFP.MGMTCLAS CLASS(FIELD) -
ID(STGADMIN) ACCESS(UPDATE)
PERMIT GROUP.DFP.DATACLAS CLASS(FIELD) -
ID(STGADMIN) ACCESS(UPDATE)
PERMIT GROUP.DFP.MGMTCLAS CLASS(FIELD) -
ID(STGADMIN) ACCESS(UPDATE)
SETROPTS REFRESH RACLIST(FIELD)
```

Figure 94. Protecting RACF Profile Fields

Controlling Default SMS Class Assignments

If you define a resource owner in a RACF data profile, or an application identifier in a RACF user or group profile, RACF supplies these values as input variables to the ACS routines. You can use these variables at data set creation time.

- You can use the resource owner to determine which RACF default SMS classes are to be assigned to the data set. The resource owner can be used to determine a management class and storage class for a data set that is different from those that would be derived using the highest-level qualifier of the data set's name.
- You can use the application identifier to associate data sets that have different highest-level qualifiers and different resource owners. For example, you can use the application identifier of PERSONNEL to associate the following data sets:

Figure 95. Using Application Identifiers in a RACF Data Profile

Data Set Name HLQ	Resource Owner	Application Identifier
PAYROLL.**	ADMIN1	PERSONNEL
ACCTING.**	ACCT1	PERSONNEL

Protecting Control Data Sets

To prevent deliberate or accidental damage or deletion, use RACF to control access to SMS control data sets (ACDS, SCDS, and COMMDS), ACS routine source data sets, and any copies of these data sets. Only your storage administration group should have ALTER and UPDATE access to SMS control data sets. All other users can be allowed READ access to the ACDS and the SCDS to let them use the ISMF end-user applications. The COMMDS and ACS routine source data sets should have a universal access of NONE.

The following example shows how you can define RACF protection for these data sets. This example assumes that:

- The user ID ADMIN1 belongs to the storage administration group
- The group ID STGADMIN belongs to the storage administration group
- All ACDS data sets have SMS.ACDS as the first two qualifiers
- All SCDS data sets have SMS.SCDS as the first two qualifiers
- All COMMDS data sets have SMS.COMMDS as the first two qualifiers
- The data sets containing the source for the ACS routines have SMS.ACSSRC as the first two qualifiers
- ADMIN1 has CREATE group authority for the group SMS

Defining the following RACF generic data set profiles ensures that all of the specified data sets are protected as long as the naming convention is followed (see Figure 96 on page 151). These commands allow universal READ access to the ACDS and the SCDS; they prevent universal access to the COMMDS and to the source data sets for the ACS routines.

```
ADDSD 'SMS.ACDS.*' UACC(READ)
ADDSD 'SMS.SCDS.*' UACC(READ)
ADDSD 'SMS.COMMDS.*' UACC(NONE)
ADDSD 'SMS.ACSSRC.*' UACC(NONE)
```

Figure 96. Allowing Read Access to Control Data Sets

The storage administrator then specifies the particular authorization requirements for the storage administration group. These commands permit UPDATE access to the SMS control data sets for the storage administration group. Figure 97 shows an example of the commands used to allow update access to SMS control data sets.

```
PERMIT 'SMS.ACDS.*' ID(STGADMIN) ACCESS(UPDATE)
PERMIT 'SMS.SCDS.*' ID(STGADMIN) ACCESS(UPDATE)
PERMIT 'SMS.COMMDS.*' ID(STGADMIN) ACCESS(UPDATE)
PERMIT 'SMS.ACSSRC.*' ID(STGADMIN) ACCESS(UPDATE)
```

Figure 97. Allowing Update Access to Control Data Sets

You can also permit your storage administrator group to create and delete the SMS control data sets. Figure 98 shows an example of the commands used to allow alter access to SMS control data sets. For added security, you might want to allow only one storage administrator ALTER access.

```
PERMIT 'SMS.ACDS.*' ID(STGADMIN) ACCESS(ALTER)
PERMIT 'SMS.SCDS.*' ID(STGADMIN) ACCESS(ALTER)
PERMIT 'SMS.COMMDS.*' ID(STGADMIN) ACCESS(ALTER)
PERMIT 'SMS.ACSSRC.*' ID(STGADMIN) ACCESS(ALTER)
```

Figure 98. Allowing Alter Access to Control Data Sets

DFSMSShsm invokes RACF to check any DFSMSShsm user command that manipulates data or that affects DFSMSShsm control records. To provide DFSMSShsm with the RACF authority necessary to perform all DFSMSShsm functions, you need to define a RACF user ID and ensure that the DFSMSShsm RACF user ID is added to the RACF started task table. For more detail, see *DFSMS/MVS V1R2 DFSMSShsm Implementation and Customization Guide*.

You can also use RACF to protect DFSMSrmm control data sets and ensure that only authorized users can access data defined to DFSMSrmm. To protect the DFSMSrmm control data set, create a security resource profile for it. To prevent inadvertent updates to the data set, specify a RACF universal access of NONE. Give CONTROL access only to authorized users. For more information on protecting DFSMSrmm data sets, see *DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide*.

Controlling the Use of ISMF Functions

Use the RACF PROGRAM resource class to control the use of ISMF, including individual ISMF applications, dialog commands, and line operators. With the RACF program control feature, you can set up authorization levels that meet the needs of your installation for each of these categories.

Define a number of RACF profiles to limit the use of ISMF applications. See *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference* for a list of these profiles. Consider, however, making both the list and display dialogs available to all users.

Controlling Configuration Activation

Use the RACF FACILITY resource class to control the ability to activate a configuration and to perform various catalog and DFSMSdss functions against system-managed data sets:

- The FACILITY resource class profile STGADMIN.IGD.ACTIVATE.CONFIGURATION protects the ability to activate an SMS configuration. You must use ISMF to define this RACF profile to activate a configuration. If the FACILITY resource class is inactive, or if the named profile does not exist, the operator is queried to decide whether the activate action should be allowed.
- Catalog and DFSMSdss functions on system-managed data sets are only allowed if profiles exist or if the FACILITY resource class is active. Accordingly, you should not activate SMS without protecting these functions. A list of the profiles you must define to protect catalog and DFSMSdss functions on system-managed data sets is provided in *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

You can create a separate RACF profile to individually authorize each function, keyword, and command against system-managed data sets. Or, using the common highest-level qualifier, STGADMIN, you can create RACF generic profiles for command or operation authorization.

Figure 99 illustrates the RACF commands that must be issued to define the ability to activate an SMS configuration:

```
SETROPTS CLASSACT(FACILITY)
RDEFINE FACILITY STGADMIN.IGD.ACTIVATE.CONFIGURATION UACC(NONE)
PERMIT STGADMIN.IGD.ACTIVATE.CONFIGURATION CLASS(FACILITY) -
      ID(STGADMIN) ACCESS(READ)
```

Figure 99. Controlling the Activation of an SMS Configuration

Controlling Execution of Load Modules

The *program resource class* can prevent unauthorized users from executing ISMF functions. A user is allowed to execute a load module when one of the following is true:

- Individual users are authorized to execute the load module corresponding to the function they request. Authorization is defined as READ level access or greater.

- The user's RACF profile has the OPERATIONS attribute.
- The user's group is authorized to execute the load module.
- The universal access authority (UACC) for the load module is READ or greater. This makes the load module available to anyone who can access ISMF.
- RACF is disabled, or program control is turned off.

Figure 100 shows the RACF commands you use to:

- Activate RACF program control
- Define a profile for the ISMF load module containing the ACTIVATE command
- Allow READ access to the storage administration group

```

SETROPTS WHEN(PROGRAM)
RDEFINE PROGRAM DGTFCAT UACC(NONE) -
        ADDMEM('SYS1.DGTLIB'/MVSRES/NOPADCHK)
PERMIT DGTFCAT CLASS(PROGRAM) ID(STGADMIN) ACCESS(READ)
SETROPTS REFRESH WHEN(PROGRAM)

```

Figure 100. Controlling Load Module Execution

DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference lists the profiles that you define if you want to limit the use of ISMF functions. For more information on using program resource class, see *RACF Security Administrator's Guide*.

Controlling the Use of Storage Management Subsystem Functions

Use the RACF FACILITY resource class to control the use of functions, keywords, and commands at other than the load module level. With FACILITY resource class, you can control the use of:

- Catalog functions for SMS data sets. These functions include:
 - Defining a non-VSAM catalog entry for a data set that does not have a BCS entry
 - Deleting the BCS entry for a non-VSAM data set without deleting the associated VVDS and VTOC entries.
- DFSMSdss functions on data sets. These functions include:
 - Allowing the user to bypass the ACS routines on DFSMSdss COPY and RESTORE, and to supply SMS classes
 - Using the INCAT parameter on the COPY and DUMP commands.
- Using the ISMF function to activate a configuration.

DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference lists the profiles that you need to define to use FACILITY resource class to control the use of these functions.

Figure 101 on page 154 shows the RACF commands you use to:

- Activate RACF FACILITY resource class
- Define a resource class for the DFSMSdss COPY function with BYPASSACS
- Allow READ access to this function for the storage administration group

```
SETROPTS CLASSACT(FACILITY)
RDEFINE FACILITY STGADMIN.ADR.COPY.BYPASSACS UACC(NONE)
PERMIT STGADMIN.ADR.COPY.BYPASSACS CLASS(FACILITY) -
        ID(STGADMIN) ACCESS(READ)
SETROPTS REFRESH RACLIST(FACILITY)
```

Figure 101. Controlling SMS Functions

Controlling Storage Management Subsystem Classes and Groups

Use RACF to set up default data classes, management classes, and storage classes to be assigned when new system-managed data sets are allocated. You can use the SMS classes as input variables to the ACS routines. You can also control the use of SMS classes with RACF.

Using Default SMS Classes

RACF uses the *resource owner* (data set owner) to determine the user or group profiles that contain the default SMS classes. The resource owner is specified in the RESOWNER field in the data set profile or, if none is specified, it defaults to the user or group named by the high-level qualifier of the data set name. The resource owner is also available as an input variable to the ACS routines. The owner must be a valid RACF user or group.

When you decide whether to use RACF default SMS classes, keep in mind it is unlikely that the same classes can be used to effectively manage all of the data sets created by one user. However, default classes might be appropriate for a user group that creates only one type of data set. For example, an administrative support group might create only text data sets and these data sets can be managed using common SMS classes.

Defining Default SMS Classes

You define default SMS classes using RACF commands or the RACF ISPF panels. To use the defaults as input to the ACS routines, specify ACSDEFAULTS=YES in the IGDSMSxx member of SYS1.PARMLIB. To assign the RACF default classes, set the SMS classes equal to the RACF default classes in the ACS routines. For example, to use the RACF default for storage class, specify the storage class ACS routine as Figure 102 shows:

```
SET &STORCLAS=&DEF_STORCLAS
```

Figure 102. Using the RACF Default for Storage Class

Controlling the Use of SMS Classes

Use RACF to control the ability of a resource owner to use a particular storage class or management class at data set creation. You cannot and do not need to protect data classes and storage groups.

RACF contains two resource classes: STORCLAS and MGMTCLAS. These two resource classes are distinct from SMS storage class and management class.

You use the RACF *management class and storage class resource classes* to authorize the use of individual management classes and storage classes at data set or object creation. Authorization is based on the user ID or the group ID that is identified by the resource owner field in the RACF profile for the data set or object. If a resource owner is not specified, authorization is determined from the highest-level qualifier of the data set or object name.

You can use RACF to authorize the use of storage classes and management classes by most users based on the highest-level qualifier of the data set or object name. You define the highest-level qualifier as either a RACF user or group. However, use RACF to authorize the use of special classes by users or groups who require a level of access that cannot be inferred from the highest-level qualifier of the data set or object name. Checking authorization is especially important when you use RACF default classes or if you allow users to specify classes when they allocate data sets or store objects.

The following example illustrates using the storage class and management class resource classes to limit the use of storage classes and management classes.

The users of a payroll application create data sets with the highest-level qualifier of PAYROLL. Most of these data sets require only a storage class of STANDARD. However, data sets that start with the qualifiers PAYROLL.MAINFILE require the CRITICAL storage class. For these data sets, we define a special RACF data set profile with a resource owner of PAYADMIN. For new data set allocations we use ACS routines to:

- Assign the CRITICAL storage class to PAYROLL.MAINFILE.** data sets
- Assign the STANDARD storage class to other PAYROLL.** data sets

We use RACF to prevent unauthorized access to the CRITICAL storage class and to allow access to the PAYADMIN resource owner.

Figure 103 shows using RACF commands to do the following:

- Activate the RACF storage class resource class
- Define a storage class profile with a universal access of NONE
- Allow READ access to the storage class for the PAYADMIN resource owner

```
SETROPTS CLASSACT(STORCLAS) RACLIST(STORCLAS)
RDEFINE STORCLAS CRITICAL UACC(NONE)
PERMIT CRITICAL CLASS(STORCLAS) -
      ID(PAYADMIN) ACCESS(READ)
SETROPTS REFRESH RACLIST(STORCLAS)
```

Figure 103. Controlling Use of SMS Classes

You only need to specify the SETROPTS command once for each resource class. After the command is issued, the resource class is active and you can define profiles to it.

Protecting User Data

This section presents recommendations for protecting DASD and tape data sets, objects, catalogs, database data, tape volumes, and data in a distributed environment.

Data Sets and Objects

Although OS and VSAM passwords are maintained in a DFSMS environment, these passwords are not used for system-managed data sets that include integrated catalog facility catalogs on system-managed volumes. RACF does all authority checking using the SAF interface. Therefore, if you require any authorization checking, you must ensure that your data sets and catalogs are protected by RACF in one of the following ways:

- If you have implemented the RACF protect-all option, no additional work is required. RACF checking supersedes password checking, and access fails if the data set is not RACF-protected.
- If you use RACF and do not use the protect-all option, then you might want to review protected data sets and convert password-only protected data sets to RACF protection.

You can use the RACF LISTRP sample CLIST to identify data sets that are not protected by a RACF profile (that is, data sets and catalogs that are either unprotected or are protected by VSAM or OS passwords).

Object data are stored in DB2 table spaces on DASD volumes. Therefore, you use the DB2 authorization to protect object data. For more information on DB2 authorization functions, see *DB2 Administration Guide*.

Catalogs

Without proper catalog security measures, the risk is great that data can be exposed to unauthorized access. By following the guidelines listed below, you secure your current data sets and prepare yourself for the security of data sets under system-managed storage:

- Use RACF, rather than passwords, for protecting catalogs. RACF offers effective user verification, resource authorization, and logging capabilities. Passwords are ignored for system-managed catalogs.
- Ensure that master catalogs contain only the required system data sets and user catalog connectors and aliases. To achieve this, protect master catalogs with the UPDATE level of authority, with very limited update access. Note that this also prevents the creation of data sets with nonstandard names.
- If security requirements are stringent for a particular application, establish a separate catalog for it. The catalog should be protected at the ALTER level to safeguard the catalog against unauthorized deletion of the catalog entries, but still allow authorized access and additions.
- Use RACF facility class to control use of repair functions on catalog entries for system-managed data sets.
- Maintain an access list for each catalog.
- Monitor and report attempted security violations.

Database Data

DB2 has its own authorization that can be grouped into administrative roles. The database administrator should define the needed authorization when the configuration database is created.

You use DB2 authorization to protect object data.

Certain authorization functions are needed beyond the data set level and are outside the scope of RACF. Because of the special nature of these functions, they can be implemented in a particular database product (for example, DB2 and IMS). For more information about security and DB2, see *DB2 Administration Guide*. For more information about security and IMS, see *IMS/ESA System Administration Guide*.

Tape Data Sets and Volumes

RACF controls user access to data on both tape and DASD. However, if you have DFSMS/MVS installed, you can use DFSMSrmm to extend the tape data and volume protection.

By tracking the expiration dates and retention period of data sets on a tape volume, DFSMSrmm prevents write access to a data set before it expires. In addition, DFSMSrmm does not release a volume until either the last data set on the volume is expired, or the retention period you specified in the vital record specifications definition is fulfilled.

DFSMSrmm also provides support for using RACF standard tape volume security protection, using the RACF TAPEVOL and TAPEDSN options. This support allows DFSMSrmm to create, change, and delete volume profiles on volume registration, expiration, or release.

DFSMSrmm can provide automatic security classification through installation-specified criteria. For classified volumes, DFSMSrmm offers the following control:

- Audit trail of access and change of status through system management facilities (SMF). The audit produces information about RACF userid, group, and jobname.
- Operator confirmation of use.
- Erasure of data on volume release and recycle.

For more information on using DFSMSrmm or DFSMSrmm in conjunction with RACF to protect your tape data and volumes, see *DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide*.

Data in a Distributed Environment

If you maintain data in a distributed environment, we recommend that you use RACF and define generic profiles to protect the data. With generic profiles, you can maintain the various security levels and access authorization for the data as appropriate for the multiple user groups. For more information on data security in a distributed environment, see *ADSM Administrator's Guide*.

Controlling the Use of Aggregate Backup and Recovery Commands

With RACF FACILITY class profile support, you can control the use of the ABACKUP and ARECOVER commands by any console operator or user (including users who are not DFSMSHsm-authorized users). If the RACF FACILITY class profile support is installed and activated using the SETROPTS command, DFSMSHsm checks the console operators' or users' RACF profile for authorization to issue the ABACKUP and ARECOVER commands. If the RACF FACILITY class profile is not active, only DFSMSHsm-authorized users are allowed to issue the ABACKUP and ARECOVER commands. For more information, see the *DFSMS/MVS V1R2 Managing Data Availability in a System-Managed Storage Environment*.

Defining Access Profile for the Storage Administration Group

As a storage administrator, you negotiate with your security administrator to gain the following RACF access authorization:

- CLAUTH authority to storage class and management class resource classes. This allows you to define, modify, delete, and build access lists for the storage class and management class profiles.
- Authority for all ISMF functions.
- Authority to perform special catalog functions and DFSMSdss operations for system-managed data sets.
- Field resource class UPDATE authority for the RESOWNER, DATAAPL, MGMTCLAS, STORCLAS, and DATACLAS fields.

You might want to set up a group profile for the storage administration group that provides these levels of authorization.

Chapter 9. Monitoring and Reporting for Storage Management

Monitoring storage activities in your installation helps meet user requirements. By using the IDCAMS DCOLLECT command, and the Service Level Reporter (SLR), you can track changing space trends, identify ineffective use of space, and detect space-related errors. By collecting specific System Management Facilities (SMF) records, you can obtain information about the caching activities in your installation. You can use this information to decide how to improve the overall system performance. In addition, the volume mount analyzer can help you analyze current tape mount practices so that you can plan for tape mount reduction.

This chapter explains how you can use DCOLLECT, SMF records, RMF, SLR, and the volume mount analyzer to monitor and report storage activities. It describes the following tasks:

- Monitoring space use
- Measuring DASD and HSM-owned tape capacity
- Monitoring performance
- Collecting cost accounting information.
- Analyzing tape use to reduce tape mounts

Monitoring Space Use

To ensure efficient use of DASD space, you need to make sure that space is allocated and used according to storage management standards and practices. By using the information DCOLLECT produces on active and inactive data, and on DASD and tape volumes, you can monitor space use.

This section describes the information DCOLLECT provides, ways you can use DCOLLECT, and the types of reports you can generate with DCOLLECT output.

Collecting Information on Data

DCOLLECT gathers information on active and inactive data. The information it collects on active data is placed in the DCOLLECT active data set records. The inactive data information, consisting of backed up data and migration data information, is kept in DCOLLECT backup records and migration records. DCOLLECT also gathers VSAM data set space use (HURBA) information. For a summary of these record types and their contents, see Appendix B, "DCOLLECT Record Types" on page 211.

Information in these records can help you in the following ways:

- Identify where space is wasted in primary storage. For example, using the information from the active data set records, you can determine the amount of allocated but unused space in your installation. You can then use this information to develop a plan to reclaim unused space; assign data sets a management class that releases allocated but unused space.

- Track wasted storage due to inactive data sets that are not migrated, or data sets that can reside on tape but are kept on DASD. You can use the information in migration or backup records to help you track wasted storage.
- Assess the overall space savings from compressing VSAM key sequenced data sets or sequential data sets. DCOLLECT collects information on the size of data sets before and after compression.

Use the IDCAMS DCOLLECT command to gather information on data. You can invoke the command through JCL or ISMF.

The JCL example in Figure 104 tells DCOLLECT to collect information on all migrated and backed up data sets that reside on the system.

```

//COLLECT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//MCDS DD DSN=HSM.MCDS,DISP=SHR
//BCDS DD DSN=HSM.BCDS,DISP=SHR
//OUTDS DD DSN=DCOLLECT.OUTPUT,
//      SPACE=(1,(100,100)),
//      DATACLAS=DATAV,
//      STORCLAS=LARGE,
//      DISP=NEW,
//      AVGREC=K
//SYSIN DD *
        DCOLLECT OFILE(OUTDS) -
            MIGRATEDATA -
            BACKUPDATA
/*
//

```

Figure 104. Using DCOLLECT to Collect Migrated and Backup Data Set Data

You can also use ISMF to generate the JCL for IDCAMS DCOLLECT. Figure 105 on page 161 shows a sample ISMF Data Collection Application panel used to generate the JCL in Figure 104.

