z/OS
Version 2 Release 3

DFSMS Introduction

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 81.

This edition applies to Version 2 Release 3 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2019-02-15

Contents

List of Figures

List of Tables

x

About This Information

This information describes the primary functions that DFSMSdfp, DFSMSdss, DFSMShsm, DFSMSrmm, and DFSMStvs provide within the z/OS operating system. It also discusses system-managed storage, an enterprise-based solution that combines hardware, software, and installation-defined policies to enable a system to automatically perform storage tasks that system administrators and users would otherwise need to perform manually.

For information about the accessibility features of z/OS, for users who have a physical disability, see Appendix A, "Accessibility," on page 77.

Required product knowledge

This document is intended for the executive, data processing manager, or z/OS® system programmer who is interested in learning about z/OS DFSMS.

If you are trying to decide whether to implement system-managed storage at your installation, read the first three chapters of this book. If you want to learn how to process z/OS and UNIX data sets, read Chapter 4, "Managing Data with DFSMSdfp," on page 25 and Chapter 5, "Using Access Methods, Commands, and Utilities," on page 31. If you are trying to decide whether to exploit DFSMSdss, DFSMShsm, and DFSMSrmm, read the first two chapters and the chapter about the particular DFSMS element.

Referenced documents

For information on all the new functions in the current z/OS DFSMS release, see *z/OS DFSMS Using the New Functions*. Within the text, references are made to other DFSMS documents, as well as to documents of related products. The titles and order numbers of referenced publications in the z/OS library are listed in the *z/OS Information Roadmap*.

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, go to IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xiii.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
> If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Knowledge Center function
> If your comment or question is about the IBM Knowledge Center functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Knowledge Center Support at ibmkc@us.ibm.com.

Feedback on the z/OS product documentation and content
> If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

> To help us better process your submission, include the following information:

> * Your name, company/university/institution name, and email address
> * The following deliverable title and order number: z/OS DFSMS Introduction, SC23-6851-30
> * The section title of the specific information to which your comment relates
> * The text of your comment.

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

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

* Go to the IBM Support Portal (support.ibm.com).
* Contact your IBM service representative.
* Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Summary of changes for z/OS V2R3 DFSMS Introduction

The following changes are made for z/OS Version 2 Release 3 (V2R3).

Changed

- Updated "Backing Up and Restoring Data" on page 47 to include backing up data to a cloud.
- Updated "Extended-Format Data Sets" on page 26 to include data set level encryption support.

z/OS Version 2 Release 1 summary of changes

See the Version 2 Release 1 (V2R1) versions of the following publications for all enhancements related to z/OS V2R1:

- *z/OS Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Introduction and Release Guide*

Chapter 1. Understanding the DFSMS Environment

This chapter gives a brief overview of Data Facility Storage Management Subsystem (DFSMS) and its primary functions in the z/OS operating system. DFSMS comprises a suite of related data and storage management products for the z/OS system. DFSMS is an operating environment that helps automate and centralize the management of storage based on the policies that your installation defines for availability, performance, space, and security. The heart of DFSMS is the Storage Management Subsystem (SMS). Using SMS, the storage administrator defines policies that automate the management of storage and hardware devices. These policies describe data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements for the system.

Related reading:

- For information on DFSMS operating requirements, see *z/OS Planning for Installation*.
- For DFSMS migration and coexistence information, see *z/OS Migration*.

DFSMS in the z/OS Operating System

DFSMS is an exclusive element of the z/OS operating system. DFSMS is a software suite that automatically manages data from creation to expiration. The following elements comprise DFSMS:

- DFSMSdfp, a base element of z/OS
- DFSMSdss, an optional feature of z/OS
- DFSMShsm, an optional feature of z/OS
- DFSMSrmm, an optional feature of z/OS
- DFSMStvs, an optional feature of z/OS

The z/OS operating system enables you to efficiently manage e-business workloads and enterprise transactions 24 hours a day. DFSMSdfp is automatically included with z/OS. DFSMSdfp performs the essential data, storage, and device management functions of the system. DFSMSdfp and DFSMShsm provide disaster recovery functions such as Advanced Copy Services and aggregate backup and recovery support (ABARS).