```

                                DATA COLLECTION ENTRY PANEL                PAGE 1 of 2
COMMAND ==>>

SELECT DATA COLLECTION OPTIONS:
  DATA SET INFORMATION    ==>> Y      (Y or N; Y requires volume(s) or
  VOLUME INFORMATION       ==>> N      storage group(s) on next page)
  MIGRATION DATA          ==>> Y      (Y or N)
  BACKUP DATA             ==>> Y      (Y or N)
  CAPACITY PLANNING DATA  ==>> N      (Y or N)

SPECIFY OUTPUT DATA SET:          (1 to 44 characters)
  DATA SET NAME                  ==>> DCOLLECT.OUTPUT
  OPTIONAL PASSWORD               ==>>      (Ignored if SMS-managed data set)
  REPLACE CONTENTS                ==>> N      (Y or N)
  NUMBER OF DATA SETS            ==>> 1      (1 to 99999999; new data set only))

SPECIFY INPUT DATA SET:          (1 to 44 characters)
  MIGRATION DATA SET NAME        ==>> 'HSM.MCDS'
  BACKUP DATA SET NAME           ==>> 'HSM.BCDS'

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT
ENTRY PANEL; USE HELP COMMAND FOR HELP;
USE END COMMAND TO EXIT.

```

Figure 105. Using ISMF to Collect Migrated and Backup Data Set Data

We recommend that you run DCOLLECT once a day, after the DFSMSHsm space management cycle. However, depending on the needs of your installation, you might need to run DCOLLECT more frequently.

For more information on using IDCAMS DCOLLECT, see *DFSMS/MVS V1R2 Access Method Services for ICF*. For information on using ISMF for IDCAMS DCOLLECT, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Collecting Information on Volumes

DCOLLECT also gathers information on volumes and storage groups. The information it collects on volumes and storage groups is kept in the DCOLLECT volume records. For a summary of the volume record and its contents, see Appendix B, “DCOLLECT Record Types” on page 211.

You can use the information from volume and storage groups to monitor free space capacity and storage group thresholds. Monitoring free space capacity and storage group thresholds helps meet the expected amount of free space in the storage groups, as documented in the service-level agreement. It also helps prevent space-related abends.

Use the IDCAMS DCOLLECT command to collect volume and storage group information. You can invoke the command through JCL or ISMF.

The JCL example in Figure 106 on page 162 specifies a partially-qualified volume serial to collect data for data sets residing on all online volumes whose serial numbers begin with PRM.

```

//STEP1 EXEC PGM=IDCAMS
//*
//*
//SYSPRINT DD SYSOUT=*
//DCOUT DD DSN=DEPTM86.DCOLLECT.DATASET.OUTPUT,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(1,(25,12)),AVGREC=M,
// DSORG=PS,RECFM=VB,LRECL=264
//SYSIN DD *
DCOLLECT -
        OUTFILE(DCOUT) -
        VOLUMES( -
                PRM* -
                ) -
/*
//

```

Figure 106. Using DCOLLECT to Collect Generic Volume Data

The JCL example in Figure 107 specifies that only volume data is to be collected for all online volumes residing in storage group PRIME90.

```

//COLLECT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DCOUT DD DSN=DEPTM86.DCOLLECT.VOLUME.OUTPUT,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(1,(100,100)),AVGREC=K,
// DSORG=PS,RECFM=VB,LRECL=264
//SYSIN DD *
DCOLLECT -
        OUTFILE(DCOUT) -
        STORAGEGROUP(PRIME90) -
        NODATAINFO
/*
//

```

Figure 107. Using DCOLLECT to Collect Storage Group Data

You can also use ISMF to generate the JCL for IDCAMS DCOLLECT. Figure 108 on page 163 shows a sample ISMF Data Collection Application panel used to generate the JCL in Figure 106.

```

                                DATA COLLECTION ENTRY PANEL                PAGE 1 of 2
COMMAND ==>>

SELECT DATA COLLECTION OPTIONS:
  DATA SET INFORMATION    ==>> Y      (Y or N; Y requires volume(s) or
  VOLUME INFORMATION       ==>> Y      storage group(s) on next page)
  MIGRATION DATA          ==>> N      (Y or N)
  BACKUP DATA             ==>> N      (Y or N)
  CAPACITY PLANNING DATA ==>> N      (Y or N)

SPECIFY OUTPUT DATA SET:                (1 to 44 characters)
  DATA SET NAME             ==>> DEPTM86.DCOLLECT.DATASET.OUTPUT
  OPTIONAL PASSWORD         ==>>      (Ignored if SMS-managed data set)
  REPLACE CONTENTS          ==>> N      (Y or N)
  NUMBER OF DATA SETS      ==>> 1      (1 to 99999999; new data set only)

SPECIFY INPUT DATA SET:                (1 to 44 characters)
  MIGRATION DATA SET NAME ==>>
  BACKUP DATA SET NAME    ==>>

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT ENTRY PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 108. Using ISMF to Collect Volume Information

We recommend that you run DCOLLECT once a day, after the DFSMSHsm space management cycle. However, depending on the needs of your installation, you might want to run DCOLLECT more frequently.

For more information on using IDCAMS DCOLLECT, see *DFSMS/MVS V1R2 Access Method Services for ICF*. For information on using ISMF for IDCAMS DCOLLECT, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Producing Space Use Reports

The amount of DCOLLECT output depends on the size of the DASD subsystem, the number of data sets (active and inactive), and the options selected in the DCOLLECT job. Because a large amount of raw data can be collected over time, you need to implement a process that summarizes the data.

The Service Level Reporter (SLR), Structured Query Language (SQL), or Query Management Facility (QMF*) are products that you can use to summarize and present the DCOLLECT data. You can also use ICETOOL, a data processing and reporting utility introduced in DFSORT 11.1, to extract information from DCOLLECT and produce a variety of useful reports.

Reporting with SLR

We recommend using SLR to produce sophisticated and detailed reports. You can provide select statements to list desired DCOLLECT records for reporting.

Figure 109 shows sample SLR control statements that extract information from the DCOLLECT active data set, migration and backup records.

```
REPORT DASD_DSNSTAT
  SEL C(PROJECT NUMDATASET ALLOC_SP UNUSED_SP UNUSED_BLK  +
        AVG_DS_NREF)
  SEL R((MONTH=*) AND (DAY=16))
  SORT MAX( 200)
  LIST ALL
  PRINT TABLE
ENDREP
```

Figure 109. Using SLR to Produce Project Space Allocation Reports

Unless you need to review the raw data on a regular basis, we recommend that you store a summarized form of the DCOLLECT output on DASD and migrate the DCOLLECT output to tape.

The types of SLR reports you can produce with DCOLLECT output are trend reports, detail reports, and exception reports. Trend reports provide an overview of your installation's storage use. Capacity planners use these reports to forecast future requirements. Detail reports provide information that helps you to enforce storage management standards and to focus on short-term space requirements. Exception reports contain information on infrequent errors, and are only produced when abnormal situations occur.

As the storage administrator, you probably need to produce detail reports on a regular basis. Figure 110 on page 165 shows a sample detail report on space allocation at the project level.

PROJECT	NUMBER DATASETS	ALLOC_SP K-BYTES	USED_SP K-BYTES	UNUSED_SP K-BYTES	UNUSED_BLK K-BYTES	AVG_NREF DAYS
*****	*****	*****	*****	*****	*****	*****
SYSTEM	455	5217.5	846.6	4370.87	75.47	126
HSM	9573	6838.6	3689.3	3149.36	9.60	13
PROJ017	1362	8625.9	5677.5	2948.41	190.80	95
PROJ004	6303	5602.2	4129.2	1473.04	532.35	97
PROJ003	10341	3927.1	3296.6	630.56	320.86	120
PROJX15	227	530.5	102.2	428.25	9.17	83
PROJDC0	40	713.8	351.3	362.54	88.01	1
PROJMF2	6242	2168.3	2029.4	138.93	181.99	106
PROJ005	4296	1724.2	1589.8	134.45	189.22	108
PROJ001	2610	604.8	501.0	103.80	60.16	109
TOTALS	41449	35952.9	22212.9	13740.21	1657.63	86
LEGEND:						

PROJECT	: Project identifier					
NUMBER DATASETS	: Number of data sets allocated					
ALLOC_SP K-BYTES	: Allocated kilobytes of space					
USED_SP K-BYTES	: Used kilobytes of space					
UNUSED_SP K-BYTES	: Allocated but unused kilobytes of space					
UNUSED_BLK K-BYTES	: Unused kilobytes of space due to inefficient blocksize					
AVG_NREF DAYS	: Average number of days data sets not referenced					

Figure 110. Generating a Detail Report on Space Use by Project

You can use the report in Figure 110 to identify projects that are using space inefficiently. When you identify the projects, assign selected data sets a management class that enforces the release of allocated but unused space. You can also ensure these data sets use system-determined blocksize, or specify a more efficient blocksize in JCL during initial data set allocation.

For more information on using SLR, see *SLR User's Guide: Reporting Planned Availability*. For information on other available SLR reports, see the SLR 3.2 publications updates for IDCAMS DCOLLECT.

Reporting with DFSORT

You can use the DFSORT utility, ICETOOL, to analyze and report on your DCOLLECT data in a variety of ways. Figure 111 on page 166 shows a sample job using the DFSORT Release 12 version of ICETOOL. The example illustrates some of the many ways you can use this data processing and reporting utility with DCOLLECT data. For complete details about DFSORT and ICETOOL, see *DFSORT Application Programming Guide*.

Note: You must add 5 to a DCOLLECT field offset to get its position for DFSORT and ICETOOL statements. For example, the DCURCTYP field is at position 9 (offset 4 plus 5).

```

//S1      EXEC  PGM=ICETOOL,REGION=1024K
//DFSMSG  DD   SYSOUT=*
//TOOLMSG DD   SYSOUT=*
//INDD    DD   DSN=DELLI.DCOLLECT,DISP=SHR
//OUTDD1  DD   DSN=&&TEMP1,DISP=(NEW,PASS),LIKE=DELLI.DCOLLECT
//SGVOLS  DD   SYSOUT=*
//OUTDD2  DD   DSN=&&TEMP2,DISP=(NEW,PASS),LIKE=DELLI.DCOLLECT
//SCRPT   DD   SYSOUT=*
//OUTDD3  DD   DSN=&&TEMP3,DISP=(NEW,PASS),LIKE=DELLI.DCOLLECT
//VOLRPT  DD   SYSOUT=*
//TOOLIN  DD   *
* Part 1
* Show the number of system-managed data sets that may be reblocked
  COUNT FROM(INDD) USING(SMSM)

* Part 2
* Create a temporary data set with only 'V' type records
  COPY FROM(INDD) TO(OUTDD1) USING(CPY1)

* Show free space statistics for the 'V' type records
  STATS FROM(OUTDD1) ON(40,1,BI)

* Show the number of volumes with free space > 30%
  RANGE FROM(OUTDD1) ON(40,1,BI) HIGHER(30)

* Part 3
* Print a report showing the number of volumes in each storage group
  OCCUR FROM(OUTDD1) LIST(SGVOLS) BLANK -
  DATE TITLE('Volumes in Each Storage Group') -
  HEADER('Storage Group') HEADER('Volume Count') -
  ON(87,30,CH)          ON(VALCNT)

```

Figure 111 (Part 1 of 3). Using ICETOOL for Analysis and Reporting

```

* Part 4
* Create a temporary data set with the total allocated space
* for each storage class using type 'D' records
  SORT FROM(INDD) TO(OUTDD2) USING(SRT1)

* Print a report showing total allocated space by storage class
* and the maximum, minimum and average allocated space among
* the storage classes
  DISPLAY FROM(OUTDD2) LIST(SCRPT) BLANK -
  TITLE('Total Allocated Space (KB) by Storage Class') DATE TIME -
  HEADER('Storage Class') HEADER('Total Space (KB)') -
  ON(171,10,CH) ON(93,4,BI) -
  MAXIMUM('Most Space:') MINIMUM('Least Space:') -
  AVERAGE('Average Space:')

* Part 5
* Create a temporary data set with only records meeting all
* of the following criteria:
* - Type 'C'
* - Occupancy after processing > target occupancy
* - Volume level is L0
  COPY FROM(INDD) TO(OUTDD3) USING(CPY2)

* Print a report showing volume, target occupancy,
* occupancy after processing and date
  DISPLAY FROM(OUTDD3) LIST(VOLRPT) BLANK -
  DATE(DMY.) TITLE('Volumes Not Meeting Target Threshold') -
  HEADER('Volume') ON(29,6,CH) -
  HEADER('Target Occupancy') ON(45,1,FI) -
  HEADER('Occupancy After Processing') ON(48,1,FI) -
  HEADER('Date') ON(35,4,PD)
//SMSMCNTL DD *
* Include only D-type records with DCDSMSM and DCDREBLK flags on
  OPTION VLSHRT
  INCLUDE COND=(9,1,CH,EQ,C'D',AND,74,1,BI,EQ,B'.1...1..')
/*
//CPY1CNTL DD *

```

Figure 111 (Part 2 of 3). Using ICETOOL for Analysis and Reporting

```

* Include only V-type records
  INCLUDE COND=(9,1,CH,EQ,C'V')
/*
//SRT1CNTL DD *

* Include only D-type records, sort by storage class and
* sum (total) allocated space
  INCLUDE COND=(9,1,CH,EQ,C'D')
  SORT FIELDS=(171,10,CH,A)
  SUM FIELDS=(93,4,BI)
/*
//CPY2CNTL DD *

* Include only C-type records with volume level of L0 and
* occupancy after processing > target occupancy
  OPTION VLSHRT
  INCLUDE COND=((9,1,CH,EQ,C'C'),&,
               (39,1,BI,EQ,X'00'),&,
               (48,1,FI,GT,45,1,FI))
/*

```

Figure 111 (Part 3 of 3). Using ICETOOL for Analysis and Reporting

The job in Figure 111 on page 166 does the following:

1. Using the DCOLLECT active data set records, ICETOOL shows the number of system-managed data sets that may be reblocked. Note the use of DFSORT's bit level logic on the INCLUDE statement to keep just the records with the DCDSMSM and DCDREBLK flags on.
2. Using the volume records, it shows the maximum, minimum, average, and total free space among all of the volumes. It also shows the number of volumes with more than 30% free space.
3. Using the volume records, it shows the number of volumes in each storage group. The row with a blank name represents the group of non-system-managed volumes.
4. Using the active data set records, it prints a report showing the total allocated space for each storage class as well as the maximum, minimum and average space among all of the storage classes.
5. Using the DASD Capacity Planning records, it prints a report showing primary volumes whose target threshold was not met during primary space management. Note that this information can be used to plan for additional storage and should be discussed with your capacity planner.

Figure 112 on page 169 shows the sample output from ICETOOL.

TOOLMSG Output

```
ICE600I 0 DFSORT ICETOOL UTILITY RUN STARTED
ICE630I 0 MODE IN EFFECT: STOP

ICE632I 0 SOURCE FOR ICETOOL STATEMENTS: TOOLIN

      * Part 1
      * Show the number of system-managed data sets that may be reblocked
        COUNT FROM(INDD) USING(SMSM)
ICE606I 0 DFSORT CALL 0001 FOR COPY FROM INDD TO E35 EXIT USING SMSMCNTL
        COMPLETED
ICE628I 0 RECORD COUNT: 00000000003574
ICE602I 0 OPERATION RETURN CODE: 00

      * Part 2
      * Create a temporary data set with only 'V' type records
        COPY FROM(INDD) TO(OUTDD1) USING(CPY1)
ICE606I 0 DFSORT CALL 0002 FOR COPY FROM INDD TO OUTDD1 USING CPY1CNTL
        COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

      * Show free space statistics for the 'V' type records
        STATS FROM(OUTDD1) ON(40,1,BI)
ICE627I 0 DFSORT CALL 0003 FOR COPY FROM OUTDD1 TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 00000000000232
ICE607I 0 STATISTICS FOR (40,1,BI) :
ICE608I 0 MINIMUM: +00000000000000, MAXIMUM: +00000000000100
ICE609I 0 AVERAGE: +00000000000052, TOTAL : +000000000012183
ICE602I 0 OPERATION RETURN CODE: 00

      * Show the number of volumes with free space > 30%
        RANGE FROM(OUTDD1) ON(40,1,BI) HIGHER(30)
ICE627I 0 DFSORT CALL 0004 FOR COPY FROM OUTDD1 TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 00000000000232
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (40,1,BI) : 00000000000124
ICE602I 0 OPERATION RETURN CODE: 00

      * Part 3
      * Print a report showing the number of volumes in each storage group
        OCCUR FROM(OUTDD1) LIST(SGVOLS) BLANK -
        DATE TITLE('Volumes in Each Storage Group') -
        HEADER('Storage Group') HEADER('Volume Count') -
        ON(87,30,CH) ON(VALCNT)
ICE627I 0 DFSORT CALL 0005 FOR SORT FROM OUTDD1 TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN SGVOLS DATA SET
ICE628I 0 RECORD COUNT: 00000000000232
ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 00000000000005
ICE602I 0 OPERATION RETURN CODE: 00
```

Figure 112 (Part 1 of 3). Sample ICETOOL Output

TOOLMSG Output

```
* Part 4
* Create a temporary data set with the total allocated space
* for each storage class using type 'D' records
  SORT FROM(INDD) TO(OUTDD2) USING(SRT1)
ICE606I 0 DFSORT CALL 0006 FOR SORT FROM INDD    TO OUTDD2    USING SRT1CNTL
COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

* Print a report showing total allocated space by storage class
* and the maximum, minimum, and average allocated space among
* the storage classes
  DISPLAY FROM(OUTDD2) LIST(SCRPT) BLANK -
  TITLE('Total Allocated Space (KB) by Storage Class') -
  DATE TIME -
  HEADER('STORAGE CLASS') HEADER('TOTAL SPACE (KB)') -
  ON(171,10,CH)           ON(93,4,BI) -
  MAXIMUM('Most Space') MINIMUM('Least Space') -
  AVERAGE('Average Space')
ICE627I 0 DFSORT CALL 0007 FOR COPY FROM OUTDD2  TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN SCRPT DATA SET
ICE628I 0 RECORD COUNT: 0000000000000003
ICE602I 0 OPERATION RETURN CODE: 00

* Part 5
* Create a temporary data set with only records meeting all
* of the following criteria:
* - Type 'C'
* - Occupancy after processing > target occupancy
* - Volume level is L0
  COPY FROM(INDD) TO(OUTDD3) USING(CPY2)
ICE606I 0 DFSORT CALL 0008 FOR COPY FROM INDD    TO OUTDD3    USING CPY2CNTL
COMPLETED
ICE602I 0 OPERATION RETURN CODE: 00

* Print a report showing volume, target occupancy,
* occupancy after processing and date
  DISPLAY FROM(OUTDD3) LIST(VOLRPT) BLANK -
  DATE(DMY.) TITLE('Volumes Not Meeting Target Threshold') -
  HEADER('Volume') ON(29,6,CH) -
  HEADER('Target Occupancy') ON(45,1,FI) -
  HEADER('Occupancy After Processing') ON(48,1,FI) -
  HEADER('Date') ON(35,4,PD)
ICE627I 0 DFSORT CALL 0009 FOR COPY FROM OUTDD3  TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN VOLRPT DATA SET
ICE628I 0 RECORD COUNT: 0000000000000008
ICE602I 0 OPERATION RETURN CODE: 00

ICE601I 0 DFSORT ICETOOL UTILITY RUN ENDED - RETURN CODE: 00
```

Figure 112 (Part 2 of 3). Sample ICETOOL Output

SGVOLS Output			
10/15/92 Volumes in Each Storage Group			
Storage Group			Volume Count
-----			-----
			156
LARGE			36
PRIMARY			36
SCR3380			2
SCR3390			2

SCRPT Output			
Total Allocated Space (KB) by Storage Class 10/15/92 15:10:51			
Storage Class	Total Space (KB)		
-----	-----		
	117099657		
NORMAL	64050140		
NORMALG	7234481		
Most Space:	117099657		
Least Space:	7234481		
Average Space:	62794759		

VOLRPT Output			
15.10.92 Volumes Not Meeting Target Threshold			
Volume	Target Occupancy	Target Occupancy After Processing	Date
-----	-----	-----	-----
SYS302	80	81	1992287
SYS350	80	81	1992269
SYS350	80	82	1992277
SYS351	80	83	1992277
SYS351	80	82	1992278
SYS450	70	85	1992260
SYS450	70	94	1992261
SYS450	70	96	1992262

Figure 112 (Part 3 of 3). Sample ICETOOL Output

Measuring DASD and DFSMSHsm-Owned Tape Capacity

You can measure DASD and DFSMSHsm-owned tape capacity so that capacity planners can identify current requirements and evaluate future needs. This data is useful in helping to predict additional space requirements before the demand becomes critical.

This section describes the information DCOLLECT provides for capacity planning, and presents a sample capacity report.

Collecting Capacity Information

DCOLLECT gathers information you can use to measure DASD and DFSMSHsm-owned tape capacity. You can use the information in the DCOLLECT active data set, migration, and backup records for capacity planning. (“Collecting Information on Data” on page 159 and “Collecting Information on Volumes” on page 161 discuss some uses of these records.) In addition, DCOLLECT places certain volume and migration information in the DCOLLECT DASD and tape capacity planning records. For a summary of these records and their contents, see Appendix B, “DCOLLECT Record Types” on page 211.

You can use information from any of the DCOLLECT records to help track volume capacity. For example, you can use information in the volume records to determine if space use is increasing or decreasing, and whether installed DASD is meeting space requirements. You can also use information in the capacity planning records to report on the amount of space not migrated so you can take appropriate actions. (If the amount of migratable space is decreasing, plan to add storage. If the migratable space is increasing, plan further investigation.)

To collect capacity information, use the IDCAMS DCOLLECT command. You can invoke the command through JCL or through ISMF.

The example in Figure 113 shows invoking IDCAMS DCOLLECT in a JCL. The JCL specifies that capacity planning data is to be collected for all volumes whose serial numbers begin with PRIM, and for all volumes in the two storage groups STGGP001 and STGGP002.

```
//COLLECT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTDS DD DSN=DCOLLECT.DASDCAP.OUTPUT,
//        SPACE=(1,(100,100)),
//        DATACLAS=DATAV,
//        STORCLAS=LARGE,
//        DISP=NEW,
//        AVGREC=M
//SYSIN DD *
        DCOLLECT OFILE(OUTDS) -
                VOL(PRIM*) -
                STOG(STGGP001,STGGP002) -
                CAPD
/*
//
```

Figure 113. Using DCOLLECT to Collect Capacity Planning Data

You can also use the ISMF Data Collection Application to generate the JCL for IDCAMS DCOLLECT. Figure 114 on page 173 shows the ISMF Data Collection Entry Panel used to generate the JCL in Figure 113.

```

                                DATA COLLECTION ENTRY PANEL          PAGE 1 of 2
COMMAND ==>

SELECT DATA COLLECTION OPTIONS:
  DATA SET INFORMATION    ==> N      (Y or N; Y requires volume(s) or
  VOLUME INFORMATION       ==> Y      storage group(s) on next page)
  MIGRATION DATA          ==> N      (Y or N)
  BACKUP DATA             ==> N      (Y or N)
  CAPACITY PLANNING DATA  ==> Y      (Y or N)

SPECIFY OUTPUT DATA SET:          (1 to 44 characters)
  DATA SET NAME                ==> DCOLLECT.DASDCAP.OUTPUT
  OPTIONAL PASSWORD              ==>      (Ignored if SMS-managed data set)
  REPLACE CONTENTS               ==> N      (Y or N)
  NUMBER OF DATA SETS           ==> 1      (1 to 99999999; new data set only)

SPECIFY INPUT DATA SET:          (1 to 44 characters)
  MIGRATION DATA SET NAME       ==>
  BACKUP DATA SET NAME         ==>

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT ENTRY PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 114. Using the ISMF Data Collection Application

For capacity planning, you need to accumulate DCOLLECT data weekly, monthly, quarterly, or for longer periods. Work with capacity planners to decide how often you need to run the DCOLLECT jobs.

For more information on using IDCAMS DCOLLECT, see *DFSMS/MVS Version 1 Release 2 Access Method Services for the Integrated Catalog Facility*. For information on using ISMF for IDCAMS DCOLLECT, see *DFSMS/MVS V1R2 DFSMSdfp Storage Administration Reference*.

Producing Capacity Planning Reports

A part of the capacity planner's function is to estimate total space required for the next business cycle. This estimate can be based on tracking space use on many different levels. Because DCOLLECT data includes many different types of information, you can use the output to produce a variety of reports.

For example, to produce a DASD capacity report, you might use SLR control statements to select DCOLLECT volume records, as Figure 115 shows.

```
REPORT DASD_VOLSTAT
  SEL C(MONTH TOTALLOC TOTCAP)
  SEL R((YEAR =TOT) AND (MONTH =TOT))
  PRINT CHART X(MONTH) Y(TOTALLOC TOTCAP) FORMAT(INST001)
ENDREP
```

Figure 115. Using SLR to Produce DASD Capacity Report

The SLR control statements in Figure 115 produce the trend report in Figure 116 on page 175. The planned capacity line in the graph is manually added.

Using the trend report in Figure 116 on page 175, your capacity planner can compare:

- The amount of DASD installed to the amount planned
- The rate of actual storage growth to the expected rate
- The actual storage use to the total amount of capacity

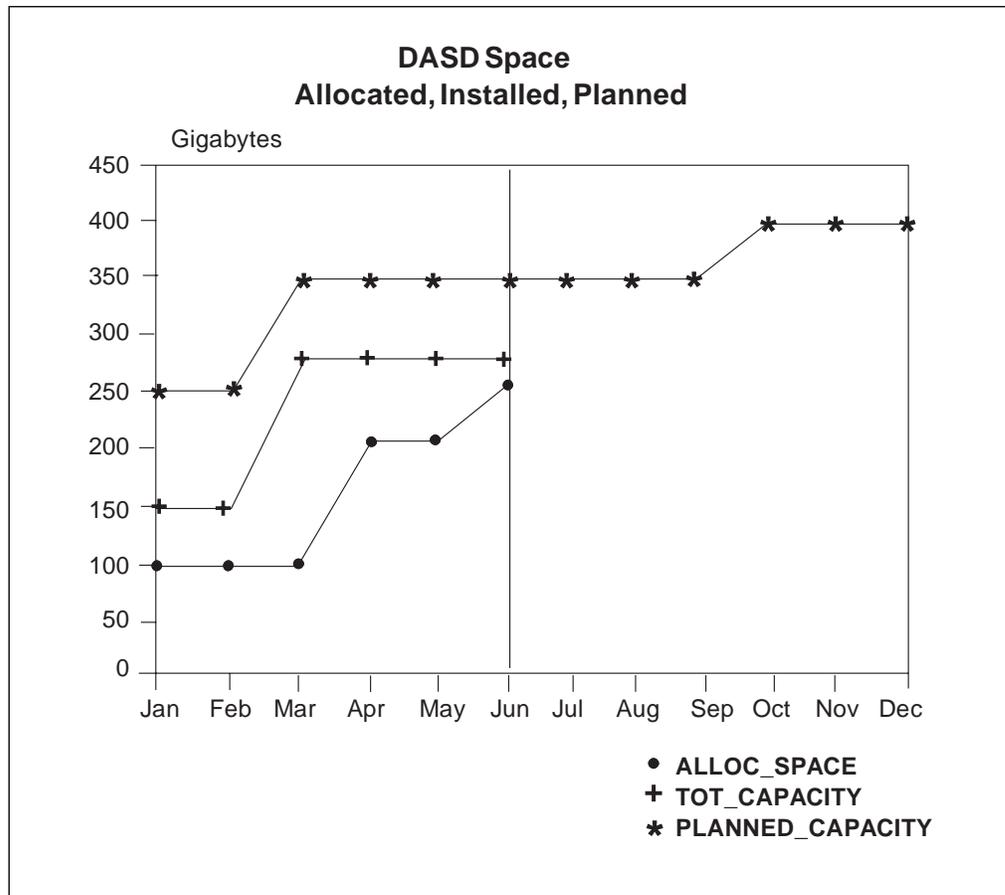


Figure 116. Producing DASD Capacity Report

If the report shows a significant deviation from the plan, further investigation is required. For example, you can produce detail reports by extracting space use information at the project level. Figure 110 on page 165 shows a sample detail report on space allocation at the project level. Although this report is used to monitor space allocation, you can also use it to determine the cause of deviations in trend reports and take corrective actions. Figure 109 on page 164 shows the SLR control statements used to generate the detail report. For more information on using SLR, see *SLR User's Guide: Reporting Planned Availability*.

Monitoring Performance

By tracking data set I/O activities, you can make better decisions about data set caching policies and improve overall system performance. For object data, you can track transaction activities to monitor and improve OAM's performance.

This section explains the use of SMF record subtypes to obtain data set performance information. It also discusses the use of RMF reports to obtain object performance information.

Collecting I/O Statistics

SMF record type 42 subtypes 5 and 6 contain I/O statistics for system-managed data sets only. The subtype 5 records contain I/O statistics by storage class and the subtype 6 records contain I/O statistics by data set.

For more information on the SMF records and subtypes, see *MVS/ESA System Management Facilities (SMF)*.

I/O Statistics by Storage Class

SMF record type 42 subtype 5 contains I/O statistics by storage class. These statistics are recorded at the same time as the Resource Measurement Facility (RMF) statistics so that they can be easily correlated. You can use storage class statistics to summarize the I/O performance in your installation.

The subtype 5 record includes the following information:

- Storage class name
- I/O counts
- Average I/O response time
- Average I/O service time components
 - connect time
 - pending time
 - disconnect time
 - control unit queue time
- 3990 Cache statistics
 - total number of cache candidate I/Os
 - total number of cache hits
 - total number of write cache candidate I/Os
 - total number of write hits

The statistics represent the I/O activity during the SMF interval. If a data set or a storage class has no I/O activity, then a data set entry is recorded. If an address space has no I/O activity during the interval, then an SMF record is not produced.

You can estimate the IOS (I/O supervisor) queue time by subtracting the average I/O service time from the average I/O response time.

You can determine read cache statistics by subtracting the write cache statistics from the total cache statistics. Read and write statistics are not recorded for non-cached I/Os or for non-3990 I/Os. If a multivolume data set partially resides behind a 3990, then the cache statistics reflect only the activity of the 3990 volumes.

An I/O is a cache candidate if the device is capable of caching the I/O (cache is active), and if the Define Extent CCW does not inhibit the caching. An I/O is considered to be a cache hit if the disconnect time is small. An inhibit-cache-load hit is counted as a cache candidate and as a cache hit.

I/O Statistics by Data Set

SMF record type 42 subtype 6 contains I/O statistics by data set. One set of statistics is recorded for each non-VSAM open, and each VSAM component per address space. At the end of each SMF interval, coincident with the recording of type 30 records, each job records a type 42 subtype 6 record, which contains one entry for each open data set. A type 42 subtype 6 record is also recorded when a data set is closed.

The subtype 6 record includes the following information:

- Job name
- Data set name
- I/O counts
- Average I/O response time
- Average I/O service time components
 - connect time
 - pending time
 - disconnect time
 - control unit queue time
- Caching statistics
 - total number of cache candidate I/Os
 - total number of cache hits
 - total number of write cache candidate I/Os
 - total number of write hits

You can determine read cache statistics using the total cache and write cache statistics. You can estimate the IOS queue time using the average I/O service time and the average I/O response time values.

VSAM statistics are collected by data set by VSAM space. When a VSAM data set is closed, and there are no more active opens to the data set, then the system produces a type 42 record. When a non-VSAM data set is closed, the system records a type 42 record.

All SMF records recorded during close contain residual statistics from the previous SMF interval. To determine all statistics for a long-running application, you must collect the SMF records for the application.

Reporting and Interpreting I/O Statistics

Use SLR, or an equivalent tool, to generate reports from the collected statistics. With the help of the reports, you can make the necessary adjustments to improve the performance of a system or application.

Some of the adjustments you might need to make include:

- Re-distribute data sets on different volumes or strings
- Re-assign storage classes to data sets
- Change block-sizes or buffer allocations
- Use system-determined block size

For example, by analyzing the reports, you might find that the average connect time for a sequential application is too small. To correct this problem, increase the block size or the number of access method buffers.

Using the data set statistics report, you might find that certain may-cache data sets are caching poorly. These data sets might contribute to the large disconnect, pending, or IOS queue times of other data sets on the same volume or 3990 string. If this condition persists, move these non-cache data sets to other volumes or strings.

Collecting OAM Transaction Statistics

RMF collects transaction information from OAM to generate performance-related statistical reports. The transaction information includes a transaction class and associated transaction names.

Each transaction class represents a component of OAM and its activities:

OSREQ invocations of the OSREQ macro

OSMC activities performed by the OAM storage management component

Each transaction class has several transaction names associated with it. The transaction names represent actions that can be taken by each component of OAM. For example, a transaction name associated with OSREQ is DELETETB, which represents the completion of an OSREQ DELETE request, where the primary copy of the object resides on tape and the backup copy of the object exists and was also deleted.

You can indicate the type of transaction to report by defining a report performance group. For example, if you are interested in OSREQ DELETE transactions, then define OSREQ DELETETB as a report performance group.

To generate RMF reports, you must define the report performance groups in the installation control specifications, the IEAICSxx member of SYS1.PARMLIB. This is done by adding a SUBSYS specification in the IEAICSxx member. Figure 117 shows an example of how report performance groups are defined.

```
SUBSYS=OAM,                /* OAM REPORT PERF GROUP    */
TRXCLASS=OSREQ,RPGN=400    /* OSREQ TRANSACTION CLASS  */
TRXNAME=ACCESS,RPGN=401    /* OSREQ ACCESS              */
TRXNAME=CHANGE,RPGN=402    /* OSREQ CHANGE              */
TRXNAME=DELETED,RPGN=403   /* OSREQ DELETE FROM DASD   */
TRXNAME=DELETED0,RPGN=404  /* OSREQ DELETE FROM OPTICAL */
TRXNAME=DELETET,RPGN=405   /* OSREQ DELETE FROM TAPE   */
TRXNAME=QUERY,RPGN=406     /* OSREQ QUERY               */
TRXNAME=QUERYG,RPGN=407    /* OSREQ GENERIC QUERY       */
TRXNAME=RETRVBO,RPGN=408   /* OSREQ RETRIEVE BACKUP OPT */
TRXNAME=RETRVBT,RPGN=409   /* OSREQ RETRIEVE BACKUP TAPE */
TRXNAME=RETRVPD,RPGN=410   /* OSREQ RETRIEVE PRIMARY DASD */
TRXNAME=RETRVPO,RPGN=411   /* OSREQ RETRIEVE PRIMARY OPT */
TRXNAME=RETRVPT,RPGN=412   /* OSREQ RETRIEVE PRIMARY TAPE */
TRXNAME=STORDASD,RPGN=413  /* OSREQ STORE TO DASD      */
TRXNAME=STOROPT,RPGN=414   /* OSREQ STORE TO OPT       */
TRXNAME=STORTAPE,RPGN=415  /* OSREQ STORE TO TAPE      */
TRXNAME=UNACCESS,RPGN=416  /* OSREQ UNACCESS           */
```

Figure 117. Defining OAM Performance Groups

For more information on defining performance groups, see *DFSMS/MVS V1R2 OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Creating and Interpreting RMF Transaction Reports

You can obtain two types of RMF transaction reports for the performance groups defined: the Workload Activity Report and the Transaction Activity Report.

The Workload Activity Report displays information on the amount of workload carried or the resources consumed by the transactions during the measured interval. The contents of the report include:

- The average number of active transactions during the measurement interval.
- The average number of transactions resident in central storage during the interval.
- The number of transactions that ended during the interval.
- The average elapsed time of ended transactions.
- The standard deviation of the average elapsed time of ended transactions.

Figure 118 shows a sample RMF Monitor I Workload Activity Report.

WORKLOAD ACTIVITY														PAGE 1	
MVS/ESA		SYSTEM ID 3090		DATE 09/01/93		INTERVAL 12.26.657									
SP4.3.0		RPT VERSION 4.3.0		TIME 11.15.26		IPS = IEAIPS04						SU/SEC=1162.3			
OPT = IEAOPT00		SUMMARY BY PERFORMANCE GROUP		SERVICE DEFINITION COEFFICIENTS		IOU = 5.0 CPU = 10.0 SRB = 10.0						MSO = 3.0000			
ICS = IEAICSTU												AVG TRANS. TIME,			
TIME				AVERAGE ABSORPTION,		PAGE-IN RATES		STORAGE		TRANSACTIONS		STD. DEVIATION			
PGN	PGP	DMN	SLICE	INTERVAL	SERVICE	AVG TRX SERV RATE,	TCB+SRB SECONDS, %					HHH.MM.SS.TTT			
0000	ALL	ALL	ALL	IOC	1,170	ABSRPTN	11	SINGLE	0.02	AVERAGE	101.33	AVG	7.00	TRX 000.00.00.000	
				CPU	25423	TRX SERV	11	BLOCK	0.03			MPL	7.00	SD 000.00.00.000	
				MSO	20602	TCB	2.1	HSP	0.00	TOTAL	709.80	ENDED	0		
				SRB	11766	SRB	1.0	HSP MISS	0.00	CENTRAL	709.80	END/SEC	0.00	QUE 000.00.00.000	
				TOT	58961	TCB+SRB%	0.4	EXP SNGL	0.00	EXPAND	0.00	#SWAPS	2		
				PER SEC	78			EXP BLK	0.00					TOT 000.00.00.000	
SUBSYS = OAM		TRXCLASS =		ACCTINFO = NO											
USERID =		TRXNAME =													
0001	ALL	ALL	ALL	IOC	23091	ABSRPTN	8	SINGLE	0.01	AVERAGE	142.10	AVG	22.99	TRX 000.00.00.000	
				CPU	31226	TRX SERV	8	BLOCK	0.07			MPL	22.99	SD 000.00.00.000	
				MSO	74840	TCB	2.6	HSP	0.00	TOTAL	3,268.5	ENDED	0		
				SRB	10387	SRB	0.8	HSP MISS	0.00	CENTRAL	3,268.5	END/SEC	0.00	QUE 000.00.00.000	
				TOT	139.5K	TCB+SRB%	0.4	EXP SNGL	0.00	EXPAND	0.00	#SWAPS	0		
				PER SEC	186			EXP BLK	0.00					TOT 000.00.00.000	
0002	ALL	ALL	ALL	IOC	17730	ABSRPTN	1,577	SINGLE	2.48	AVERAGE	343.16	AVG	0.12	TRX 000.00.01.924	
				CPU	40034	TRX SERV	1,522	BLOCK	1.40			MPL	0.12	SD 000.00.03.470	
				MSO	80126	TCB	3.4	HSP	0.00	TOTAL	41.46	ENDED	61		
				SRB	4,399	SRB	0.3	HSP MISS	0.00	CENTRAL	41.46	END/SEC	0.08	QUE 000.00.11.126	
				TOT	142.3K	TCB+SRB%	0.4	EXP SNGL	0.00	EXPAND	0.00	#SWAPS	86		
				PER SEC	190			EXP BLK	0.00					TOT 000.00.13.050	
0003 ALL ALL ALL		THRU		ZEROS											
0099															
SUBSYS = OAM		TRXCLASS = OSREQ		ACCTINFO = NO											
USERID =		TRXNAME =													
0400	ALL	ALL	ALL	IOC	0	ABSRPTN	0	SINGLE	0.00	AVERAGE	0.00	AVG	0.00	TRX 000.00.00.616	
				CPU	0	TRX SERV	0	BLOCK	0.00			MPL	0.00	SD 000.00.01.248	
				MSO	0	TCB	0.0	HSP	0.00	TOTAL	0.00	ENDED	51		
				SRB	0	SRB	0.0	HSP MISS	0.00	CENTRAL	0.00	END/SEC	0.06	QUE 000.00.00.000	
				TOT	0	TCB+SRB%	0.0	EXP SNGL	0.00	EXPAND	0.00	#SWAPS	0		
				PER SEC	0			EXP BLK	0.00					TOT 000.00.00.000	
SUBSYS = OAM		TRXCLASS =		ACCTINFO = NO											
USERID =		TRXNAME = ACCESS													

Figure 118. Reporting OAM Workload Activity (page 1 of 4)

WORKLOAD ACTIVITY

MVS/ESA SYSTEM ID 3090 DATE 09/01/93 INTERVAL 12.26.657
 SP4.3.0 RPT VERSION 4.3.0 TIME 11.15.26 IPS = IEAIPS04
 OPT = IEAOPT00 SUMMARY BY PERFORMANCE GROUP SERVICE DEFINITION COEFFICIENTS SU/SEC=1162.3
 ICS = IEAICSTU IOC = 5.0 CPU = 10.0 SRB = 10.0 MSO = 3.0000

PGN	PGP	DMN	SLICE TIME GROUP	INTERVAL	SERVICE	AVERAGE ABSORPTION, AVG TRX SERV RATE, TCB+SRB SECONDS, %	PAGE-IN RATES	STORAGE	TRANSACTIONS	AVG TRANS. TIME, STD. DEVIATION HHH.MM.SS.TTT
0401	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE 0.00	AVERAGE 0.00	AVG 0.00	TRX 000.00.01.039
				CPU	0	TRX SERV 0	BLOCK 0.00		MPL 0.00	SD 000.00.00.178
				MSO	0	TCB 0.0	HSP 0.00	TOTAL 0.00	ENDED 16	
				SRB	0	SRB 0.0	HSP MISS 0.00	CENTRAL 0.00	END/SEC 0.02	QUE 000.00.00.000
				TOT	0	TCB+SRB% 0.0	EXP SNGL 0.00	EXPAND 0.00	#SWAPS 0	
				PER SEC	0		EXP BLK 0.00			TOT 000.00.00.000
0402	ALL	ALL	ALL	ZEROS						

SUBSYS = OAM TRXCLASS = ACCTINFO = NO
 USERID = TRXNAME = DELETED

0403	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE 0.00	AVERAGE 0.00	AVG 0.00	TRX 000.00.00.062
				CPU	0	TRX SERV 0	BLOCK 0.00		MPL 0.00	SD 000.00.00.079
				MSO	0	TCB 0.0	HSP 0.00	TOTAL 0.00	ENDED 5	
				SRB	0	SRB 0.0	HSP MISS 0.00	CENTRAL 0.00	END/SEC 0.00	QUE 000.00.00.000
				TOT	0	TCB+SRB% 0.0	EXP SNGL 0.00	EXPAND 0.00	#SWAPS 0	
				PER SEC	0		EXP BLK 0.00			TOT 000.00.00.000
0404	ALL	ALL	ALL	ZEROS						

THRU 0408

SUBSYS = OAM TRXCLASS = ACCTINFO = NO
 USERID = TRXNAME = QUERY

0409	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE 0.00	AVERAGE 0.00	AVG 0.00	TRX 000.00.00.667
				CPU	0	TRX SERV 0	BLOCK 0.00		MPL 0.00	SD 000.00.01.259
				MSO	0	TCB 0.0	HSP 0.00	TOTAL 0.00	ENDED 5	
				SRB	0	SRB 0.0	HSP MISS 0.00	CENTRAL 0.00	END/SEC 0.00	QUE 000.00.00.000
				TOT	0	TCB+SRB% 0.0	EXP SNGL 0.00	EXPAND 0.00	#SWAPS 0	
				PER SEC	0		EXP BLK 0.00			TOT 000.00.00.000

SUBSYS = OAM TRXCLASS = ACCTINFO = NO
 USERID = TRXNAME = QUERYG

Figure 119. Reporting OAM Workload Activity (page 2 of 4)

WORKLOAD ACTIVITY

PAGE 3

MVS/ESA SYSTEM ID 3090 DATE 09/01/93 INTERVAL 12.26.657
 SP4.3.0 RPT VERSION 4.3.0 TIME 11.15.26 IPS = IEAIPS04
 OPT = IEAOPT00 SUMMARY BY PERFORMANCE GROUP SERVICE DEFINITION COEFFICIENTS SU/SEC=1162.3
 ICS = IEAICSTU IOC = 5.0 CPU = 10.0 SRB = 10.0 MSO = 3.0000

PGN	PGP	DMN	Slice TIME GROUP	INTERVAL	SERVICE	AVERAGE ABSORPTION, AVG TRX SERV RATE, TCB+SRB SECONDS, %	PAGE-IN RATES	STORAGE	TRANSACTIONS	STD. DEVIATION HHH.MM.SS.TTT
0410	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE 0.00	AVERAGE 0.00	AVG 0.00	TRX 000.00.01.305
				CPU	0	TRX SERV 0	BLOCK 0.00		MPL 0.00	SD 000.00.00.000
				MSO	0	TCB 0.0	HSP 0.00	TOTAL 0.00	ENDED 1	
				SRB	0	SRB 0.0	HSP MISS 0.00	CENTRAL 0.00	END/SEC 0.00	QUE 000.00.00.000
				TOT	0	TCB+SRB% 0.0	EXP SNGL 0.00	EXPAND 0.00	#SWAPS 0	
				PER SEC	0		EXP BLK 0.00			TOT 000.00.00.000
0411	ALL	ALL	ALL	ZEROS						
THRU 0412										

SUBSYS = OAM TRXCLASS = ACCTINFO = NO
 USERID = TRXNAME = RETRVPD

0413	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE 0.00	AVERAGE 0.00	AVG 0.00	TRX 000.00.00.162
				CPU	0	TRX SERV 0	BLOCK 0.00		MPL 0.00	SD 000.00.00.208
				MSO	0	TCB 0.0	HSP 0.00	TOTAL 0.00	ENDED 2	
				SRB	0	SRB 0.0	HSP MISS 0.00	CENTRAL 0.00	END/SEC 0.00	QUE 000.00.00.000
				TOT	0	TCB+SRB% 0.0	EXP SNGL 0.00	EXPAND 0.00	#SWAPS 0	
				PER SEC	0		EXP BLK 0.00			TOT 000.00.00.000
0414	ALL	ALL	ALL	ZEROS						
THRU 0419										

SUBSYS = OAM TRXCLASS = ACCTINFO = NO
 USERID = TRXNAME = STORDASD

0420	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE 0.00	AVERAGE 0.00	AVG 0.00	TRX 000.00.01.735
				CPU	0	TRX SERV 0	BLOCK 0.00		MPL 0.00	SD 000.00.03.672
				MSO	0	TCB 0.0	HSP 0.00	TOTAL 0.00	ENDED 5	
				SRB	0	SRB 0.0	HSP MISS 0.00	CENTRAL 0.00	END/SEC 0.00	QUE 000.00.00.000
				TOT	0	TCB+SRB% 0.0	EXP SNGL 0.00	EXPAND 0.00	#SWAPS 0	
				PER SEC	0		EXP BLK 0.00			TOT 000.00.00.000
0421	ALL	ALL	ALL	ZEROS						
THRU 0422										

SUBSYS = OAM TRXCLASS = ACCTINFO = NO
 USERID = TRXNAME = UNACCESS

Figure 120. Reporting OAM Workload Activity (page 3 of 4)

WORKLOAD ACTIVITY

MVS/ESA		SYSTEM ID 3090		DATE 09/01/93		INTERVAL 12.26.657					
SP4.3.0		RPT VERSION 4.3.0		TIME 11.15.26		IPS = IEAIPS04					
OPT = IEAOPT00		SUMMARY BY PERFORMANCE GROUP				SERVICE DEFINITION COEFFICIENTS		SU/SEC=1162.3			
ICS = IEAICSTU						IOC = 5.0 CPU = 10.0 SRB = 10.0		MSO = 3.0000			
PGN	PGP	DMN	Slice	Interval	Service	Average Absorption,	Page-In Rates	Storage	Transactions	Avg Trans. Time,	
			GROUP			AVG TRX SERV RATE,				STD. DEVIATION	
						TCB+SRB SECONDS, %				HHH.MM.SS.TTT	
0423	ALL	ALL	ALL	IOC	0	ABSRPTN 0	SINGLE	0.00	AVG	0.00	TRX 000.00.00.051
				CPU	0	TRX SERV 0	BLOCK	0.00	MPL	0.00	SD 000.00.00.053
				MSO	0	TCB 0.0	HSP	0.00	TOTAL	0.00	ENDED 17
				SRB	0	SRB 0.0	HSP MISS	0.00	CENTRAL	0.00	END/SEC 0.02
				TOT	0	TCB+SRB% 0.0	EXP SNGL	0.00	EXPAND	0.00	#SWAPS 0
				PER SEC	0		EXP BLK	0.00			TOT 000.00.00.000

Figure 121. Reporting OAM Workload Activity (page 4 of 4)

The transaction activity report enables you to evaluate the effectiveness of the performance objective in the Installation Performance Specification (IPS) and the workload classification in the Installation Control Specification (ICS). The report is a "snapshot" of transaction activity at the time the report was requested. Figure 122 shows a sample RMF Monitor II Transaction Activity Report. The report, shown by performance group period, include:

- The transaction rate (the number of ended transaction per second).
- The average elapsed time of each ended transaction.

The data in the report is collect by SRM and passed to RMF when a Monitor I session, that is measuring workload activity, is active. Thus, the transaction activity report can be obtained only when a Monitor I session with the WKLD option is running at the same time as the report request. For more information, see *MVS/ESA Resource Measurement Facility Version 4 Monitor I and II Reference and User's Guide*.

11:24:28		ICS=IEAICSTU		IPS=IEAIPS04		CPU= 3 UIC=183 PFR= 0		TRX T		
SUBSYS	TRXCLASS	USERID	TRXNAME	ACCT	PERF	INFO	GRP	PER	RATE	HHH.MM.SS.TTT
						C	2	1	0.079	000.00.00.134
								2	0.004	000.00.00.699
								3	0.041	000.00.05.426
OAM	OSREQ			NO R	400			1	0.118	000.00.00.616
OAM			ACCESS	NO R	401			1	0.037	000.00.01.039
OAM			DELETED	NO R	403			1	0.011	000.00.00.062
OAM			QUERY	NO R	409			1	0.011	000.00.00.667
OAM			QUERYG	NO R	410			1	0.002	000.00.01.305
OAM			RETRVPD	NO R	413			1	0.004	000.00.00.162
OAM			STORDASD	NO R	420			1	0.011	000.00.01.735
OAM			UNACCESS	NO R	423			1	0.039	000.00.00.051

Figure 122. Reporting OAM Transaction Activity

The data RMF collects can be captured in SMF data sets during the RMF measurement interval. The data can then be used to determine long-term trends and perform capacity planning for your installation. For more information on recording RMF data in SMF records, see *MVS/ESA SPL: SMF*.

Collecting Cost Accounting Information

We recommend using DCOLLECT to gather data for DASD cost accounting in the DFSMS environment. DCOLLECT is easier to use, provides more information, and requires less processing time than using VTOC information, collecting SMF records, or using DFSMSHsm features alone. You can also feed DCOLLECT output to SLR programs to generate reports or bills.

You can also use DFSMSrmm utilities to collect information on tape data sets and volumes for required tape cost accounting. For example, you can obtain information on the amount of volumes users own and bill the users for that number of tape volumes. For more information on using DFSMSrmm utilities for reporting purposes, see *DFSMS/MVS V1R2 DFSMSrmm Implementation and Customization Guide*.

There are a number of ways to establish cost accounting models. This section presents sample methods for determining DASD service charges and storage charges.

DCOLLECT does not collect information on data that is created and deleted within a job.

Determining Service Charges

You can base your installation service charges on allocated space, and performance and availability characteristics. Because these characteristics are associated with management class and storage class, you can establish service charge rates based on these classes.

Establish Service Charge Rates

Create a charge rate table by listing the management and storage classes, and the rates for each class. You might need to work with your finance or business plans department to determine the appropriate rates for the classes.

Figure 123 shows a sample service rate charge table. It lists sample management and storage classes and corresponding installation-determined charge rates.

Figure 123. Establishing Service Charge Rates

Class Name	Charge Rate	Description
SCNORM	1.0	Standard storage class
SCFAST	1.2	Cached storage class
SCCRIT	2.0	Dual copy storage class
MCNORM	1.0	Standard management class
MCXBKUP	1.2	Extra backup management class
MCNOMIG	1.3	No migration management class
MCNOBKUP	0.9	No backup management class
No SC	1.1	Non-system-managed, DFSMSHsm-managed
Volume	1.8	User-owned volume

Calculating Service Charges

You can use various units to calculate service charges. Figure 124 shows a formula you can use if you are charging by allocated megabyte units.