The other elements of DFSMS—DFSMSdss, DFSMShsm, DFSMSrmm, and DFSMStvs—complement DFSMSdfp to provide a fully-integrated approach to data and storage management. In a system-managed storage environment, DFSMS automates and centralizes storage management based on the policies that your installation defines for availability, performance, space, and security. With these optional features enabled, you can take full advantage of all the functions that DFSMS offers.

Storage management policies reduce the need for users to make many detailed decisions that are not related to their business objectives.

DFSMS Elements

This section describes the roles of each DFSMS element. illustrates the data, storage, and device management functions that DFSMS performs.
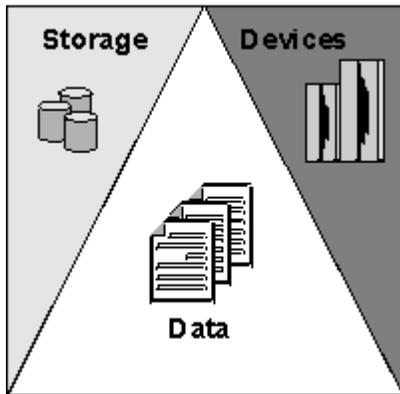
Figure 1: DFSMS data, storage, and device management functions

DFSMSdfp

DFSMSdfp provides the following functions:

Storage management
DFSMSdfp includes Interactive Storage Management Facility (ISMF), which lets you define and maintain policies to manage your storage resources. These policies help to improve the usage of storage devices and to increase levels of service for user data, with minimal effort required from users. Storage Management Subsystem (SMS) uses these policies to manage storage for the operating system. More specifically, the storage administrator uses SMS to define data classes, storage classes, management classes, storage groups, aggregate groups, copy pools, and automatic class selection routines. You can also use the NaviQuest tool under ISMF to migrate to SMS, maintain your SMS configuration, and perform testing, implementation, and reporting tasks in batch. For more information on storage, ISMF, and NaviQuest, see Chapter 3, "Managing Storage with DFSMSdfp," on page 17.

Tape mount management
Tape mount management is a methodology for improving tape usage and reducing tape costs. This methodology involves intercepting selected tape data set allocations through the SMS automatic class selection (ACS) routines and redirecting them to a direct access storage device (DASD) buffer. Once on DASD, you can migrate these data sets to a single tape or small set of tapes, thereby reducing the overhead associated with multiple tape mounts. For more information, see *z/OS DFSMS Using the Volume Mount Analyzer* and "Implementing Tape Mount Management" on page 70.

Data management
DFSMSdfp helps you store and catalog information on DASD, optical, and tape devices so that it can be quickly identified and retrieved from the system. DFSMSdfp provides access to both record- and stream-oriented data in the z/OS environment. For more information, see Chapter 4, "Managing Data with DFSMSdfp," on page 25 and Chapter 5, "Using Access Methods, Commands, and Utilities," on page 31.

Device management
DFSMSdfp can be used when you define your input and output (I/O) devices to the system and in controlling those devices in the z/OS environment. For more information, see Chapter 6, "Managing Devices with DFSMSdfp," on page 37.

Distributed data access
Distributed data access allows all authorized systems and users in a network to use system-managed storage or automated storage management. DFSMSdfp uses the Distributed FileManager/MVS (DFM) or the z/OS Network File System to enable remote clients in a network to access data and storage resources on z/OS systems. For more information, see Chapter 7, "Distributed Data Access with DFSMSdfp," on page 43 and *z/OS Network File System Guide and Reference*.

z/OS UNIX System Services (z/OS UNIX) provides the command interface that interactive UNIX users can use. z/OS UNIX allows z/OS programs to directly access UNIX data.

Advanced copy services

Advanced Copy Services includes remote and point-in-time copy functions that provide backup and recovery of data. When used before a disaster occurs, Advanced Copy Services provides rapid backup of critical data with minimal impact to business applications. If a disaster occurs to your data center, Advanced Copy Services provides rapid recovery of critical data. For more information, see "DFSMS Advanced Copy Services" on page 4.

Object access method

Object access method (OAM) provides storage, retrieval, and storage hierarchy management for objects. For more information, see "Objects" on page 27. OAM also manages storage and retrieval for tape volumes that are contained in system-managed libraries.