```
MB x storage class rate x management class rate
```

Figure 124. Calculating Service Charge

To bill service charges by project, for example, you can use the number of megabytes allocated by a project and calculate the charge using the formula.

Producing a Service Charge Report

The DCOLLECT active data set records contain the allocated size of the data set, the storage class, and management class information. Using the service charge formula and the DCOLLECT active data set records, you can generate a service charge report. We recommend using SLR to generate the report because you can provide SLR controlling parameters to select appropriate DCOLLECT data records to calculate service charges.

General-Use Programming Interface

Figure 125 shows a sample program used to produce the report in Figure 126 on page 185. The program uses SLR control parameters to select project name, management class name, storage class name, and allocated space information from DCOLLECT data records.

```
REPORT DASD_ACCT

  SEL C(PROJECT ST_CLASS MGT_CLASS ALLOC_XP)

  SEL R((YEAR=*) AND (MONTH=*) AND
        (DAY=*-1) AND (PROJECTϕ=TOT) AND
        (ST_CLASSϕ=TOT) AND (MGT_CLASSϕ=TOT))

  PRINT TABLE

ENDREP
```

Figure 125. Using SLR Program to Produce Service Charge Reports

End of General-Use Programming Interface

Figure 126 on page 185 shows the generated report. The first line of each project shows the amount of non-system-managed storage allocated. These are usually privately-owned volumes that are not system-managed, or volumes in a non-system-managed storage pool.

PROJECT	STORCLAS	MGMTCLAS	ALLOC_SP M-BYTES	SERVICE CHARGE
*****	*****			
PROJDC0			0.0	\$ 0.00
	SCNORM	MCXBKUP	213.7	\$ 256.44
		MCNORM	1538.5	\$1538.50
		MCNOMIG	13.4	\$ 17.42
	SCFAST	MCNOBKUP	95.3	\$ 148.67
PROJMF2			0.0	\$ 0.00
	SCNORM	MCNORM	1568.4	\$1568.40
		MCXBKUP	33.4	\$ 40.08
	SCCRIT	MCNOMIG	31.2	\$ 81.12
PROJX15			75.7	\$ 136.26 (1.8 rate)
	SCNORM	MCNORM	871.2	\$ 871.20
	SCFAST	MCNOMIG	5.2	\$ 8.11
		MCXBKUP	43.0	\$ 61.92
	SCCRIT	MCNOMIG	105.6	\$ 274.56
		MCXBKUP	19.9	\$ 47.75

Figure 126. Reporting Service Charges by Project

Provide this type of report to your company's finance department. Using this service charge method, the finance department then cross-charges the costs for storage services to the departments who own these projects.

Determining Storage Charges

You can base your installation storage charges on the actual storage used by active, migrated, and backed up data. The cost rate is related to where the data resides in the storage hierarchy.

Developing Service Charge Rates

Create a charge rate table by listing the data types and the rates for each type. You might need to work with your finance or business plans department to determine the appropriate rates.

Figure 127 shows a sample storage rate charge table. It lists the data types and corresponding installation-determined charge rates.

Figure 127. Establishing Storage Charge Rates

Data Type	Charge Rate
Active	1.0
Migrated	0.6
Backup	0.2

Calculating Storage Charges

You can use various units to calculate storage charges. Figure 128 shows a formula calculating the charge, based on the number of megabytes of storage used.

$$\begin{aligned} \text{Charge} = & (\text{MB Active} \times \text{rate for Active}) + \\ & (\text{MB Migrated} \times \text{rate for Migrated}) + \\ & (\text{MB Backup} \times \text{rate for Backup}) \end{aligned}$$

Figure 128. Calculating Service Charge

To bill service charges by project, for example, you can use the number of megabytes used by a project and calculate the charge using the formula.

Producing a Storage Charge Report

The DCOLLECT active data set, migration, and backup records contain the data set's size information. Using the service charge formula and these DCOLLECT records, you can generate a storage charge report. We recommend using SLR to generate the report because you can provide SLR controlling parameters to select appropriate DCOLLECT records to calculate storage charges.

General-Use Programming Interface

Figure 129 shows a sample program used to produce the report in Figure 130 on page 187. The program uses SLR control parameters to select project name, allocated space, migrated space, and backup space information from DCOLLECT records.

```
REPORT DASD_ACCT

SEL C(PROJECT ALLOC_SP MIGR_SP BCK_SP)

SEL R((YEAR=*) AND (MONTH=*) AND
      (DAY=*-1) AND (PROJECTϕ=TOT) AND
      (ST_CLASSϕ=TOT) AND (MGT_CLASSϕ=TOT))

PRINT TABLE

ENDREP
```

Figure 129. Using SLR Program to Produce Storage Charge Reports

End of General-Use Programming Interface

Figure 130 on page 187 shows what the generated report looks like. The report lists the amount of storage used by each project.

PROJECT	ALLOC_SP M-BYTES	MIGRAT_SP M-BYTES	BACKUP_SP M-BYTES
*****	*****	*****	*****
PROJDC0	889.4	527.1	444.4
PROJMF2	705.9	384.7	542.4
PROJX15	618.5	361.6	140.5

Figure 130. Reporting Storage Charges by Project

Analyzing Tape Use to Reduce Tape Mounts

You can use the volume mount analyzer to help you track tape mount activity, monitor tape media use, and implement tape mount management techniques in your installation. The volume mount analyzer produces reports to help you identify the data sets, jobs, or programs that are causing excessive tape mount activity. You can then use tape mount management techniques on selected applications to reduce the number of mounts and maximize the use of your tape media.

This section explains briefly the steps involved in using the volume mount analyzer and provides scenarios, with sample reports, to show some strategies you can use in routine tape mount analysis. Details on how to use the volume mount analyzer are covered in *DFSMS/MVS V1R2 Using the Volume Mount Analyzer*. Familiarity with the tape mount management concepts and implementation techniques described in *MVS/ESA SML: Implementing System-Managed Storage* is assumed.

Using the Volume Mount Analyzer

The volume mount analyzer consists of the following programs:

GFTAXTR Extracts information from SMF records and builds a file containing condensed SMF information to be processed by GFTAVMA.

GFTAVMA Performs the analysis and produces reports.

This section briefly describes the steps involved in performing a tape mount analysis and what role each program has in the process.

Collecting and Extracting SMF Data for Analysis

The volume mount analyzer uses SMF data for analysis. Both system-managed and non-system-managed tape data are eligible for the analysis.

To generate an accurate report on tape mount activity, we recommend you use the SMF data collected over 30 days, ending on a business day. This period must cover peak loads as well as routine processing.

You use the GFTAXTR program to read and extract the SMF records. GFTAXTR requires the following SMF records as input:

- Type 14 - Data set open for input
- Type 15 - Data set open for output
- Type 21 - Tape volume demount
- Type 30 - Address space record

The extraction process totally reconstructs tape job streams and builds a data set containing condensed data records for tape mount analysis. In addition, the extraction process provides a summary report that shows the number of tape mounts and some of the tape data sets' characteristics.

For detailed descriptions of the SMF record types, see *MVS/ESA System Management Facility (SMF)*. For more information on GFTAXTR requirements, see *DFSMS/MVS V1R2 Using the Volume Mount Analyzer*.

Categorizing Tape Data

The GFTAXTR program also assigns a usage type to all data sets it analyzes, based on the data set naming convention, data set attributes, or patterns of read/write activity. The usage types are later used to determine tape mount management categories so that data sets will be assigned the proper management class for DFSMS/MVS management.

GFTAXTR assigns the following usage types:

HSM

Data sets written to tape by DFSMSHsm. DFSMSHsm transfers large amounts of data to tape and uses tape very differently from other applications. Therefore, the volume mount analyzer must separate these data sets. If this data type is allowed to fall into the ACTIVE or BACKUP group, DFSMSHsm data can drastically influence any volume mount analyzer-derived information and result in inaccurate conclusions during data analysis.

Temporary

Data sets allocated with DISP=(NEW,DELETE), or with the name, SYS9*.R*.**. These data sets exist only for the life of the job.

Backup

Data sets that are opened for output only, and created by backup programs (such as DFSMSDss). Many of the data sets in this category are referenced on a single calendar day, and can be migrated directly to migration level 2 after that date.

Single

Data sets that are referenced within a single calendar day and can be migrated to migration level 2 after that day. The reference activity can be a read activity.

BCOPY

Data sets with an access of 1 open for input and 1 open for output. BCOPY includes recovered backups, or backups created for disaster recovery (off-site).

Active

Data sets that do not fall into any of the previous categories. These data sets exhibit a mixture of OPENS for INPUT and OUTPUT activity. They are typically small and have a relatively higher number of tape-mounts than other categories.

When generating reports, GFTAVMA categorizes all data sets according to the data set name, data set size, the pattern of read and write activity, the keywords you have specified, and, in some cases, the usage types assigned by GFTAXTR. If you do not agree with the usage types that GFTAXTR assigns, you can use GFTAVMA CLASSIFY keyword to override the type. This is the normal approach after you decide on a set of filters for an application. You can also use the various other GFTAVMA keywords to refine the output of the reports. Discussions on reporting strategies appear later in this section.

The data sets fall into one of six possible tape mount management categories. These categories are relevant to the tape mount reduction process and do not have the same definitions as the data types in the DFSMS environment.

HSM

Data sets created by DFSMSHsm and have the USAGE(HSM) attribute. These data sets already use tapes effectively, so the volume mount analyzer automatically separates these data sets.

Large

Data sets whose sizes are greater than or equal to the value specified by the LARGE keyword. The default is 600 MB.

Temporary

Data sets allocated with DISP=(NEW,DELETE), or with the name, SYS9*.R*.**.

Backup

Data sets built by a backup program and categorized as USAGE(BACKUP).

Active

Data sets that do not fall into any of the previous categories, or are categorized as active by the CLASSIFY keyword.

Exceptions

Data sets that are identified by the EXCLUDE filters and will be excluded from analysis.

Tape data sets that are good candidates for tape mount management are temporary, backup, and active data sets. DFSMSHsm and very large data sets are not candidates for tape mount management because they are large enough to go to tape directly and effectively use the space. Therefore, these data sets should be added to a tape storage group.

You can use the volume mount analyzer to help identify patterns for very large data sets in your installation. Typically, a small set of programs or applications are responsible for generating most of the large data sets. Volume mount analyzer can show which programs, jobs, and high-level qualifiers generate most of the tape mounts, and what portion of those are large data sets that are completely filling the tape volumes.

For more information on tape data categories or GFTAVMA keywords, see *DFSMS/MVS V1R2 Using the Volume Mount Analyzer*.

Producing Volume Mount Analyzer Reports

Run the GFTAVMA program to analyze tape mount categories and suggest data for system management. You need to make multiple GFTAVMA runs against the GFTAXTR output to refine your analysis.

Determining Filter Criteria: Due to the large amount of input data, you might need to use GFTAVMA keywords to filter the information and produce a useful report. Some of the keywords you can specify include:

- Account code
- Expiration date
- System ID
- Data set name
- Job name
- Program name

Unit name (generic) or unit address
Volume name
Data set size
Single file or multiple file data sets
Single volume or multiple volume data sets
Usage pattern (temporary, single, bcopy, HSM, backup, active)

We recommend using program name as the filter criteria, because one or a few programs tend to be responsible for many tape mounts. In addition, data sets stacked on a single tape are usually put there by the same program. Therefore, if you use program name as the filter, you can get a complete set of multi-file data sets, assuming that you already determined that the data sets are suitable for interception.

The filter list you develop is later used in data class ACS routines. Therefore, we recommend that you keep the list small so it is easier to maintain.

For information on writing the tape mount management ACS routines, see *MVS/ESA SML: Implementing System-Managed Storage*.

Generating Volume Mount Analyzer Reports: GFTAVMA automatically produces the following summary reports for each run:

GFTASRT1 Input Analysis Report
GFTASRT2 Tape Volume Statistics Report and Tape Volume Size Range Report
GFTASRT3 Tape Data Set Statistics Report and Tape Data Set Size Range Report

GFTAVMA also generates the optional detailed analysis reports if you specify the REPORT keyword in the JCL. The detailed reports you can produce include:

- Usage (concurrent tape drive allocations and concurrent mounts)
- Volume (supplement to the GFTASRT2 report)
- Data Set (supplement to the GFTASRT3 report)
- Maximum Gigabyte (maximum tape data flows at hourly and daily)
- Top (top tape mount contributors in the installation)
- Estimate (tape mount savings and DASD cost estimates)
- Improved Data Set Recording Capability (device savings estimates)
- Management Class (possible management class values and simulations)

To generate graphs from GFTAVMA output, specify the CHART keyword in the GFTAVMA run. The CHART keyword enables GFTAVMA to generate a sequential data set which can be used as input to a graphics package. You can use the Interactive Chart Utility (part of GDDM*-PGF 5668-812), or an equivalent graphics program, to produce graphs representing GFTAVMA output.

Tracking Tape Mounts

You can use different volume mount analyzer reports to help you track tape mounts in your installation. This section explains the use of the top tape mount contributor report. A discussion of the Estimate and Management Class Reports is also included to show how you can prepare for tape mount management.

Reporting Top Contributors

By using the REPORT and TOP keywords, you can generate reports on top tape mount contributors in your installation. You can then use the reports to plan how to reduce the number of tape mounts using tape mount management techniques.

The following scenario shows some Top reports. The control statement in Figure 131 produced the sample reports in this section.