DFSMSdss

DFSMSdss provides the following functions:

Data movement and replication

DFSMSdss lets you move or copy data between volumes of like and unlike device types. If you create a backup in DFSMSdss, you can copy a backup copy of data. DFSMSdss also can produce multiple backup copies during a dump operation.

Space management

DFSMSdss can reduce or eliminate DASD free-space fragmentation.

Data backup and recovery

DFSMSdss provides you with host system backup and recovery functions at both the data set and volume levels. It also includes a stand-alone restore program that you can run without a host operating system.

Data set and volume conversion

DFSMSdss can convert your data sets and volumes to system-managed storage. It can also return your data to a non-system-managed state as part of a recovery procedure.

Related reading: For more information on each of the DFSMSdss functions, see Chapter 8, "Understanding DFSMSdss," on page 45.

DFSMShsm

DFSMShsm complements DFSMSdss to provide the following functions:

Storage management

DFSMShsm provides automatic DASD storage management, relieving users from manual storage management tasks.

Space management

DFSMShsm improves DASD space usage by keeping only active data on fast-access storage devices. It automatically frees space on user volumes by deleting eligible data sets, releasing overallocated space, and moving low-activity data to lower cost-per-byte devices, even if the job did not request tape.

Tape mount management

DFSMShsm can write multiple output data sets to a single tape, making it a useful tool for implementing tape mount management under SMS. When you redirect tape data set allocations to DASD, DFSMShsm can move those data sets to tape, as a group, during interval migration. This methodology greatly reduces the number of tape mounts on the system. DFSMShsm uses a single-file format, which improves your tape usage and search capabilities.

Availability management

DFSMShsm backs up your data—automatically or by command—to ensure availability if accidental loss of the data sets or physical loss of volumes should occur. DFSMShsm also allows the storage administrator to copy backup and migration tapes, and to specify that copies be made in parallel with the original. You can store the copies on site as protection from media damage, or offsite as protection from site damage. DFSMShsm also provides disaster backup and recovery for user-defined groups of

data sets (aggregates) so that you can restore critical applications at the same location or at an offsite location.

Requirement: You must also have DFSMSdss to use the DFSMShsm functions.

Related reading: For more information, see Chapter 9, "Understanding DFSMShsm," on page 53.

DFSMSrmm

DFSMSrmm manages your removable media resources, including tape cartridges and reels. It provides the following functions:

Library Management
You can create tape libraries, or collections of tape media associated with tape drives, to balance the work of your tape drives and help the operators that use them.

DFSMSrmm can manage the following devices:

- A removable media library, which incorporates all other libraries, such as:
  - System-managed manual tape libraries
  - System-managed automated tape libraries

    Examples of automated tape libraries include IBM TotalStorage Enterprise Automated Tape Library (3494) and IBM TotalStorage Virtual Tape Servers (VTS).
- Non-system-managed or traditional tape libraries, including automated libraries such as a library under Basic Tape Library Support (BTLS) control.

Shelf Management
DFSMSrmm groups information about removable media by shelves into a central online inventory, and keeps track of the volumes residing on those shelves. DFSMSrmm can manage the shelf space that you define in your removable media library and in your storage locations.

Volume management
DFSMSrmm manages the movement and retention of tape volumes throughout their life cycle.

Data set management
DFSMSrmm records information about the data sets on tape volumes. DFSMSrmm uses the data set information to validate volumes and to control the retention and movement of those data sets.

Related reading: For more information, see Chapter 10, "Managing Removable Media with DFSMSrmm," on page 73.

DFSMStvs

DFSMS Transactional VSAM Services (DFSMStvs) allows you to share VSAM data sets across CICS®, batch, and object-oriented applications on z/OS or distributed systems. DFSMStvs enables concurrent shared updates of recoverable VSAM data sets by CICS transactions and multiple batch applications. DFSMStvs enables 24-hour availability of CICS and batch applications. DFSMStvs is built on top of VSAM record-level sharing (RLS), which permits sharing of recoverable VSAM data sets at the record level.

Related reading: For more information, see the following publications:

- *z/OS DFSMStvs Planning and Operating Guide*
- *z/OS DFSMStvs Administration Guide*
- *z/OS DFSMS Using Data Sets*

DFSMS Advanced Copy Services

Advanced Copy Services, offered on IBM TotalStorage Enterprise Storage Server® (ESS), consists of the following individual functions and enhancements:

- Remote copy, which consists of the following:

- Extended Remote Copy (XRC): XRC is a combined hardware and software solution that offers the highest levels of continuous data availability in a disaster recovery and workload movement environment. Offered on the ESS, XRC provides an asynchronous remote copy solution, of both system-managed and non-system-managed data, to a second, remote location.
- Coupled extended remote copy (CXRC) and Multiple Extended Remote Copy (MXRC): You can use CXRC and MXRC in large environments that have an expanded number of primary storage controls and DASD volumes, in excess of those supported by a single data mover configuration.
- Peer-to-Peer Remote Copy (PPRC): PPRC is a hardware solution to rapid and accurate disaster recovery as well as a solution to workload and DASD migration.

- FlashCopy®: FlashCopy is a point-in-time copy services function that can quickly copy data from a source location to a target location.
- Global Mirror: Global Mirror is a disaster recovery environment that uses a combination of PPRC and FlashCopy to maintain consistent up-to-date volumes at the remote disaster recovery site.
- Concurrent copy: Concurrent copy is a copy services function that generates a copy of data while applications are updating that data.
- SnapShot copy: SnapShot copy is an Advanced Copy Services function offered on the RAMAC Virtual Array (RVA) Storage Subsystem. SnapShot is a point-in-time copy services function that allows you to "snap" (quickly copy) data directly from the source location to a target location.

Many of these functions rely on the System Data Mover (SDM). The SDM is the high-speed data transfer engine behind the Extended Remote Copy, SnapShot, and concurrent copy functions. Peer-to-Peer Remote Copy does not rely on the SDM for data movement.

Related reading: For further information on Advanced Copy Services, see *z/OS DFSMS Advanced Copy Services*.

Chapter 2. An Overview of System-Managed Storage

This chapter briefly discusses how DFSMS can help you manage your storage. It is intended as a high-level overview. For a more detailed discussion of how each of the DFSMS elements can help you manage storage, see the subsequent chapters in this book.

How DFSMS Can Solve Your Storage Problems

As your business expands, so do your needs for storage to hold your applications and data, and the costs of managing that storage. Storage costs include more than the price of the hardware, with the highest cost being the people needed to perform storage management tasks. Additionally, you must pay for people to install, monitor, and operate your storage hardware devices, for electrical power to keep each piece of storage hardware cool and running, and for floor space to house them. Removable media, such as optical and tape storage, cost less per gigabyte than online storage. However, removable media requires additional time and resources to locate, retrieve, and mount.

Business systems need to be available continuously, efficient, easy to use, and secure from unauthorized users. System outages can be costly because users cannot access the system and data loss might result. As businesses grow, the systems tend to become extremely complicated and difficult to maintain, which results in lost productivity and time spent troubleshooting problems.

DFSMS elements, together with IBM hardware products, and your installation-specific requirements for data and resource management comprise the key to system-managed storage in a z/OS environment. The elements of DFSMS automate and centralize storage management, based on policies your installation defines for availability, performance, space, and security. The Interactive Storage Management Facility (ISMF) provides the user interface for defining and maintaining these policies, while the Storage Management Subsystem (SMS) governs the system.

DFSMS provides functions that reduce the occurrence of system outages, and enhance disaster recovery capabilities and system security. DFSMS also improves your business efficiency by providing better system performance and throughput, and usability enhancements that increase storage administrator productivity.

Implementing Your Storage Management Policies

To allow your business to grow efficiently and profitably, you want to find ways to control the growth of your information systems and use your current storage more effectively.

In an SMS-managed storage environment, your enterprise establishes centralized policies for how to use your hardware resources. These policies balance your available resources with your users' requirements for data availability, performance, space, and security. SMS implements these policies and manages most of your storage management tasks. This frees users from manually administering storage and makes more efficient use of your storage resources.

The policies defined by your installation represent decisions about your resources, such as:

- What performance objectives are required at your site?
- When and how to back up data?
- Whether data sets should be kept available for use during backup or copy?
- How to manage backup copies kept for disaster recovery?
- What to do with data that is obsolete or seldom used?