```
REPORT(TOP(PGM,JOB,HLQ,PERCENT(50)))
```

Figure 131. Producing Volume Mount Analyzer Top Reports

In this medium size installation, the TOP keyword lists the top tape mount contributors based on three filtering parameters: program name, job name, and data set name highest-level qualifier. PERCENT(50) specifies that contributors listed must cause up to 50% of the tape mounts in the installation. The report is generated with the assumption that large data sets are 600 MB or greater. In the scenario, the SMF data is collected over 30 days.

Top Programs Contributing to Tape Mounts: Figure 132 shows the report generated on the program name filter, PGM.

GFTASRT3 -- REPORT TOP PROGRAM NAMES -- 50% MOUNTS -- "LARGE" = 600 MB															
RANK	PROGRAM	# DSNs	% TOT	CUM DSN	% TOT	> LARGE	AVG SIZE	# MOUNTS	% TOT	CUM MNTS	% TOT	SIZE (GB)	% TOT	CUM GB	% TOT
1	IDCAMS	12386	30.0	12386	30.0	0	16.0	6611.6	17.3	6611.6	17.3	198.3	15.1	198.3	15.1
2	IEBGENER	3273	7.9	15659	38.0	3	7.9	5429.7	14.2	12041.3	31.4	25.8	2.0	224.1	17.1
3	SORT	5966	14.5	21625	52.4	0	51.4	4223.9	11.0	16265.3	42.5	306.8	23.4	530.8	40.5
4	DUMP	146	0.4	21771	52.8	130	837.5	2673.0	7.0	18938.3	49.4	122.3	9.3	653.1	49.9
5	BILLWKY1	5218	12.7	26989	65.4	0	6.5	2447.9	6.4	21386.2	55.8	34.1	2.6	687.2	52.5

GFTAVMA PROGRAM COMPLETED AT 11:52:31 A.M. ON 09/11/1992 -- RETURN CODE = 0

Figure 132. Displaying Top Program Names That Produce 50% of Tape Mounts

The report in Figure 132 shows some important points:

- Program name is a very efficient filter to use. The five programs listed account for a total of 55.8% of the installation's tape mounts during the examined period (see bottom numbers in the CUM MNTS and % TOT column). According to the CUM MNTS column, this is a total of 21 386.2 tape mounts, with IDCAMS causing the most number of mounts.
- The data sets from the DUMP program can be best managed by sending them directly to tape. The DUMP program generates data sets with an average size of 837.5 MB. Most of these data sets (130 out of 146) are greater than the LARGE value of 600 MB. Therefore, it is best to consider the entire application as a tape mount management category of LARGE.

You might want to generate an Estimate Report to calculate the tape mount or volume savings from sending the data sets directly to tape. To create the Estimate Report, submit the statement in Figure 133 on page 192 with a GFTAVMA run:

```
REP(EST) PGM(INC(DUMP)) LARGE(0)
```

Figure 133. Calculating Savings After Refining the Initial Assessment

For more information about the Estimate Report, see “Estimating Tape Mount Savings” on page 194.

Note that there might be other reasons that make the data sets bad tape mount management candidates. For example, they might be off-site exchange data sets. The top report in this section is used to provide an initial analysis of what data sets to exclude in the sample installation.

General-Use Programming Interface

- The remaining programs generate data sets that are eligible for tape mount management. You can construct a filter list for the tape mount management data class ACS routine for the sample installation. Figure 134 shows a portion of the routine, filtering on the program names listed in Figure 132 on page 191.

```
FILTLIST PROGS  INCLUDE('IDCAMS','IEBGENER','SORT','BILLWKY1')
               .
               .
               .
               WHEN (&PGM = &PROGS && &DSN = &BACKUP)

               SET &DATACLAS = 'TMMBKUP'
```

Figure 134. Writing Program Name Filter List for Data Class ACS Routine

End of General-Use Programming Interface

For more information on filter lists, data classes and groups, and ACS routines used in tape mount management, see *MVS/ESA SML: Implementing System-Managed Storage*.

Top Jobs Contributing to Tape Mounts: Figure 135 on page 193 shows a report generated on the job name filter, JOB.

GFTASRT3 -- REPORT TOP JOB NAMES														-- 50% MOUNTS -- "LARGE" = 600 MB		
RANK	JOBNAME	# DSNS	% TOT	CUM DSN	% TOT	> LARGE	AVG SIZE	# MOUNTS	% TOT	CUM MNTS	% TOT	SIZE (GB)	% TOT	CUM GB	% TOT	
1	BCOL02D	1038	2.5	1038	2.5	0	12.0	1622.0	4.2	1622.0	4.2	12.4	0.9	12.4	0.9	
2	DUMPHY	29	0.1	1067	2.6	0	30.4	1104.0	2.9	2726.0	7.1	0.9	0.1	13.3	1.0	
3	BASN02D	703	1.7	1770	4.3	0	34.3	914.9	2.4	3640.9	9.5	24.1	1.8	37.4	2.9	
/																
61	BJDC104D	2	0.0	17723	43.0	2	619.0	150.0	0.4	18978.5	49.5	1.2	0.1	672.9	51.4	
62	BJDC156D	2	0.0	17725	43.0	2	700.0	150.0	0.4	19128.5	49.9	1.4	0.1	674.3	51.5	
63	BJDC157D	2	0.0	17727	43.0	2	843.0	150.0	0.4	19278.5	50.3	1.7	0.1	676.0	51.6	

Figure 135. Displaying Top Job Names That Produce 50% of Tape Mounts

Figure 135 shows the first three and last three entries of a report that contains 63 job names. The jobs do not appear to cause significant tape mounts, because it takes more than 60 entries to account for 50% of the installation's tape mounts. Therefore, JOB is probably not a good filter criteria to use in this installation.

Top Highest-Level Qualifiers Contributing to Tape Mounts: Figure 136 shows a report generated on the data name highest-level qualifier filter, HLQ.

GFTASRT3 -- REPORT TOP HIGH-LEVEL QUALIFIERS -- 50% MOUNTS -- "LARGE" = 600 MB															
RANK	HLQ	# DSNS	% TOT	CUM DSN	% TOT	> LARGE	AVG SIZE	# MOUNTS	% TOT	CUM MNTS	% TOT	SIZE (GB)	% TOT	CUM GB	% TOT
1	PAYROLL	671	1.6	671	1.6	140	235.5	4179.3	10.9	4179.3	10.9	158.0	12.1	158.0	12.1
2	FINANCE	5969	14.5	6640	16.1	0	9.8	2859.9	7.5	7039.2	18.4	58.7	4.5	216.7	16.5
3	PARTS	2768	6.7	9408	22.8	0	14.1	2719.9	7.1	9759.1	25.5	39.0	3.0	255.7	19.5
4	SALES	1721	4.2	11129	27.0	0	13.4	2717.3	7.1	12476.4	32.6	23.0	1.8	278.7	21.3
5	BILLING	4940	12.0	16069	39.0	10	49.3	2605.7	6.8	15082.1	39.4	243.3	18.6	522.0	39.9
6	ORDERS	2057	5.0	18126	44.0	1	57.3	2026.3	5.3	17108.4	44.7	117.8	9.0	639.8	48.8
7	ORDENTRY	2211	5.4	20337	49.3	4	19.9	1954.4	5.1	19062.8	49.8	44.0	3.4	683.8	52.2
8	IMS	1393	3.4	21730	52.7	1	7.3	1765.8	4.6	20828.6	54.4	10.2	0.8	694.0	53.0

Figure 136. Displaying Top HLQs That Produce 50% of Tape Mounts

The report in Figure 136 shows that these HLQs account for a total of 54.4% of the installation's tape mounts during the examined period (see bottom numbers in the CUM MNTS and % TOT column). According to the CUM MNTS column, there is a total of 20828.6 tape mounts. Therefore, applying tape mount management techniques on most of these data sets can reduce a significant number of tape mounts.

Some of the data sets with the HLQ of PAYROLL are too large to be good candidates for tape mount management. According to the report, 140 of the 671 PAYROLL data sets are greater than 600 MB. These data sets need to be filtered out and added to a tape storage group. One way you can do this is to lower the data set size to identify durable, large patterns for the PAYROLL data sets, as Figure 137 on page 194 shows.

```
DSN(INC(PAYROLL.**)) MINSIZE(400)
```

Figure 137. Filtering Out Data Sets by Data Set Name and Size

Before you begin constructing a filter list for the tape mount management data class ACS routine, you might want to obtain a clean list of TOP(HLQ) data sets that did not overlap with those already captured by TOP(PGM). The best way to do this is to make another run of the report, this time excluding the top programs first, as Figure 138 shows.

```
Rep(TOP(HLQ,PERCENT(50)))  
PGM(EXC(IDCAMS,IEBGENER, SORT,DUMP,BILLWKY1))
```

Figure 138. Obtaining a Clean List of TOP(HLQ) Data Sets

General-Use Programming Interface

You can construct a filter list for the tape mount management data class ACS routine for this sample installation. Figure 139 shows a portion of the routine, filtering on the HLQs listed in Figure 136 on page 193.

```
FILTLIST HLQS    INCLUDE('FINANCE','PARTS','SALES','BILLING',  
                        'IMS',ORDE*)  
.  
.  
.  
WHEN (&HLQ = &HLQS)  
  
SET &DATACLAS = 'TMMBKUP'
```

Figure 139. Writing HLQ Filter List for Data Class ACS Routine

End of General-Use Programming Interface

For more information on filter lists, data classes and groups, and ACS routines used in tape mount management, see *MVS/ESA SML: Managing Storage Groups*.

Estimating Tape Mount Savings

After you have determined good filters that capture top tape mount contributors, you might want to calculate the cost and savings of applying tape mount management techniques on each of the contributors.

You can use the Estimate Report to determine the potential cost and savings of implementing tape mount management techniques in your installation. The Estimate Report simulates the tape mount management environment by assuming certain values such as the target tape and DASD device used, the tape mount management categories assigned to data, and the management class migration and recall values assigned to the categories.

Before you run the Estimate Report, you might want to create a list of certain data types, such as off-site data or vault tapes, that would not benefit from tape mount management. These data types can be excluded from your Estimate Report.

To run the Estimate Report, use the REPORT(ESTIMATE) keywords. You can supply additional keywords on the control statement to focus on a particular subset of data.

Figure 140 shows the control statement used in the first-pass assessment for IDCAMS program, one of the five programs listed in “Top Programs Contributing to Tape Mounts” on page 191.

```
REPORT(ESTIMATE) TAPEDEV(3490E) TAPELEN(2)
PGM(INC(IDCAMS)) LARGE(999999999)
```

Figure 140. Producing Volume Mount Analyzer Estimate Reports

The specification, LARGE(999999999), makes sure that the estimate is based on the assumption that all IDCAMS data sets are to be managed through the tape mount management buffer. None will go directly to tape. Later, if you find that certain IDCAMS data sets can benefit by being moved directly to tape, then run an estimate report to exclude those data sets using the DSN and EXCLUDE keywords in the control statement.

Using the Savings/Cost Summary Report: The Estimate report consists of an Estimate Report legend that explains what assumptions are made in the run, a summary report, and three tables detailing the information outlined in the summary report. Figure 141 on page 196 shows the summary report generated.

SAVINGS/COST SUMMARY			
THE FOLLOWING CHART ASSUMES THAT THE CURRENT DATE IS THE LAST DAY OF THE INPUT SAMPLE AND DFHSM IS SET UP TO PROCESS THE TMM DASD BUFFER ONLY ONCE A DAY -- THE ACTUAL TMM DASD BUFFER MAY BE LESS IF DFHSM PROCESSING IS PERFORMED ON AN HOURLY BASIS			
STATISTICAL MOUNT SAVINGS:			
0.0	MOUNTS	(LARGE)	- 0.0%
0.0	MOUNTS	(DFHSM)	- 0.0%
6576.5	MOUNTS	(DFSMS)	- 99.5%

6576.5	MOUNTS		- 99.5%
- 600.0	MOUNTS	(MGMTCLAS)	- 9.1%

5976.5	MOUNTS	(TOTAL)	- 90.4%
STATISTICAL VOLUME SAVINGS:			
0.0	VOLUMES	(LARGE)	- 0.0%
0.0	VOLUMES	(DFHSM)	- 0.0%
1302.8	VOLUMES	(DFSMS)	- 97.4%

1302.8	VOLUMES		- 97.4%
- 0.8	VOLUMES	(MGMTCLAS)	- 0.1%

1302.0	VOLUMES	(TOTAL)	- 97.3%
DASD BUFFER COST:			
3.5	3390-3	VOLUMES (PRIMARY)	
0.1	3390-3	VOLUMES (MGMTCLAS)	
0.2		VOLUMES (+ 'FSPACE'====> 5%)	
0.0	3390-3	VOLUMES (DFHSM LEVEL 1)	
0.0	3390-3	VOLUMES (MGMTCLAS)	
0.0		VOLUMES (+ 'L1FSPACE'==> 5%)	

3.8		TOTAL VOLUMES	

Figure 141. Assessing Tape Mount Savings from the Estimate Summary Report

Initially, review the Savings/Cost Summary report to see if the savings outweigh the costs. Look at the bottom-line values to see how many mounts (90.4%) and volumes (97.3%) you can save and compare it to the amount of DASD volumes it would cost (3.8).

Note the DASD buffer calculations assume that DFSMSHsm migrates data from the buffer every 24 hours. The buffer requirements might be smaller if interval migration is performed hourly, especially for data sets in the backup category. You can run the Maximum Gigabyte Report to verify the hourly traffic.

To determine the net savings, you must consider the cost per mount (including the hardware, operators, the amount of physical space) compared to the cost of a DASD volume (not the entire device) per period of time.

Another number you need to look at is the recall rate (MGMTCLAS). Recalls are specific mounts, not scratch mounts, therefore, lower recall rates are more desirable. In general, a reasonable recall rate is 5.0% or lower. A low recall rate means the management class values assumed in the analysis are good and can be used in the implementation of tape mount management. If the recall rate is high, you need to choose a better management class to reduce the rate.

You can use the Management Class Report to help select the best management class for each application. You then rerun the Estimate Report using the new management class values to get a more accurate picture of the savings. For more information on Management Class Reports, see "Determining Management Classes for Tape Mount Management" on page 198.

Assume that after this initial assessment, you decide the benefits significantly outweigh the costs in this sample installation. The next step is to look at how the volume mount analyzer handled the IDCAMS data in the simulation, and what you need to do to prepare for tape mount management.

Using the DASD Buffer Cost Breakdown Table: You can use the three detailed reports that accompany the Savings/Cost Summary report to find out how the volume mount analyzer handled the data sets generated by IDCAMS. For information on all three detailed reports, See *DFSMS/MVS V1R2 Using the Volume Mount Analyzer*.

As an example, we will use the DASD Buffer Cost Breakdown table, one of the detailed reports, to refine the original estimates for our sample installation. Figure 142 shows the DASD Buffer Cost Breakdown table generated with our Estimate Report.

GFTASRT3 -- ESTIMATE REPORT -- 33 DAY SAMPLE											
DASD BUFFER COST BREAKDOWN											
DATA SET CATEGORY	DSN#	ORIGINAL SIZE-MB	ORIGINAL SIZE-TK	DFSMS - PRIMARY DSN#	PRIMARY SIZE-TK	DFHSM - LEVEL 1 DSN#	LEVEL 1 SIZE-TK	DFHSM - LEVEL 2 DSN#	LEVEL 2 SIZE-MB	TAPE - DIRECT DSN#	DIRECT SIZE-MB
TEMPORARY	1	2	45	1	1						
ACTIVE	365	1174	35849	4	1086	0	0	361	63		
ACTV GDG	1714	32436	954915	75	28937	0	0	1639	2514		
BACKUP	10306	164673	4868078	633	147518			9673	12075		
LARGE	0	0	0							0	0
DFHSM SFF	0	0	0							0	0
DFHSM GEN	0	0	0							0	0
MGMTCLAS					7401		0		322		
T O T A L	12386	198285	5858887	713	184943	0	0	11673	14974	0	0

Figure 142. Refining Tape Mount Management Categories Using the Estimate Report

The table shows that data sets created by IDCAMS fall in the five different tape mount management categories (see the DSN# column). However, when implementing tape mount management, you can choose only one management class to manage all of these data sets.

Because most of the data sets in this study are considered to be in the BACKUP category, you can try to force all of these data sets into the BACKUP category. You then run another Estimate Report using the statements in Figure 143:

```
REPORT(ESTIMATE) TAPEDEV(3490E) TAPELEN(2)
PGM(INC(IDCAMS)) CLASSIFY(BACKUP) LARGE(999999999)
```

Figure 143. Producing a Focused Estimate Report

Note that if GFTAVMA assumes the data sets generated by the program are active but most of the data set records are output, you need to code BPGM(INC(pgmname)) to tell GFTAVMA to assume pgmname is a backup program. Then all data sets produced by the program will be considered backup if they were written but never read.

To have GFTAVMA show a worst-case scenario of moving backup data sets every hour, specify the statements in Figure 144.

```
REP(EST) PGM(INC(IDCAMS))
CLASS(ACTV) L0AGE(0) L1AGE(0)
```

Figure 144. Generating a Pessimistic Estimate Report

In this estimate, data sets are migrated immediately. In addition, GFTAVMA assumes a recall in all cases, if a second reference occurs minutes later. Because of the high recall rate, the estimate is considered highly pessimistic.

With any of the new Savings/Cost Summary report generated, you need to look again at the total savings and compare them with the cost. You also need to verify that the recall rate is acceptable (5.0% or lower).

Assume that the estimate report generated by the statements in Figure 143, show a slightly high recall rate. You now need to run a Management Class Report to select an appropriate management class for the IDCAMS data sets.

Determining Management Classes for Tape Mount Management

The Management Class Report is used with the Estimate Report to determine cost and savings. It also helps you determine the management class definitions needed to manage the tape mount management data sets.

Run the Management Class Report for each major application you identified. Because IDCAMS is one of the programs identified in the Top Report, and the Estimate Report showed that IDCAMS caused a high rate of recalls, it is used to run the Management Class Report in this section.

Figure 145 on page 199 shows the statement that generated the Management Class Report in Figure 146 on page 199. IDCAMS is the application name used in our filter, and information on the 3420 device is excluded.

```
REP(MGMTCLAS) PGM(INC(IDCAMS)) UNIT(EXC(3420)) LARGE(999999999)
```

Figure 145. Generating Management Class Reports

Figure 146 shows portions of the Management Class Report produced. The report simulates every combination of L0AGE and L1AGE that will use DASD most cost effectively, and presents all the possible management classes for the application. The management classes are sorted in descending order, starting with the most number of mounts incurred, the most number of recalls from level 1 and level 2 storage, and the most number of tracks used.

```
GFTASRT3 -- REPORT(MGMTCLAS)
```

ORDER	L0AGE	L1AGE	L2-MNTS	RECALLS	TOT TRKS	L0-TRKS	L1-TRKS	L2-MB
1	1	1	592	592	7384	7384	0	639
2	0	1	592	592	6498	3824	2674	746
3	0	0	592	592	3824	3824	0	639
4	1	2	235	592	8911	7384	1527	746
5	0	2	235	592	7645	3824	3821	694
/	/	/	/	/	/	/	/	/
11	1	4	92	592	11022	7384	3638	651
12	0	4	92	592	9513	3824	5689	610
13	2	4	92	235	14292	12181	2111	651
14	3	4	92	188	16814	15850	964	651
15	4	4	92	92	18990	18990	0	510
/	/	/	/	/	/	/	/	/
117	11	14	52	57	40237	38244	1993	295
118	12	14	52	54	41767	40443	1324	295
119	13	14	52	53	43285	42626	659	295
120	14	14	52	52	44792	44792	0	231

1GFTAVMA PROGRAM COMPLETED AT 04:30:56 P.M. ON 09/17/1992 -- RETURN CODE = 0

Figure 146. Determining Management Class Attributes for Tape Mount Management

The best management class may vary by installation. When you choose a management class, consider the trade-offs you need to make between the DASD space required, the number of recalls, and the number of mounts required.

After you select a management class that suits the application and your installation's needs, rerun the Estimate Report to determine the savings based on the new management class values. For example, if you decided L0AGE=2 and L1AGE=4 are the values that provide the best results for IDCAMS, then use the statement in Figure 147 on page 200 to rerun the Estimate Report.

One way to reduce the number of mounts is to make sure you are using the capacity of the tape cartridges. But the report in Figure 148 on page 200 shows that 43.8% of all the tapes selected contain at most 5 MB of data, causing 43.7% of the tape mounts. In addition, 65.5% of the total tape cartridges contain at most 50 MB of data.

The numbers in the Tape Volume Size Ranges report indicate poor tape allocation management and poor tape media use. A large portion of the tape cartridges can be used more efficiently by implementing tape mount management techniques.

Using the Tape Data Set Reports

The GFTASRT3 summary report consists of the Tape Data Set Statistics and the Tape Data Set Size Ranges tables. You can use these tables to determine which types of data sets are causing excessive mounts and need to be handled under tape mount management. Figure 149 shows a sample Tape Data Set Statistics table generated in the same GFTAVMA run as the Tape Volume Size Ranges report.

```

*****
*
*           GFTASRT3 TAPE DATA SET STATISTICS
*
*****
38702   GDS(S) - ( 94.8%) --   2049 GDG GROUP(S)   200 MAX GDS/GDG
2118 NON-GDS(S) - (  5.2%)

-----
40820   TOTAL - (100.0%)   40618 IDRC DS ( 99.5%)
  32.0 MB (AVG PER DATA SET)   36127.8 MOUNTS (STATISTICAL)
1306.0 GB (CUM TOTAL ALL DS)   6170.7 VOLUMES (STATISTICAL)
  31 DAYS (MAX REF AGE)         0.1 DAYS (AVG MAX REF AGE)

-----
|                                     SINGLE-VOLUME DATA SET(S)                                     |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| DATA SET | # OF | % OF | # OF GB | % OF GB | # OF STAT | % STAT | # OF STAT | % STAT |
| FILE TYPE | DSNS | DSNS | # OF GB | % OF GB | MOUNTS | MOUNTS | VOLUMES | VOLUMES |
-----+-----+-----+-----+-----+-----+-----+-----+-----+
| SING-FILE | 9337 | 22.9% | 273.3 | 20.9% | 19960.5 | 55.2% | 3777.5 | 61.2% |
-----+-----+-----+-----+-----+-----+-----+-----+
| MULT-FILE | 28833 | 70.6% | 328.3 | 25.1% | 10230.1 | 28.3% | 1474.4 | 23.9% |
-----+-----+-----+-----+-----+-----+-----+
|                                     MULTI-VOLUME DATA SET(S)                                     |
-----+-----+-----+-----+-----+-----+-----+-----+-----+
| DATA SET | # OF | % OF | # OF GB | % OF GB | # OF STAT | % STAT | # OF STAT | % STAT |
| FILE TYPE | DSNS | DSNS | # OF GB | % OF GB | MOUNTS | MOUNTS | VOLUMES | VOLUMES |
-----+-----+-----+-----+-----+-----+-----+-----+-----+
| SING-FILE | 451 | 1.1% | 352.1 | 27.0% | 4507.1 | 12.5% | 664.3 | 10.8% |
-----+-----+-----+-----+-----+-----+-----+-----+
| MULT-FILE | 2199 | 5.4% | 352.3 | 27.0% | 1430.1 | 4.0% | 254.5 | 4.1% |
-----+-----+-----+-----+-----+-----+-----+-----+
| TOTAL DSN | 40820 | 100.0% | 1306.0 | 100.0% | 36127.8 | 100.0% | 6170.7 | 100.0% |
-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 149. Obtaining Information from the Tape Data Set Statistics Report

The Tape Data Set Statistics report shows that the size of a data set on the cartridge device averages 32.0 MB (see top of report). The Single-Volume Data Set(s) section shows that most of the mounts are caused by single-file data sets on a cartridge (55.2% of the mounts, and multiple files stacked on a single cartridge (28.3% of the mounts). Also, although the installation is trying to fill cartridges by heavily stacking data sets (70.6% of the data sets are single-volume multiple-file data sets), the tapes are not very full, as the Tape Volume Size Ranges report in Figure 148 on page 200 indicates.