To implement a policy for managing storage, your storage administrator defines classes of space management, performance, and availability requirements for data sets at your installation. For example, the administrator can define one storage class for data entities requiring high performance and another

7

for those requiring standard performance. Then, the administrator writes automatic class selection (ACS) routines that use naming conventions, or other criteria of your choice, to automatically assign the classes that have been defined to data as that data is created. These ACS routines can then be validated and tested.

When the ACS routines are started and the classes (also referred to as *constructs*) are assigned to the data, SMS uses the policies defined in the classes and applies them to the data for the life of the data. Additionally, devices with various characteristics can be pooled together into storage groups so that new data can be automatically placed on devices that best meet the needs of the data.

The ISMF panels make it easy to define SMS classes and groups, test and validate ACS routines, and perform other tasks to analyze and manage your storage. Many of these functions are also available in batch through NaviQuest. For more information about NaviQuest, see the *z/OS DFSMSdfp Storage Administration*.

Monitoring Your Policies

After you have established your installation's service levels and implemented policies based on those levels, you can use DFSMS facilities to see if your objectives have been met. Information on past use can help you develop more effective storage administration policies and manage growth effectively.

Volume Mount Analyzer

DFSMS reporting facilities help you set policies for placing or migrating data onto DASD or tape. With the volume mount analyzer tool, you can print DFSMS reports on tape usage to obtain statistics about how often tapes are mounted or how often tape data sets are used. These reports can help you decide which classes of data you want to store onto DASD or tape and how long you want data to remain in active storage before being moved.

Related reading: For more information, see *z/OS DFSMS Using the Volume Mount Analyzer*.

DASD Volume Usage

DFSMS provides access method services to examine DASD volume usage, and to produce a data set containing the results. Looking at the past history and growth of your storage can help you decide when to add DASD volumes to accommodate future growth. This function extracts information about data sets on DASD and migrated and backed-up data from DFSMShsm control data sets. You can use this information for billing or analysis.

Related reading: For more information about access method services, see *z/OS DFSMS Access Method Services Commands*.

Data Set Performance

DFSMS provides statistics in system management facility (SMF) records that you can use to analyze data set performance. DFSMS also provides a data set performance metric, *I/O delay*, which provides batch job analysis.

The IBM TotalStorage DFSMS Optimizer uses these records to analyze and recommend changes to optimize performance of data sets and volumes. The DFSMS Optimizer can also monitor and tune multiple DFSMShsm address spaces in a Parallel Sysplex®. The DFSMS Optimizer can help you maximize storage use and minimize storage costs.

Related reading: For more information, see the following sources:

- *z/OS MVS System Management Facilities (SMF)*
- *DFSMS Optimizer User's Guide and Reference*
- DFSMS Optimizer website (www.storage.ibm.com/optimizer)

DASD Free Space

DFSMShsm provides a reporting facility to help you decide when you want to consolidate your data on DASD to eliminate or minimize free space fragmentation. You can print a report that gives you information about your DASD, including how much space is unused on each track, what percentage of space is unused, and how many tracks are empty.

Related reading: For more information, see *z/OS DFSMSdfp Storage Administration*.

Using Storage Devices

SMS simplifies the process of allocating data on various types of storage devices.

Data Set Allocation Templates

The storage administrator provides standard models or templates, each with a set of commonly used logical data set attributes that determine the processing characteristics of a data set. Users can reference these templates when creating data sets instead of having to continually re-specify frequently needed attributes in JCL.

Selection of Storage Devices

SMS can select the type of storage device that is most appropriate for your data, based on criteria that you set. SMS places the data on an appropriate tape volume, optical disk volume, or DASD.

DASD Volumes

You can use your DASD volumes more efficiently by allowing SMS to manage data placement. For example, SMS can help you avoid running out of space and wastes system resources by repeatedly migrating and recalling relatively active data, while another user group has excess space that is either unused or used for data that is rarely needed.

Tape Libraries

With SMS and system-managed tape, you can manage your tape volumes and drives by using ISMF to define one or more automated tape libraries or manual tape libraries. These definitions, along with other SMS constructs and ACS routines, enable you to manage your tape environment more effectively. A *tape library* is a named set of related tape volumes and the set of tape drives where those volumes are mounted. You can associate each tape library with one or more tape storage groups to direct a new allocation request to a specific set of tape libraries. Existing tape data sets are allocated to tape devices within the library in which the volume resides. Associating tape volumes with a tape library ensures that SMS mounts tape volumes on tape devices that belong to that tape library.

Optical Libraries

You can manage your optical drives more efficiently by grouping them and defining optical libraries using ISMF. These definitions are used with other SMS constructs, ACS routines, and the object access method (OAM) to manage the object data. You can then use automated optical libraries with pseudo optical libraries to more efficiently manage the placement and migration of your object data.

With OAM, rewritable optical disks in automated optical libraries can be automatically reused when all the data has expired. When all the data expires on optical disks that are not rewritable, the disks can be removed from the automated optical library. For more information, see "Objects" on page 27.

Virtual Tape Servers, Devices, and Volumes

You can reduce the number of tape cartridges, devices, and automated tape libraries by implementing an IBM TotalStorage Virtual Tape Server (VTS) or an IBM TotalStorage Peer-to-Peer Virtual Tape Server (PtP VTS). This virtual tape subsystem consists of virtual tape devices, virtual tape volumes, tape volume cache (DASD), and hierarchical storage management software.

The PtP VTS addresses data availability, system availability, remote copy, and data vaulting needs by coupling together two stand-alone VTSs. Virtual tape controllers provide replication, transparent tracking, and synchronization of the logical tape volumes. The VTS appears as one library image to the attached z/OS hosts. It provides multiple copies of all tape data. It also supports an immediate or deferred copy option.

DFSMS supports the importing and exporting of VTS tape data. For example, VTS tape data can be exported (physically ejected) from a virtual library for storage management tasks, such as application migration, data interchange, offsite vaulting, exterior (nonlibrary) shelf storage, or disaster recovery.

For more information on managing VTS tape data, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Outboard Policy Management

Outboard policy management enhancements for VTS tape libraries allow you to more effectively manage your physical and logical tape volumes. This support utilizes the names of the existing SMS management constructs: storage group, management class, storage class, and data class. You use the ACS routines to assign the eight-character construct names, which are then passed to the library at mount time. The construct names and actions are defined outboard using the library manager.

The assigned constructs, which are policy names stored by the library, become attributes of the tape volume. A VTS then manages a tape volume according to the volume's assigned policy names and the actions associated with the policy names. The policy names and their associated actions are defined at the library.

With the outboard policy management capability, you can use the library manager storage group policy to specify a stacked volume pool for a group of logical volumes, thus giving you the ability to manage the volumes by groups. This is known as physical volume pooling. You can also implement selective dual copy capability by defining a secondary stacked volume pool through the library manager management class policy. Using the library manager management class, you can determine whether the PtP VTS makes a volume's copy immediately or defers it. Additionally, using the library manager storage class, you can define a cache preference level for a volume. This indicates the length of time that volume remains in the tape volume cache, thus improving mount response time.

Related reading: For more information on outboard policy management, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Managing Data Storage

After you allocate a data set, use DFSMSdfp, DFSMShsm, DFSMSdss, DFSMSrmm, and DFSMStvs to manage your data storage more efficiently.

Related reading: For more information, see *z/OS DFSMShsm Storage Administration* and *z/OS DFSMSdss Storage Administration*.

Data Compression

You can compress VSAM or sequential extended-format data sets. Data compression lets you use your system buffers, I/O channels, and DASD more efficiently. You can also use the IBM RAMAC Virtual Array Subsystem to further compress this data, as well as all other data in the subsystem.

Space Availability on DASD Volumes

DFSMShsm makes sure that space is available on your DASD volumes so that you can extend your old data sets and allocate new ones. DFSMShsm automatically deletes expired or temporary data on your DASD volumes and lets you reuse that space for new data. DFSMSdss eliminates or reduces the fragmentation of free space. Your installation can choose to have DFSMShsm automatically reclaim your excess requested space once a day, or choose to have DFSMSdfp reclaim excess requested space each time a data set is closed.