Figure 150 on page 202 shows a sample Tape Data Set Size Ranges report.

```

*****
*
* THE FOLLOWING "DATA SET SIZE RANGE TABLE" REPRESENTS THE CURRENT PATTERN
* OF DATA SET SIZES FOR THE PARTICULAR SET OF DATA IN THE SMF SAMPLE
* SELECTED. IT DOES NOT IMPLY THAT ALL OF THESE DATA SETS STILL EXIST.
* SOME OF THEM MAY HAVE ALREADY EXPIRED. THE TABLE DOES SHOW THE SIZE OF
* EACH DATA SET THE LAST TIME THAT DATA SET HAD BEEN WRITTEN (IF KNOWN).
* THE PURPOSE OF THIS TABLE IS TO SHOW THE CURRENT SIZE RANGES OF DATA SETS
* THAT ARE BEING CREATED IN YOUR INSTALLATION ON THE PARTICULAR SET OF DATA
* YOU HAVE FILTERED ON. THESE SIZE RANGES SHOULD CLOSELY MATCH THE SIZE
* RANGES OF THE DATA SETS THAT ACTUALLY DO EXIST ON YOUR SYSTEM. THE MOST
* RECENT SIZE OF THE DATA SET IS ALWAYS USED.
*
*****
VOLUME MOUNT ANALYZER -- COPYRIGHT IBM CORP. 1987, 1992 -- GFTAVMA VERSION 1.1.0
-----+-----
|                                     GFTASRT3 -- TAPE DATA SET SIZE RANGES                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| DSN SIZE | # OF | % TOT | CUM # | % TOT | # OF STAT | % TOT | CUM STAT | % TOT | # OF STAT | % TOT | CUM STAT | % TOT |
|          | DSNS | DSNS  | DSNS  | DSNS  | MOUNTS   | MNTS  | MOUNTS   | MNTS  | VOLUMES  | VOLS  | VOLUMES  | VOLS  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| = 0 MB  | 2548 | 6.2 | 2548 | 6.2 | 1196.5 | 3.3 | 1196.5 | 3.3 | 186.1 | 3.0 | 186.1 | 3.0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 1 MB | 21088 | 51.7 | 23636 | 57.9 | 13483.7 | 37.3 | 14680.3 | 40.6 | 2385.4 | 38.7 | 2571.5 | 41.7 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 5 MB | 4816 | 11.8 | 28452 | 69.7 | 3287.5 | 9.1 | 17967.8 | 49.7 | 611.4 | 9.9 | 3183.0 | 51.6 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 10 MB | 2177 | 5.3 | 30629 | 75.0 | 1991.8 | 5.5 | 19959.6 | 55.2 | 412.6 | 6.7 | 3595.6 | 58.3 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 25 MB | 2837 | 7.0 | 33466 | 82.0 | 3280.6 | 9.1 | 23240.1 | 64.3 | 502.9 | 8.1 | 4098.5 | 66.4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 40 MB | 1883 | 4.6 | 35349 | 86.6 | 1835.5 | 5.1 | 25075.7 | 69.4 | 327.6 | 5.3 | 4426.0 | 71.7 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 50 MB | 689 | 1.7 | 36038 | 88.3 | 640.6 | 1.8 | 25716.2 | 71.2 | 116.6 | 1.9 | 4542.6 | 73.6 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 75 MB | 1376 | 3.4 | 37414 | 91.7 | 1781.2 | 4.9 | 27497.4 | 76.1 | 276.4 | 4.5 | 4819.1 | 78.1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=100 MB | 707 | 1.7 | 38121 | 93.4 | 981.0 | 2.7 | 28478.4 | 78.8 | 148.8 | 2.4 | 4967.8 | 80.5 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=125 MB | 499 | 1.2 | 38620 | 94.6 | 676.0 | 1.9 | 29154.4 | 80.7 | 89.7 | 1.5 | 5057.6 | 82.0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=150 MB | 141 | 0.3 | 38761 | 95.0 | 195.7 | 0.5 | 29350.1 | 81.2 | 36.1 | 0.6 | 5093.6 | 82.5 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=175 MB | 245 | 0.6 | 39006 | 95.6 | 349.5 | 1.0 | 29699.6 | 82.2 | 69.7 | 1.1 | 5163.3 | 83.7 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=200 MB | 335 | 0.8 | 39341 | 96.4 | 353.0 | 1.0 | 30052.6 | 83.2 | 63.7 | 1.0 | 5227.0 | 84.7 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=300 MB | 584 | 1.4 | 39925 | 97.8 | 704.4 | 1.9 | 30757.1 | 85.1 | 136.5 | 2.2 | 5363.5 | 86.9 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=400 MB | 257 | 0.6 | 40182 | 98.4 | 586.0 | 1.6 | 31343.1 | 86.8 | 104.2 | 1.7 | 5467.7 | 88.6 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=500 MB | 48 | 0.1 | 40230 | 98.6 | 192.3 | 0.5 | 31535.4 | 87.3 | 18.4 | 0.3 | 5486.0 | 88.9 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=600 MB | 172 | 0.4 | 40402 | 99.0 | 542.4 | 1.5 | 32077.8 | 88.8 | 126.9 | 2.1 | 5612.9 | 91.0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=700 MB | 21 | 0.1 | 40423 | 99.0 | 466.6 | 1.3 | 32544.4 | 90.1 | 50.0 | 0.8 | 5662.9 | 91.8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=800 MB | 30 | 0.1 | 40453 | 99.1 | 511.0 | 1.4 | 33055.4 | 91.5 | 80.0 | 1.3 | 5742.9 | 93.1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <=900 MB | 58 | 0.1 | 40511 | 99.2 | 805.0 | 2.2 | 33860.4 | 93.7 | 111.2 | 1.8 | 5854.2 | 94.9 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 1 GB | 84 | 0.2 | 40595 | 99.4 | 826.3 | 2.3 | 34686.6 | 96.0 | 137.1 | 2.2 | 5991.3 | 97.1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 2 GB | 216 | 0.5 | 40811 | 100.0 | 1380.1 | 3.8 | 36066.8 | 99.8 | 169.5 | 2.7 | 6160.7 | 99.8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 3 GB | 6 | 0.0 | 40817 | 100.0 | 49.0 | 0.1 | 36115.8 | 100.0 | 6.0 | 0.1 | 6166.7 | 99.9 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 4 GB | 1 | 0.0 | 40818 | 100.0 | 4.0 | 0.0 | 36119.8 | 100.0 | 2.0 | 0.0 | 6168.7 | 100.0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <= 5 GB | 1 | 0.0 | 40819 | 100.0 | 4.0 | 0.0 | 36123.8 | 100.0 | 2.0 | 0.0 | 6170.7 | 100.0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| > 5 GB | 1 | 0.0 | 40820 | 100.0 | 4.0 | 0.0 | 36127.8 | 100.0 | 0.0 | 0.0 | 6170.7 | 100.0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 150. Obtaining Information from the Tape Data Set Size Ranges Report

The Tape Data Set Size Ranges table in our example shows that small data sets cause most of the mounts in the installation. In the <=100 MB row, we see that data sets that are at most 100 MB account for 93.4% of the data sets examined, and they cause 78.8% of the tape mounts. In the <=10 MB row, we see that 75.0% of the data sets analyzed are at most 10 MB, and they account for 55.2% of all tape mounts in the installation. Therefore, small data sets are not allocated efficiently to tape and are causing excessive mounts.

The report also shows that certain data sets are large enough to be placed on larger capacity tapes without being managed through the DASD buffer. The <=600 MB row shows that 99.0% of the data sets fall in the "not large" category (these represent 88.8% of all mounts). Then the 1.0% "large" data sets can benefit from large-capacity tape.

By applying tape mount management techniques, you can improve the management of tape data allocations to make better use of the cartridge devices, reduce the number of mounts required, and improve the performance of the job that called for the tape allocation. For example, most of the data sets examined in this section belong on tape because they are not very active, but they are not using the tape capacity effectively, causing many unnecessary mounts. With a good tape mount management strategy, you can allocate these data sets to tape efficiently, thereby reducing the number of mounts. In addition, "large" data sets can be placed directly on tape, taking advantage of the compaction functions available, and filling the tapes to capacity.

See *MVS/ESA SML: Implementing System-Managed Storage* for procedures on how to develop filter criteria and define SMS classes and groups to handle tape data allocations more efficiently.

Using the Usage Report

The Usage Report shows the breakdown of tape mount activity and tape drive usage by day and hour. You can use the report to determine the affect of tape mount management on the number of tape transports needed. The report also gives clues on the critical batch windows and how tape mount management might affect the bottleneck on waiting for tape drive allocations.

To produce a Usage Report, use the REPORT and USAGE keywords. Figure 151 shows the statement that generated the Usage Report in this section. Data on the 3420s are excluded in this run.

```
REPORT(USAGE) PGM(INC(IDCAMS)) UNIT(EXC(3420))
```

Figure 151. Producing the Usage Report

Figure 152 on page 204 shows portions of the Usage Report generated. It includes both the concurrent allocations and tape mounts tables.

GFTASRT1 -- TAPE DRIVE/MOUNT USAGE REPORT

 * MAXIMUM CONCURRENT TAPE ALLOCATIONS BY HOUR *

DATE:	HOUR:																							<==== DAILY =====>		
MM/DD/YYYY	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	MAXIMUM	MOUNTS
05/27/1991	3	4	11	2	2	2	5	2	4	2	1	0	5	6	1	1	4	2	1	3	12	6	5	2	12	275
05/28/1991	4	3	10	2	4	6	5	6	7	10	5	8	7	3	6	8	7	6	5	11	14	12	10	7	14	1176
05/29/1991	10	16	16	5	5	9	5	4	8	14	7	7	6	6	4	6	7	7	7	12	14	10	14	16	16	1402
05/30/1991	12	14	10	4	3	7	6	4	11	9	11	6	5	4	5	7	6	6	9	8	14	14	13	14	14	1239
05/31/1991	12	14	9	4	6	7	5	5	7	8	6	5	5	5	8	7	6	4	3	10	12	11	8	11	14	1305
06/01/1991	14	14	15	14	16	11	9	12	12	11	12	7	9	10	5	2	3	2	4	6	10	5	9	10	16	1066
06/02/1991	3	0	16	12	8	7	6	5	2	0	0	0	0	1	0	0	1	0	0	0	2	3	8	0	16	252
06/03/1991	3	3	12	4	4	6	6	2	8	9	7	7	7	5	9	7	6	7	7	11	13	12	7	11	13	1076
06/04/1991	5	7	16	14	15	9	8	3	9	6	7	5	6	8	5	7	8	6	6	13	15	10	8	10	16	1339
06/05/1991	11	10	16	15	15	7	7	7	7	6	6	8	7	4	6	7	12	10	8	12	15	14	8	11	16	1322
06/06/1991	14	13	16	6	8	9	5	7	9	13	11	8	8	8	9	9	12	14	9	11	15	14	11	15	16	1416
06/07/1991	12	12	14	10	7	5	4	4	6	8	6	7	4	5	6	10	11	3	6	9	13	13	9	11	14	1272
06/08/1991	10	10	16	14	13	4	7	12	11	11	9	8	10	6	2	0	2	0	10	11	4	5	9	0	16	987
06/09/1991	3	4	16	9	6	8	6	6	3	0	0	0	1	2	2	0	2	0	1	2	3	1	8	0	16	290
06/10/1991	3	5	12	6	4	5	6	4	5	10	8	6	7	3	8	6	6	7	7	10	13	11	7	7	13	1058
06/11/1991	7	16	16	13	7	6	4	5	7	11	6	6	5	4	4	8	8	7	6	10	13	12	8	13	16	1391
06/12/1991	10	14	15	5	7	6	5	6	10	8	9	8	6	7	8	7	5	4	5	14	14	13	10	10	15	1341
06/13/1991	7	14	16	13	12	9	7	4	8	9	10	7	6	8	5	8	6	8	9	12	15	15	9	16	16	1314
06/14/1991	9	8	16	9	6	8	7	4	7	8	5	7	7	14	14	11	15	13	7	9	14	11	11	14	16	1283

Figure 152 (Part 1 of 4). Monitoring Tape Mounts and Concurrent Tape Allocations

GFTASRT1 -- TAPE DRIVE/MOUNT USAGE REPORT

 * MAXIMUM CONCURRENT TAPE ALLOCATIONS BY HOUR *

DATE:	HOUR:																							<=== DAILY ===>		
MM/DD/YYYY	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	MAXIMUM	MOUNTS
06/15/1991	7	9	16	14	16	9	10	13	10	10	8	8	6	3	2	2	1	0	2	4	7	3	5	6	16	897
06/16/1991	8	3	16	13	4	3	15	10	9	7	5	3	3	3	2	1	3	1	1	2	2	3	11	3	16	449
06/17/1991	3	4	11	4	5	7	5	6	10	8	6	6	4	7	6	10	8	10	10	13	14	8	7	9	14	1096
06/18/1991	6	4	15	13	15	16	7	7	9	10	8	6	8	9	9	9	11	9	9	12	15	13	15	13	16	1468
06/19/1991	9	11	14	7	5	7	5	6	9	13	10	6	7	7	7	7	8	8	7	14	15	15	14	11	15	1315
06/20/1991	9	11	15	4	4	7	4	4	8	10	9	8	8	7	8	8	9	11	9	12	15	12	12	13	15	1348
06/21/1991	9	11	12	8	6	7	7	4	11	7	3	5	4	5	5	4	6	6	8	10	13	10	8	11	13	1228
06/22/1991	13	13	14	12	9	4	12	10	10	10	9	10	4	0	0	0	2	0	2	8	8	3	12	2	14	904
06/23/1991	0	0	0	2	2	16	12	2	2	0	0	0	0	0	1	1	1	0	0	0	0	1	8	0	16	201
06/24/1991	1	2	11	2	4	10	4	3	10	8	6	8	7	9	4	5	7	3	6	11	13	12	7	11	13	1079
06/25/1991	7	4	15	3	9	13	14	14	15	14	12	10	10	9	6	11	10	9	8	10	14	13	11	13	15	1533
06/26/1991	16	14	12	7	5	9	4	6	7	7	11	7	8	8	10	10	11	7	7	13	15	11	12	13	16	1265
06/27/1991	11	10	14	10	5	10	5	3	7	5	9	6	6	6	5	6	8	9	10	11	14	15	11	10	15	1323
06/28/1991	8	11	12	11	6	9	8	4	4	4	5	5	4	4	3	5	5	6	5	10	13	13	10	9	13	1219

 * MAXIMUM TAPE ALLOCATIONS: 33 DAYS SUMMARY *

	HOUR:																							<=== 33 DAYS ===>		
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	MAXIMUM	MOUNTS
HOURLY MAXIMUM:	16	16	16	15	16	16	15	14	15	14	12	10	10	14	14	11	15	14	10	14	15	15	15	16	16	36129

Figure 152 (Part 2 of 4). Monitoring Tape Mounts and Concurrent Tape Allocations

GFTASRT1 -- TAPE DRIVE/MOUNT USAGE REPORT

 * TAPE MOUNTS PER HOUR *

DATE:	HOUR:																							<=== DAILY ===>		
MM/DD/YYYY	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	MAXIMUM	MOUNTS
05/27/1991	6	14	28	3	3	6	16	4	11	4	1	0	12	13	1	2	9	2	1	11	68	22	36	2	68	275
05/28/1991	21	10	44	2	11	10	18	21	29	55	51	56	53	29	43	88	60	59	69	88	144	102	69	44	144	1176
05/29/1991	89	103	62	44	19	21	10	18	61	42	36	30	39	39	18	34	51	46	80	142	111	95	85	127	142	1402
05/30/1991	88	40	45	15	12	14	12	20	65	50	32	25	28	27	23	45	39	57	68	52	158	109	104	111	158	1239
05/31/1991	103	49	48	17	23	18	34	22	57	56	34	17	28	17	41	64	43	16	39	107	156	87	103	126	156	1305
06/01/1991	82	121	94	52	62	44	42	49	78	78	64	54	44	54	3	1	4	6	14	38	13	11	41	17	121	1066
06/02/1991	4	0	86	50	24	9	15	3	2	0	0	0	0	1	0	0	3	0	0	0	2	8	45	0	86	252
06/03/1991	4	19	42	15	14	8	27	8	51	47	46	48	27	21	47	36	54	47	78	118	121	84	50	64	121	1076
06/04/1991	35	51	121	56	79	26	26	20	52	42	35	39	21	39	18	59	38	48	69	96	123	91	68	87	123	1339
06/05/1991	57	47	105	55	35	26	20	37	39	38	26	43	34	16	20	56	51	62	70	81	141	99	58	106	141	1322
06/06/1991	106	70	63	29	21	16	23	48	50	54	64	34	25	36	18	46	49	70	66	37	147	125	74	145	147	1416
06/07/1991	93	60	64	39	32	10	10	27	38	51	61	28	14	20	15	50	50	13	68	93	146	115	86	88	146	1272
06/08/1991	69	71	110	61	46	16	21	53	76	76	67	73	52	16	0	0	2	0	48	53	13	10	54	0	110	987
06/09/1991	6	20	86	31	25	17	16	9	4	0	0	0	1	6	4	0	3	0	2	5	8	2	45	0	86	290
06/10/1991	4	19	37	19	10	6	30	11	20	59	36	54	32	30	45	59	49	63	79	101	114	83	50	48	114	1058
06/11/1991	63	103	87	50	28	13	15	23	50	38	42	39	29	31	21	50	56	59	68	93	134	90	86	123	134	1391
06/12/1991	86	81	64	27	19	12	25	20	77	49	47	12	35	50	30	49	41	14	75	108	110	112	66	132	132	1341
06/13/1991	46	72	76	54	31	28	15	18	42	44	59	29	16	22	14	62	34	63	68	41	144	112	82	142	144	1314
06/14/1991	61	58	66	41	18	33	22	18	54	52	35	32	26	30	40	47	51	22	80	73	119	70	79	155	155	1283

Figure 152 (Part 3 of 4). Monitoring Tape Mounts and Concurrent Tape Allocations

After implementing tape mount management, you will be making effective use of high capacity tape cartridges. At this point, you might want to consider servicing your tape needs with the IBM 3495 Tape Library Dataserver. With the Tape Library Dataserver, you can automate tape mount management and eliminate the operating costs of your manual library. For more information on converting to system-managed tape environment, see *MVS/ESA SML: Implementing System-Managed Storage*.

Appendix A. VSAM Data Set Allocation Parameters

This appendix shows the parameters you can specify in JCL, data class, and IDCAMS when allocating VSAM data sets. Default values, if they exist, are provided in areas where you cannot specify a parameter.

For information on allocating VSAM data sets using JCL, see *JCL User's Guide*. For information on allocating VSAM data sets using IDCAMS, see *DFSMS/MVS V1R2 Access Method Services for ICF*.

Parameter Description	JCL	Data Class	IDCAMS
Allocation Units of Space	CYL AVGREC=K AVGREC=M ---- TRK	---- AVGREC=K AVGREC=M ---- ----	CYL KB MB REC TRK
Buffer Space	AMP=BUFSP	----	BUFSP
Control Interval Size	----	CISIZE DATA	CISZ
Data Class Name	DATACLAS	----	DATACLAS
Data Organization	RECORG=ES RECORG=LS RECORG=KS RECORD=RR	RECORG=ES RECORG=LS RECORG=KS RECORG=RR	NIXD LIN IXD NUMD
Data Set Name (Cluster Name)	DSN	DSN	NAME
Expiration Date	EXPDT	EXPDT	TO
Free space % control area % control interval	Default (if no BUFSPC specified) Index component = 512 Data component = 1024, 2048, or 4096	% FREESPACE CA % FREESPACE CI	FSPC FSPC
Imbed sequence set	---- ----	IMBED=Y IMBED=N	IMBD NIMDB
Keys	KEYLEN KEYOFF	KEYLEN KEYOFF	KEYS KEYS
Management class name	MGMTCLAS	----	MGMTCLAS
Model	LIKE= REFDD=	---- ----	MODEL MODEL
Record size	LRECL	LRECL	RECSZ
Replicate index records	---- ----	REPLICATE=Y REPLICATE=N	REPL NREPL
Retention Period	RETPD	RETPD	FOR
Share options Cross region Cross system	Default 1 3	SHAREOPTION XREGION SHAREOPTION XSYSTEM	SHR SHR
Storage class name	STORCLAS	----	STORCLAS
Volume	VOL=SER	----	VOL

Appendix B. DCOLLECT Record Types

This appendix lists the different types of DCOLLECT records and their contents. The VSAM space use (HURBA) information is available if you have DFSMS/MVS installed.

For information on DCOLLECT record formats, see *DFSMS/MVS V1R2 Access Method Services for ICF*.

Record type	Record Contents
Active Data Set (Record Type D)	<ul style="list-style-type: none"> Data set name Data set organization RACF-defined data set indicator System-managed data set indicator Temporary data set indicator Partitioned data set extended (PDSE) indicator Hierarchical file system (HFS) data set indicator Generation data group data set indicator System-reblockable data set indicator Cataloged data set indicator Integrated catalog indicator (if data set is an integrated catalog) Data set VVR indicator Volume serial number Data set record format Data set record length Data set block length Data set size before compression Data set size after compression Data set size validation indicator Amount of space allocated (kilobytes) Amount of space actually used (kilobytes) Number of extents used Secondary allocation Number of unusable bytes in blocks Creation date Expiration date Last reference date Data set serial number Volume sequence number Date and time of last backup Data class name Management class Storage class name Storage group name Error information
Volume Information (Record Type V)	<ul style="list-style-type: none"> Volume serial number VTOC index status Volume usage (Private/Public/Storage) Shared device status Percentage of free space Amount of free space (kilobytes) Amount of allocated space (kilobytes) Total capacity of the volume (kilobytes) Fragmentation index of the volume Largest extent (kilobytes) Number of free extents Number of free DSCBs in the VTOC Number of VIRs Volume device type Volume device number Storage group name Error information

Record type	Record Contents
VSAM Association (Record Type A)	<ul style="list-style-type: none"> Data set component name Data set cluster name High used relative byte address High allocated relative byte address Number of logical records Number of deleted records Number of inserted records Number of updated records Number of retrieved records Bytes of free space in the data set Number of CI splits Number of CA splits Number of EXCPs Relative key position Key length Data set size before compression Data set size after compression
Migration Data Set (Record Type M)	<ul style="list-style-type: none"> Data set name Data set organization Data set record format Data set block length RACF-defined data set indicator System-managed generation data group data set indicator System-reblockable data set indicator Partitioned data set extended (PDSE) indicator Space allocated (kilobytes) Space used (kilobytes) Migration level where the data set currently resides Number of migrations Recall space estimate Changed-since-last-backup indicator Device class of the volume Size of migrated copy (kilobytes) Date and time of the data set migration Creation date Expiration date Date of last backup Date last referenced Data class name Management class name Storage class name Storage group name
Data Set Backup Version (Record Type B)	<ul style="list-style-type: none"> Data set name Data set record format Data set block length RACF-defined data set indicator System-managed generation data group data set indicator System-reblockable data set indicator Partitioned data set extended (PDSE) indicator Space allocated (in kilobytes) Space used (kilobytes) Recovery space estimate Device class (tape or DASD) of backup volume Backup version size in kilobytes Date and time of latest backup version Data class name Management class name Storage class name

Record type	Record Contents
DASD Capacity Planning (Record Type C)	DASD volume serial number Type of volume (primary or migration level 1) Total capacity of the volume (in tracks) Date DASD capacity planning data was collected Specified target occupancy of primary volume (low threshold) Specified trigger occupancy of primary or migration level 1 volume, (high threshold for primary, threshold for migration level 1) Percentage occupancy of the volume before HSM space management processing Percentage occupancy of the volume after HSM space management processing Percentage of the volume's data not migrated but eligible to migrate Number of times the HSM interval migration function was run against the volume Number of times the target occupancy was met for the volume during interval migration
Tape Capacity Planning (Record Type T)	Type of HSM tape (backup, dump, migration) Number of full tape volumes Number of partially filled tape volumes Number of empty tape volumes

Abbreviations

The following acronyms and abbreviations are defined as they are used in the MVS/ESA Storage Management Library. If you do not find the abbreviation you are looking for, see *Dictionary of Computing*, New York: McGraw-Hill, 1994.

This list can include acronyms and abbreviations from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1).

ABARS. Aggregate backup and recovery support.
ACDS. Active control data set.
ACS. Automatic class selection.
BCDS. Backup control data set.
BCS. Basic catalog structure.
CHPID. Channel path identifier.
CLIST. Command list.
COMMDS. Communications data set.
CTC. Channel to channel.
CVOL. Control volume.
DADSM. Direct access device space management.
DASD. Direct access storage device.
DCB. Data control block.
DEDB. Data entry database.
DFDSS. Data Facility Data Set Services.
DFHSM. Data Facility Hierarchical Storage Manager.
DFSORT. Data Facility Sort.
DL/1. Data Language/1
EREP. Environmental Error Record Editing and Printing program.
GAC. Global access checking.
GDG. Generation data group.
GB. Gigabyte; 10⁹ bytes.
GRS. Global resource serialization.
HCD. Hardware Configuration Definition.

HLQ. High level qualifier.
IDRC. Improved data recording capability.
I/O. Input/output.
IMS. Information Management System.
IPL. Initial program load.
ISMF. Interactive Storage Management Facility.
ISPF. Interactive System Productivity Facility.
JCL. Job control language.
JES. Job entry subsystem.
KB. Kilobyte; 1024 bytes.
LAN. Local area network.
LLQ. Lowest level qualifier.
MB. Megabyte; 10⁶ bytes.
MCDS. Migration control data set.
ML1. Migration level 1 volume.
ML2. Migration level 2 volume.
MSR. Millisecond response.
MVS. Multiple Virtual Storage.
MVSCP. MVS configuration program.
NVR. Non-VSAM volume record.
NVS. Nonvolatile storage.
OAM. Object access method.
OS. Operating system.
OSAM. Overflow sequential access method.
OSMC. OAM storage management component.
PDSE. Partitioned data set extended.
PTF. Program temporary fix.
RMF. Resource Measurement Facility.
SAF. System Authorization Facility.
SCDS. Source control data set.
SDSP. Small data set packing.
SLR. Service Level Reporter.
SMF. System Management Facilities.
SML. Storage Management Library.
SMP/E. System Modification Program Extended.
SMS. Storage Management Subsystem.
SSL. Storage Subsystem Library.
TB. Terabyte (10¹² bytes).

TCDB.	Tape configuration database.	VTOC.	Volume table of contents.
TSO/E.	Time Sharing Option/Extended.	VVDS.	VSAM volume data set.
VIO.	Virtual input/output.	VVR.	VSAM volume record.
VSAM.	Virtual storage access method.	WORM.	Write-once-read-many.

Glossary

The following terms are defined as they are used in the MVS/ESA Storage Management Library. If you do not find the term you are looking for, see *Dictionary of Computing*, New York: McGraw-Hill, 1994.

This glossary may include terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1).

A

active control data set (ACDS). A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See also *source control data set* and *communications data set*.

active data. (1) Data that can be accessed without any special action by the user, such as data on primary storage or migrated data. Active data also can be stored on tape volumes. (2) For tape mount management, application data that is frequently referenced, small in size, and managed better on DASD than on tape. Contrast with *inactive data*.

aggregate backup. The process of copying an aggregate group and recovery instructions so that a collection of data sets can be recovered later as a group.

aggregate group. A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

automated tape library. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. See also *tape library*. Contrast with *manual tape library*.

automatic backup. (1) In DFSMSHsm, the process of automatically copying data sets from primary storage volumes or migration volumes to backup volumes. (2) In OAM, the process of automatically copying objects from DASD, optical, or tape volumes contained in an object storage group, to backup volumes contained in an object backup storage group.

automatic class selection (ACS) routine. A sequence of instructions for having the system assign data class, storage class, management class, and storage group for data sets, and storage class, management class, and storage groups for objects.

automatic data set protection. In MVS, a user attribute that causes all permanent data sets created by the user to be automatically defined to RACF with a discrete RACF profile.

automatic dump. In DFSMSHsm, the process of using DFSMSdss automatically to do a full-volume dump of all allocated space on a primary storage volume to designated tape dump volumes.

automatic primary space management. In DFSMSHsm, the process of deleting expired data sets, deleting temporary data sets, releasing unused space, and migrating data sets from primary storage volumes automatically.

automatic secondary space management. In DFSMSHsm, the process of automatically deleting expired migrated data sets, deleting expired records from the migration control data sets, and migrating eligible data sets from migration level 1 volumes to migration level 2 volumes.

automatic volume space management. In DFSMSHsm, the process that includes automatic primary space management and interval migration.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration is used to manage.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. See also *integrated catalog facility catalog*.

C

cache fast write. A storage control capability in which the data is written directly to cache without using non-volatile storage. Cache fast write is useful for temporary data or data that is readily recreated, such as the sort work files created by DFSORT. Contrast with *DASD fast write*.

capacity planning. The process of forecasting and calculating the appropriate amount of physical computing resources required to accommodate an expected workload.

Cartridge System Tape. The base tape cartridge media used with 3480 or 3490 Magnetic Tape Subsystems. Contrast with *Enhanced Capacity Cartridge System Tape*.

class transition. An event that brings about change to an object's service-level criteria that causes ACS routines to be invoked by OAM to assign a new storage class or management class to the object.

compress. (1) To reduce the amount of storage required for a given data set by having the system replace identical words or phrases with a shorter token associated with the word or phrase. (2) To reclaim the unused and unavailable space in a partitioned data set that results from deleting or modifying members by moving all unused space to the end of the data set.

communications data set (COMMDS). The primary means of communication among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS. See also *active control data set* and *source control data set*.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with normal application program processing.

connectivity. (1) The considerations regarding how storage controls are joined to DASD and processors to achieve adequate data paths (and alternative data paths) to meet data availability needs. (2) In a DFSMS environment, the system status of volumes and storage groups.

convert in place. See *in-place conversion*.

D

DASD fast write. A storage control capability in which data is written concurrently to cache and nonvolatile storage and automatically scheduled for destaging to DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. Use of DASD fast write for system-managed data sets is controlled by storage class attributes to improve performance. See also *dynamic cache management*. Contrast with *cache fast write*.

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume*, *primary storage*, *migration level 1*, *migration level 2*.

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

Data Facility Sort (DFSORT). An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

default device geometry. Part of the SMS base configuration, it identifies the number of bytes per track and the number of tracks per cylinder for converting space requests made in tracks or cylinders into bytes, when no unit name has been specified.

default management class. Part of the SMS base configuration, it identifies the management class that should be used for system-managed data sets that do not have a management class assigned.

default unit. Part of the SMS configuration base, it identifies an esoteric (such as SYSDA) or generic (such as 3390) device name. If a user fails to specify the UNIT parameter on JCL, SMS applies the default unit to all data sets having a disposition of either MOD or NEW that are *not* system-managed.

Device Support Facilities (ICKDSF). A program used for initialization of DASD volumes and track recovery.

DFSMS environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, this function is provided by MVS/ESA SP and DFSMS/MVS, DFSORT, and RACF. See also *system-managed storage*.

DFSMS/MVS. An IBM licensed program that together with MVS/ESA SP compose the base MVS/ESA oper-

ating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMSHsm, and DFSMSrmm.

DFSMSdfp. A DFSMS/MVS functional component that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component used to copy, move, dump, and restore data sets and volumes.

DFSMSHsm. A DFSMS/MVS functional component used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMSHsm-managed volume. (1) A primary storage volume, which is defined to DFSMSHsm but which does not belong to a storage group. (2) A volume in a storage group, which is using DFSMSHsm automatic dump, migration, or backup services. Contrast with *system-managed volume* and *DFSMSrmm-managed volume*.

DFSMSHsm-owned volume. A storage volume on which DFSMSHsm stores backup versions, dump copies, or migrated data sets.

DFSMSrmm. A DFSMS/MVS functional component that manages removable media.

DFSMSrmm-managed volume. A tape or optical volume that is defined to DFSMSrmm. Contrast with *system-managed volume* and *DFSMSHsm-managed volume*.

dictionary. A table that associates words, phrases, or data patterns to shorter tokens. The tokens are used to replace the associated words, phrases, or data patterns when a data set is compressed.

disaster recovery. A procedure for copying, storing, and recovering data essential to an installation's business in a secure location, and for recovering that data in the event of a catastrophic failure at the installation. Contrast with *vital records*.

drive definition. A set of attributes used to define an optical disk drive as a member of a real optical library or pseudo optical library.

dual copy. A high availability function made possible by nonvolatile storage in some models of the 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dummy storage group. A type of storage group that contains the serial numbers of volumes no longer connected to a system. Dummy storage groups allow existing JCL to function without having to be changed. See also *storage group*.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMSHsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

dynamic cache management. A function that automatically determines which data sets will be cached based on the 3990 subsystem load, the characteristics of the data set, and the performance requirements defined by the storage administrator.

E

Enhanced Capacity Cartridge System Tape. Cartridge system tape with increased capacity that can only be used with 3490E tape subsystems. Contrast with *Cartridge System Tape*.

esoteric unit name. A name used to define a group of devices having similar hardware characteristics, such as TAPE or SYSDA. Contrast with *generic unit name*.

expiration. The process by which data sets or objects are identified for deletion because their expiration date or retention period has passed. On DASD, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse.

extended format. The format of a data set that has a data set name type (DSNTYPE) of EXTENDED, for example, extended sequential and extended key-sequenced data sets. Data sets in extended format can be striped or compressed.

extended sequential data set. A sequential data set that is structured logically the same as a physical sequential data set but that is stored in a different physical format. Extended sequential data sets consist of one or more stripes. See also *stripe* and *striping*.

F

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully- or partially-qualified data set names or of certain data set characteristics.

G

generic unit name. A name assigned to a class of devices with the same geometry (such as 3390). Contrast with *esoteric unit name*.

global access checking. In RACF, the ability to allow an installation to establish in-storage table of default values for authorization levels for selected resources. RACF refers to this table prior to performing normal RACHECK processing, and grants the request without performing a RACHECK if the requested access authority does not exceed the global value. Global access checking can grant the user access to the resource, but it cannot deny access.

global resource serialization (GRS). A component of MVS/ESA SP used for sharing system resources and for converting DASD reserve volumes to data set enqueues.

H

Hardware Configuration Definition (HCD). An interactive interface in MVS/ESA SP that enables an installation to define hardware configurations from a single point of control.

hierarchical file system (HFS) data set. A data set that contains a POSIX-compliant hierarchical file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using the OpenEdition MVS facilities.

I

implementation by milestone. A conversion approach that allows for a staged conversion of your installation's data to system-managed storage on DASD, tape, or optical devices.

improved data recording capability (IDRC). A recording mode that can increase the effective cartridge data capacity and the effective data rate when enabled and invoked. IDRC is always enabled on the 3490E tape subsystem.

inactive data. (1) A copy of active data, such as vital records or a backup copy of a data set. Inactive data is never changed, but can be deleted or superseded by another copy. (2) In tape mount management, data that is written once and never used again. The majority of this data is point-in-time backups. (3) Objects infrequently accessed by users and eligible to be moved to the optical library or shelf. Contrast with *active data*.

indexed VTOC. A volume table of contents with an index that contains a list of data set names and free space information, which allows data sets to be located more efficiently.

in-place conversion. The process of bringing a volume and the data sets it contains under the control of SMS without data movement, using DFSMSdss.

integrated catalog facility catalog. A catalog that is composed of a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also *basic catalog structure* and *VSAM volume data set*.

Interactive Storage Management Facility (ISMF).

The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

interval migration. In DFSMSHsm, migration that occurs automatically when the high threshold of occupancy is exceeded on a primary storage volume. Data sets are moved from the volume, largest data set first, until the low threshold of occupancy is reached.

M

management class. A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space; to control the retention, migration, and back up of data sets; to control the retention and back up of aggregate groups, and to control the retention, back up, and class transition of objects.

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which may be mounted on those drives. See also *tape library*. Contrast with *automated tape library*.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. The data must be recalled to be used again.

migration level 1. DFSMSHsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 2*.

migration level 2. DFSMSHsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 1*.

MVS configuration program (MVSCP). A single-step, batch program that is used to define input/output configurations to MVS.

MVS/ESA. An MVS operating system environment that supports ESA/370.

MVS/ESA SP. An IBM licensed program used to control the MVS operating system. MVS/ESA SP together with DFSMS/MVS compose the base MVS/ESA operating environment.

N

nondisruptive installation. A capability that allows users to access data in an existing storage subsystem while installing additional devices in the subsystem.

nonvolatile storage (NVS). Additional random access electronic storage with a backup battery power source, available with some storage controls, used to retain data during a power failure. Nonvolatile storage, accessible from all storage directors, stores data during DASD fast write and dual copy operations.

O

OAM-managed volumes. Optical or tape volumes controlled by the object access method (OAM).

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

object backup storage group. A type of storage group that contains optical or tape volumes used for backup copies of objects. See also *storage group*.

object directory tables. A collection of DB2 tables that contain information about the objects that have been stored in an object storage group.

object storage group. A type of storage group that contains optical or DASD volumes used for objects. See also *storage group*.

object storage hierarchy. A three-level hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical or tape volumes that reside in a library, and on optical or tape volumes that reside on a shelf. See also *storage hierarchy*.

object storage tables. A collection of DB2 tables that contain objects.

OpenEdition MVS. A feature of MVS/ESA SP that allows applications that conform to POSIX standards to operate or access data on an MVS/ESA system.

optical disk drive. The mechanism used to seek, read, and write data on an optical disk. An optical disk drive can be operator-accessible, such as the 3995 Optical Library Dataserver, or stand-alone, such as the 9346 or 9347 optical disk drives.

optical library. A storage device that houses optical drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

P

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

performance. (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) In a DFSMS environment, a measurement of effective data processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with *temporary data set*.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

primary space allocation. Amount of space requested by a user for a data set when it is created. Contrast with *secondary space allocation*.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1* and *migration level 2*.

pseudo optical library. A set of shelf-resident optical volumes associated with either a stand-alone or an

operator-accessible optical disk drive; see also *real optical library*.

R

real optical library. Physical storage device that houses optical disk drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives. Contrast with *pseudo optical library*.

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

removable media library. The volumes that are available for immediate use, and the shelves where they could reside.

Resource Access Control Facility (RACF). An IBM-licensed program that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

Resource Measurement Facility (RMF). An IBM licensed program that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

S

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full. Contrast with *primary space allocation*.

service-level agreement. (1) An agreement between the storage administration group and a user group defining what service-levels the former will provide to ensure that users receive the space, availability, performance, and security they need. (2) An agreement between the storage administration group and operations defining what service-level operations will provide to ensure that storage management jobs required by the storage administration group are completed.

Service Level Reporter (SLR). An IBM licensed program that provides the user with a coordinated set of tools and techniques and consistent information to help manage the data processing installation. For example, SLR extracts information from SMF, IMS, and CICS logs, formats selected information into tabular or graphic

reports, and gives assistance in maintaining database tables.

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf location. A single space on a shelf where you store removable media.

small data set packing (SDSP). In DFSMSHsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

small-data-set-packing data set. In DFSMSHsm, a VSAM key-sequenced data set allocated on a migration level 1 volume and containing small data sets that have been migrated.

SMS complex. A collection of systems or system groups that share a common configuration. All systems in an SMS complex share a common active control data set (ACDS) and a communications data set (COMMDS). The systems or system groups that share the configuration are defined in the SMS base configuration.

SMS configuration. A configuration base, Storage Management Subsystem class, group, library, and drive definitions, and ACS routines that the Storage Management Subsystem uses to manage storage. See also *base configuration* and *source control data set*.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF. See also *active control data set* and *communications data set*.

storage administration group. A centralized group within the data processing center that is responsible for managing the storage resources within an installation.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A named list of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage control. The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

storage director. In a 3990 storage control, a logical entity consisting of one or more physical storage paths

in the same storage cluster. In a 3880, a storage director is equivalent to a storage path.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes treated as a single object storage hierarchy. See also *VIO storage group*, *pool storage group*, *tape storage group*, *object storage group*, *object backup storage group*, and *dummy storage group*.

storage group category. A grouping of specific storage groups which contain the same type of data. This concept is analogous to storage pools in a non-system-managed environment.

storage hierarchy. An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See also *primary storage*, *migration level 1*, *migration level 2*, and *object storage hierarchy*.

storage location. A location physically separate from the removable media library where volumes are stored for disaster recovery, backup, and vital records management.

storage management. The task of using storage, such as DASD, tape, or optical devices, to retain and deliver data to applications.

Storage Management Subsystem (SMS). A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage subsystem. A storage control and its attached storage devices. See also *tape subsystem*.

stripe. The portion of a striped data set (for example, an extended sequential data set) that resides on one disk. The records in that portion are not necessarily logically consecutive. The system distributes records among the stripes such that the disks can be read or written simultaneously to gain better performance. See also *extended sequential data set*.

striping. A software implementation of a disk array that distributes data sets across multiple volumes to improve performance.

sysplex. A systems complex that consists of multiple systems coupled together by hardware elements and system services. The systems must be defined by the COUPLExx members of SYS1.PARMLIB as being a sysplex.

system data. The data sets required by MVS or its subsystems for initialization and control.

system group. All systems that are part of the same sysplex and are running the Storage Management Subsystem with the same configuration, minus any systems in the sysplex that are explicitly defined in the SMS configuration.

system-managed data set. A data set that has been assigned a storage class.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, space, and security to applications. See also *DFSMS environment*.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library*.

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with *DFSMSshsm-managed volume* and *DFSMSrmm-managed volume*.

system management facilities (SMF). A component of MVS/ESA SP that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tape configuration database. One or more volume catalogs used to maintain records of system-managed tape libraries and tape volumes.

tape library. A set of equipment and facilities that support the installation's tape environment. This may include tape storage racks, a set of tape drives, and a set of related tape volumes which can be mounted on those drives. See also *system-managed tape library*.

tape librarian. The person who manages the tape library.

Tape Library Dataserver. A hardware device that maintains the tape inventory associated with a set of tape drives. An automated tape library dataserver also manages the mounting, dismounting, and storage of tapes.

tape mount management. The methodology used to optimize tape subsystem operation and use, consisting of hardware and software facilities used to manage tape data efficiently.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape subsystem. A magnetic tape subsystem consisting of controller and devices, which allows users to store data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. Storage space on tape, identified by a volume label, which contains data sets or objects and available free space. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

threshold. A storage group attribute that controls the space usage on DASD volumes. The *low threshold* is used during primary space management to determine the amount of free space to provide on the volume. The *high threshold* is used to determine candidate volumes for new data set allocations, and to determine if data sets must be migrated off of a volume during interval migration.

U

unmovable data set. A DASD data set required by an application, which cannot be moved. Unmovable data sets need to be identified to the system so that they are not relocated. They can be identified by allocating the data set in absolute tracks or by allocating a data set with the data set organization that includes the unmovable attribute. For example, data sets allocated as PSU, POU, DAU, or ABSTR are considered unmovable.

use attribute. (1) The attribute assigned to a DASD volume that controls when the volume can be used to allocate new data sets; use attributes are *public*,

private, and *storage*. (2) For system-managed tape volumes, use attributes are *scratch* and *private*.

user group. A group of users in an installation who represent a single department or function within the organization.

user group representative. A person within a user group who is responsible for representing the user group's interests in negotiations with the storage administration group.

V

validate. To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

virtual input/output (VIO) storage group. A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. See also *storage group*.

vital records. A data set or volume maintained for the purpose of meeting an externally imposed retention requirement, such as a legal requirement. Contrast with *disaster recovery*.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes for disaster recovery and vital records purposes.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, and *tape volume*.

volume status. In the Storage Management Subsystem, indicates whether the volume is fully available for system management:

- "Initial" indicates that the volume is not ready for system management because it contains data sets that are ineligible for system management.
- "Converted" indicates that all of the data sets on a volume have an associated storage class and are cataloged in an integrated catalog facility catalog.
- "Non-system-managed" indicates that the volume does not contain any system-managed data sets and has not been initialized as system-managed.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also *basic catalog structure* and *integrated catalog facility catalog*.

Index

Numerics

- 3390 Model 9 19
- 3990 Model 3 Storage Control 7
- 3990 Model 6 Storage Control 7, 19
- 3995 Optical Library Dataserver 19
- 9121 processor 20

A

- ABARS (aggregate backup and recovery support)
 - ABACKUP command 122
 - ARECOVER command 136, 140
 - description 117
 - managing output files 126—127
 - resolving naming conflicts 143
 - using the ISMF Aggregate Group Application 124, 141
- abbreviations 215
- absolute track data sets, description of 23
- access authority 148
- accounting considerations
 - See cost accounting
- ACDS (active control data set)
 - description 13
 - protecting 150
- acronyms 215
- ACS (automatic class selection) routine
 - allocating new PDSEs 52
 - description 11
 - filtering on data names 27
 - selecting data sets to allocate as PDSEs 49—51
- aggregate group
 - backing up using the ABACKUP command 123
 - backing up using the ISMF Aggregate Group Application 124
 - compatibility between DFHSM and DFSMSHsm 19
 - defining with the ISMF Aggregate Group Application 118
 - overview 11
 - verifying backup contents 123
- aliases, multilevel 31
- allocation
 - data class, using 6
 - data with lower performance requirements 91
 - extended sequential data sets 88
 - generation data sets 54
 - in average blocks 45—46
 - multivolume data sets 42
 - PDSEs 52
 - standardizing with data class 39
 - temporary data sets 91

- archives
 - archiving data with DFSMSDss 114
 - archiving object data with IAFC 115
 - definition 113
 - retrieving archived data 135
 - using DFSMSHsm 115
- attributes
 - data class 39, 59
 - management class 60, 97
 - storage class 83, 97

B

- backup
 - automating 101, 108
 - definition 93
 - frequency of 98
 - incremental 100, 102
 - issuing command 107
 - number of versions 98
 - planning for general backups 93
 - preparing for aggregate backups 117
 - preparing for disaster recovery backups 95
- backup data, deleting obsolete backups 67
- backup-while-open 103
- base configuration 13
- batch data 22
- billing
 - See cost accounting
- block allocation
 - average 45—46
 - evaluating block sizes using ISMF 47
 - increasing in block size 47
 - sample JCL 45
 - using system-determined block sizes 46

C

- caching, dynamic data set 85
- capacity planning 171
- catalog
 - allocation example 33
 - backing up 109, 112
 - eligibility for system management 30
 - performance and reliability of 32
 - protecting 156
 - recovering 133—134
 - sharing
 - between MVS/ESA and MVS/XA 14
 - in JES3 installation 14
- cataloging
 - generation data sets 55

- cataloging (*continued*)
 - objects 23
 - permanent data sets 23
 - CICS database
 - backing up user databases 97
 - backing up using backup-while-open 103
 - backing up using dual copy 111
 - improving I/O performance 85
 - migrating 75
 - coexistence
 - between MVS/ESA and MVS/XA
 - catalog sharing 14
 - DASD sharing 16
 - data set
 - among multisite systems 14
 - multisystem environment 17
 - COMMDS (communications data set)
 - description 13
 - protecting 150
 - compaction, data 79
 - compressing data sets 80
 - concurrent copy 97, 104—107
 - control data set
 - DFSMSHsm
 - backing up 109, 113
 - backing up for disaster recovery 95
 - protecting 151
 - recovering 134
 - DFSMSRmm
 - backing up 110, 113
 - backing up for disaster recovery 96
 - protecting 151
 - recovering 135
 - Storage Management Subsystem
 - ACDS (active control data set) 13
 - backing up 110, 113
 - COMMDS (communications data set) 13
 - overview 13
 - protecting 150
 - recovering 135
 - SCDS (source control data set) 13
 - conventions, data naming 24—30
 - cost accounting
 - description 183
 - using DCOLLECT 183
 - using DFSMSRmm 183
 - critical data
 - backing up 108
 - recovering 133
- D**
- DASD (direct access storage device)
 - erase-on-scratch 147
 - sharing DASD 16
 - volume
 - selection 9
 - data class
 - attributes 39
 - defining 41
 - overriding 40
 - overview 6
 - PDSE filtering 49, 50
 - planning 40
 - sample data class definition panel 46
 - sample data class values 40
 - SETOAM command 39
 - space management attributes 59
 - VSAM parameters 209
 - data compaction
 - using DFSMSHsm 79, 98
 - using IDRC (improved data recording capability) 79, 98
 - data naming conventions
 - database data 26
 - example of filtering on data names 27
 - generation data sets 26
 - highest-level qualifier 24
 - lowest-level qualifier 25
 - minimizing the effect of 29
 - object data 26
 - other qualifier levels 26
 - rationale for 24
 - data set
 - ACDS (active control data set) 13
 - allocation
 - data class 6
 - blocking 80
 - coexistence
 - among multisite systems 14
 - multisystem environment 17
 - shared DASD 16
 - COMMDS (communications data set) 13
 - compressing 56, 80
 - consolidating extents 80
 - deallocating at close time 71
 - dynamic caching 85
 - extended sequential 87
 - large 22
 - migrating 75—76
 - multivolume 22
 - policies 21, 38
 - protecting 156
 - SCDS (source control data set) 13
 - serialization 73
 - sharing 14
 - transferring between systems 14
 - VSAM, allocation parameters 209
 - with DDM attributes 102
 - data types
 - absolute track data sets 23
 - eligible for system management 21—23
 - ineligible for system management 23

- data types (*continued*)
 - unmovable data sets 23
- database data
 - backing up 103, 104, 111
 - description 22
 - improving I/O performance 85
 - migrating 75
 - protecting 157
 - recovering 132, 135
- DATACLAS keyword 34
- DB2 database
 - backing up image copies 100
 - backing up objects in table spaces 95
 - backing up table spaces 100
 - backing up user databases 97
 - backing up using dual copy 111
 - backing up with concurrent copy 104
 - deleting active logs 64
 - deleting production databases 64
 - expiring table spaces and indexes 63
 - improving I/O performance 85
 - migrating 75
 - naming convention sample 26
 - recovering considerations 132
 - recovering optical configuration databases 135
- DCOLLECT
 - capacity planning 171
 - capacity planning data collection
 - command 161
 - examples of using DCOLLECT 162
 - cost accounting 183
 - examples of using DCOLLECT 161
 - record types 211
- deletion
 - automating 63
 - issuing command 66—68
- DFDSS
 - coexistence between releases 19
- DFHSM
 - coexistence between releases 19
- DFSMS environment
 - benefits 2
 - coexistence between MVS/ESA and MVS/XA 14
 - managing data 4
 - product requirements 1
- DFSMS/MVS
 - description 1
 - DFDSS coexistence 19
 - DFHSM coexistence 19
 - sharing
 - data sets 14
- DFSMSsdfp
 - backup-while-open function 103
 - description 1
- DFSMSsdss
 - archiving data 114
- DFSMSsdss (*continued*)
 - coexistence with DFDSS 19
 - COMPRESS command 49
 - COPY command 51, 103
 - description 2
 - DUMP command 67, 102, 103, 107, 114
 - protecting DFSMSsdss functions 152
 - reblocking for PDSs 47
 - reblocking PDSs 80
 - RESTORE command 107
 - retrieving archives 135
 - retrieving vital records 135
 - storing vital records 116
 - TOLERATE(ENQFAILURE) keyword 103
- DFSMSshsm
 - aggregate backup and recovery support (ABARS) 136
 - archiving data 115
 - automatic migration and recall functions 72
 - BACKDS command 107
 - backing up aggregate groups with ABARS 117
 - backing up data
 - blocking and reblocking data sets 80
 - coexistence with DFHSM 19
 - compacting data 79, 98
 - compressing data sets 80
 - consolidating extents 80
 - control data sets
 - backing up 104, 106, 109
 - backing up for disaster recovery 95
 - protecting 151
 - recovering 134
 - data set recovery 130
 - description 2
 - disaster backup and recovery 117
 - HBACKDS command 107, 108
 - HMIGRATE command 78
 - HRECOVER command 130
 - incremental backup 100
 - journaling 109
 - MAXEXTENTS function 80
 - migration 71
 - packing small data sets 81
 - RECOVER command 130
 - recovering aggregate groups with ABARS 136
 - SETSYS command 79
 - spill processing 98
 - threshold processing 72
- DFSMSsrm
 - ABARS output file managing 121, 127
 - benefits of 95
 - control data sets
 - backing up 110, 113
 - backing up for disaster recovery 96
 - protecting 151
 - recovering 135

DFSMSrmm (*continued*)
 description 2
 expiration and retention management 64
 extending tape data and volume protection 157
 inventory 121
 vital record managing 116
 vital record specification 116, 121

DFSORT (Data Facility Sort)
 allocating output and temporary work data sets 91
 description 2
 recovering DFSMSHsm control data sets 134
 SORTWK data sets 88
 using ICETOOL for reporting 163, 165

disaster recovery
 aggregate group
 overview 11
 recovery plan 129
 using ABARS 136

Distributed FileManager
 backing up data with DDM attributes 102

Distributed FileManager/MVS
 data sets created by 12

DSCB (data set control block)
 compatibility between MVS/XA and MVS/ESA 16

DUMP parameters, for archiving with DFSMSdss 114

dynamic data set caching 85

E

EXCP data sets 23

EXPDT, RETPD parameters 37

expiration criteria for different data 64

expired data
 deleting 65
 description 65

extended sequential data set 87
 allocating 88
 reblocking using DFSMSdss 80

F

facility resource class, RACF 153

frequency of backup 98

full volume dumps 102

G

GDGs (generation data groups), migrating 76

GDSs (generation data sets)
 allocating on system-managed volumes 54
 cataloging 55
 JCL example using DATACLAS to allocate 55

generic profiles, RACF 147

global access checking, RACF 148

glossary 217

GUARANTEED BACKUP FREQUENCY attribute 100

GUARANTEED SYNCHRONOUS WRITE attribute 97

H

hardware considerations 19

HFS (hierarchical file system) data sets
 backing up 102
 migrating 76

I

IAFC (Item Access Facility)
 archiving 115
 recovering data 135

ICETOOL 163, 165

IDCAMS
 ALLOCATE command 52
 DELETE command 67
 VSAM parameters 209

improved data recording capability 98

IMS database
 backing up using dual copy 111
 deleting online log data sets 64
 improving I/O performance 85
 migrating 75

incremental backup 100, 102

inline backup 108

integrated catalog facility catalog
See catalog

interactive data sets 21

interval migration 72, 73

ISMF (Interactive Storage Management Facility)
 Aggregate Group Application 118, 124
 data class list panel example 54
 data set list panel example 53
 displaying PDSE information 53
 using for backup and recovery 107

J

JCL (job control language)
 allocation parameters 37
 FREE=CLOSE parameter 71
 keywords 34
 RLSE parameter 70
 simplifying data set allocation 34
 UNIT parameter 41—42
 using RETPD, EXPDT 37
 using STORCLAS, MGMTCLAS, DATACLAS 34
 VSAM parameters 209

M

management class
 attributes
 availability 97
 space management 60

management class (*continued*)
 defining 61—62
 defining for objects 61
 deleting 62
 description 59
 determining for tape mount management 198
 object class transition 77
 overview 8
 planning 61
 MAXEXTENTS function of DFSMSHsm 80
 MGMTCLAS keyword 34
 migration of data
 automating 74
 criteria for different data 75
 data set serialization 73
 description 71
 interval migration 72, 73
 issuing command 78
 selecting data for 76
 space management cycles 73
 threshold processing 72
 monitoring space use
 by data 159
 by volume 161
 multilevel aliases 31
 multisite systems 14
 multisystem environment
 catalog sharing 14
 DASD sharing
 between MVS/ESA and MVS/XA 16
 description 17
 system group 18
 multivolume data sets
 description 22
 nonspecific allocation 42
 specific allocation 44
 MVS/ESA
 coexistence with MVS/XA 14
 products for the DFSMS environment 1
 sharing
 data sets 14
 sharing DASD 17
 VTOC compatibility with MVS/XA 16
 MVS/XA
 coexistence with MVS/ESA 14
 sharing
 data sets 14
 VTOC compatibility with MVS/ESA 16

N

naming conventions
 See data naming conventions
 NetView
 transferring data between systems 14

O

OAM (object access method) 61, 101, 178
 object
 data class 7
 storage class 5, 8
 object data
 archiving 115
 backing up 98, 101
 class transition 72, 77
 deletion of 67
 description 22
 device type 42
 expiration and retention of 64, 65
 monitoring the performance of 178
 protecting 145, 156
 recovering 132
 retrieving archived object data 135
 obsolete and unnecessary Data
 deleting 67
 description 67
 OS/VSAM password protection
 elimination of in system-managed environment 145

P

password protection
 access to non-system-managed data 145
 access to system-managed data 145
 PDS (partitioned data set)
 compared with PDSEs 47
 compression 49
 converting to PDSE 49, 51—52
 definition 47
 inappropriate PDSE conversions 49
 protecting 148
 reblocking using DFSMSdss 80
 reblocking with DFSMSdss 47
 PDSE (partitioned data set extended)
 advantages over PDSs 47
 allocating 52
 allocation example using IDCAMS 52
 converting PDSs to PDSEs 49, 51—52
 definition 47
 displaying information about 53
 filtering 49
 sample data class filtering routine 50
 sample storage class filtering routine 51
 identifying 53, 54
 sharing of 48
 performance
 I/O
 monitoring 175
 reporting 177
 statistics by data set 176
 statistics by storage class 176

- performance (*continued*)
 - integrated catalog facility catalogs 32
 - shared data in a sysplex 92
- performance parameters, catalogs 32
- permanent data
 - cataloging 23
 - description 21
- policies, data set 21—38
- protect-all option 148
- protecting SMS facilities 152

R

- RACF (Resource Access Control Facility) 145
 - access to database data 157
 - advantages of generic profiles 147
 - backing up RACF data sets 111
 - control data set protection 150
 - commands allowing ALTER access 151
 - commands allowing READ access 151
 - commands allowing UPDATE access 151
 - controlling the use of storage classes and management classes 154
 - RACF command example 155
 - DASD erase-on-scratch 147
 - default SMS classes 154
 - description 2
 - EXECUTE access authority 148
 - facility resource class 153
 - generic profiles 147
 - global access checking 148
 - invoking 145
 - levels of authority 148
 - load module execution 152
 - RACF command example 153
 - profile fields access 149
 - RACF command example 149
 - protect-all option 148
 - protecting control data sets
 - DFSMSHsm 151
 - DFSMSrmm 151
 - protecting DASD and tape data sets 157
 - protecting SMS facilities
 - activating a configuration 152
 - protecting SMS control data sets 150
 - updating RACF profiles 152
 - use of catalog and DFSMSdss functions 152
 - use of ISMF functions 152
 - protection of data sets and objects 156
 - RACF FACILITY resource class 152
 - RACF FIELD resource class 152
 - RACF PROGRAM resource class 152
 - recovering RACF data sets 135
 - restricting use of concurrent copy 106
 - security categories 146
 - storage administrator authorization requirements 158

- recovery
 - developing a recovery plan 129
 - resolving naming conflicts during 143
 - using ABARS 136
 - using DFSMSHsm 130
 - DFSMSHsm HRECOVER command 130
 - DFSMSHsm RECOVER command example 131
 - DFSMSHsm RECOVER command example for physical data set restore 131
 - using IAFC 135
- release of allocated but unused space
 - automating 68
 - issuing command 70
 - non-VSAM 69
 - VSAM 69
- RELEASE parameter of TSO ALLOC 70
- remotely accessed data
 - backing up 102
- resource owner 154
- RETPD, EXPDT keywords 37
- retrieving archives
 - example of using DFSMSdss 135
 - using DFSMSdss 135
- RLSE parameter 70
- RMF (Resource Measurement Facility)
 - reporting object data performance 178
 - transaction reports 179

S

- SAF (System Authorization Facility)
 - invoking RACF 145
- SAM data sets
 - compaction of 40
- SCDS (source control data set)
 - description 13
 - protecting 150
- SDSP (small data set packing) 81
- security
 - access authority 148
 - categories 146
 - of DFSMSHsm control data sets 151
 - of DFSMSrmm control data sets 151
 - of SMS control data sets 150
 - of tape data and volume using DFSMSrmm 157
- serialization between systems 17
- service-level agreement 21, 40, 61, 63, 93, 127
- SETOAM command
 - assigning data class 39
 - objects on tape 42
- shared data
 - migrating 75
 - planning management class for 61
- sharing
 - DASD volumes 16
 - data sets 14

- SMS (Storage Management Subsystem)
 - aggregate group
 - overview 11
 - base configuration 13
 - control data sets
 - ACDS (active control data set) 13
 - backing up 110, 113
 - COMMDS (communications data set) 13
 - overview 13
 - protecting 150
 - recovering 135
 - SCDS (source control data set) 13
 - data class
 - overview 6
 - management class
 - overview 8
 - managing data 4
 - storage class
 - overview 7
 - storage group
 - overview 9
- space management
 - average block allocations 45—46
 - converting PDSs to PDSEs 49, 51—52
 - creating PDSE data sets 52
 - data set migration 71
 - object class transition 77
 - release of allocated but unused space 68
 - removing dependencies on volume serial number 41
 - system-determined block sizes 46
 - using data class to standardize allocation 39
 - using default device geometry 42
 - using default unit 41
- space use, using DCOLLECT to monitor 159
- spill processing 98
- split cylinder data sets 23
- SQL (Structured Query Language) 101
- STGADMIN HLQ 152
- storage class
 - attributes
 - availability 97
 - performance 83
 - defining 85
 - objects 8
 - overview 7
 - PDSE filtering 50
 - sample ACS routine 51
 - planning 84
 - storing objects 84
 - volume selection 85
- storage group
 - overview 9
 - volume selection 9
- STORCLAS keyword 34

- sysplex
 - improving performance of shared data 92
 - improving system availability 108
- system data, description 22
- system group
 - coexistence 18—19
 - description 17
- system-managed storage
 - benefits 2
 - definition 1
 - managing data 4

T

- tape data
 - analyzing with volume mount analyzer 187—208
 - categorizing by usage type 188
 - categorizing for tape mount management 188
 - compaction of 40
 - expiration and retention of 64
 - specifying format requirements 40
 - specifying media type 40
- tape data set
 - storage class 5
- Tape Library Dataserver
 - automated 23
 - benefits of 94
 - manual 23
 - using with software facilities of DFSMS/MVS 94
- tape volumes
 - eligibility for system management 23
- temporary data sets
 - allocating 91
 - deleting 65
 - description 21
 - DFSORT 91
 - PDSEs 91
 - VSAM 91
- the volume mount analyzer
 - description 187
- threshold processing 72
- TSO ALLOC 71
- TSO ALLOC RELEASE parameter 70
- TSO data sets 21
- TSO FREE 71
- TSO TRANSMIT and RECEIVE commands 14

U

- uncataloged data, deletion of 67
- unexpired data, deletion of 66
- Unit Control Blocks (UCBs) 19
- unmovable data sets 23
 - absolute track allocation 23
 - EXCP 23
 - split cylinder allocation 23

unused space, release of 70

V

versions of backup 98

vital record specification 64, 116, 121

vital records

- definition 115

- managing with DFSMSrmm 116

- retrieving 135

- storing using DFSMSdss 116

volume

- DASD (direct access storage device)

 - selection 9

- selection, using storage class 85

volume mount analyzer

- benefits of 94

- categorizing tape data 188

- description 94

- producing reports on tape data 190

volume serial numbers, remove dependence on 41

VSAM data sets 22

- compaction of 40

VSAM passwords

- elimination of in system-managed environment 145

VTOC (volume table of contents)

- compatibility between MVS/XA and MVS/ESA 16

W

workstation data

- backing up in MVS 102

- eligibility for system management 23

- migrating in MVS 76

X

XMIT (TSO TRANSMIT) command 14

Readers' Comments

**MVS/ESA Storage Management Library
Managing Data**

Publication No. SC26-3124-02

Please send us your comments about this printed book, or its softcopy version on CD-ROM. You can also send us comments on product-related online help panels, if applicable.

In sending information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Note: Do not use this form to request IBM publications. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Mail the completed form to the address on the reverse side. For postage-paid mailing from a country other than the United States, give it to your IBM representative or to the IBM branch office serving your locality.

You can also send us comments by fax:

United States: 1-800-426-6209

You can send us comments electronically:

IBMLink from U.S.: STARPUBS at SJSVM28
IBMLink from Canada: STARPUBS at TORIBM
IBM Mail Exchange: USIB3VVD at IBMMAIL
Internet: starpubs at sjsvm28.vnet.ibm.com
or, you can use
starpubs@sjsvm28.vnet.ibm.com

Name

Address

Company or Organization

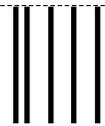
Phone No.



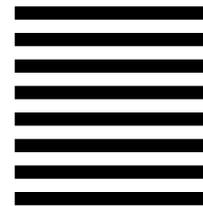
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department M29, Building 86
International Business Machines Corporation
5600 Cottle Road
SAN JOSE CA 95123-9988



Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.

MVS/ESA Storage Management Library

SC26-3123 Implementing System-Managed Storage
SC26-3126 Leading a Storage Administration Group
SC26-3124 Managing Data
SC26-3125 Managing Storage Groups

SC26-3124-02



Spine information:



MVS/ESA SML

Managing Data

SC26-3124-02