Space Availability on Tape Volumes

When your DFSMShsm tapes contain obsolete or expired data, use DFSMShsm to transfer the remaining valid data from multiple tapes onto fewer tapes. (Obsolete or expired data include recalled data sets and expired backup copies.) Consolidating the data in this manner allows you to use your tapes more economically. You can define whether to return the expired tape volumes to the DFSMShsm scratch pool or specify MVS™ scratch status.

After DFSMShsm or other applications no longer need a particular tape volume, DFSMSrmm can reuse that tape volume.

Tapes that contain OAM objects can be reused automatically when all the data has expired. You can define whether to return expired tape volumes to OAM scratch status or MVS scratch status, or retain the tape volumes in the object or object backup storage group.

Migration of Low-Activity Data

If you have data on your DASD that you do not use very often, DFSMShsm can automatically move that data to slower and cheaper devices. For example, DFSMShsm can move your low-activity data sets onto slower DASD or tape volumes. DFSMShsm saves on storage costs by automatically compacting or compressing your low-activity data. You can use OAM to move your low-activity objects onto slower access media such as tape or optical volumes.

Recall of Migrated Data

DFSMShsm automatically recalls migrated data when you need it, using DFSMSdss. Whenever DFSMSdss moves a data set to a new device, it automatically converts the data if the different device type requires it. You can also use DFSMSdss to move data between devices of different types or between system-managed volumes and non-system-managed volumes. DFSMSdss automatically converts the data if required.

Expired Data Sets

You can assign a management class to SMS-managed data sets to indicate how long to retain the data sets. DFSMShsm automatically deletes eligible DASD data sets during its space management cycle based on this management class specification. You can set tape volume expiration policies with DFSMSrmm.

Backing Up and Recovering Data

DFSMS helps ensure that your data remains available by automatically backing up your data according to the service levels that you define. If your original data is accidentally lost, you can replace your originals with backup copies. With DFSMS, you can perform the following data availability functions:

- Automatically back up your DASD data
- Prepare for disaster recovery with aggregate groups
- Perform backups and make copies of data easily and nondisruptively
- Access data during operating system outages or hardware outages
- Create a fast replication of a copy pool

Automatic Data Backups

DFSMShsm provides several automatic backup capabilities that work together. By having DFSMShsm automatically back up any new or changed data, you can keep your backups current but reduce how often you have DFSMShsm back up entire DASD volumes. When you want to recover a data set or a volume, DFSMShsm uses all sets of backups to provide you with the most current version.

Aggregate Group Backup and Recovery

DFSMShsm provides the aggregate group backup and recovery support (ABARS), another solution for the backup and recovery of data sets. ABARS lets you define critical applications and back up each application as an aggregate group. An *aggregate group* is a group of related data sets that require concurrent action. If a disaster occurs, you can use ABARS to restore critical applications after restoring the base operating systems. If you identify and back up only critical data, you can reduce the total amount of data going offsite for disaster recovery.

You can define policies specifying how you want the aggregate copies to be managed, such as how many copies of versions you want, or how long you want to keep your extra versions, and then allow DFSMS to automatically implement those policies. You can use DFSMSrmm to manage the movement of both ABARS and DFSMShsm backup tapes. DFSMSrmm keeps track of where tapes are stored and can provide you with a list of tapes you need to move offsite for storage or to retrieve in case of an unrecoverable error. ABARS can also be used for application transfer from one site to another. Host systems can act as disaster recovery sites for each other when you use tape library partitioning support to set up sets of tape libraries at different locations accessed by the host systems.

Fast Replication Backup

DFSMS provides a volume-level fast replication that allows you to create a point-in-time backup that is managed by DFSMShsm with minimal application outage. The fast replication function only supports the FlashCopy and SnapShot functions.

By using the copy pool SMS construct, you can define a set of storage groups that DFSMShsm processes collectively for fast replication functions. DFSMShsm recovers a fast replication backup version from the volume or copy pool level, but not at the data set level.

Point-in-Time Copy Functions

DFSMS supports the following functions on Enterprise Storage Server and RAMAC Virtual Array that allow you to copy your data quickly and easily.

FlashCopy:
FlashCopy is available on the IBM Enterprise Storage Server (ESS). DFSMSdss uses this function to quickly copy data from source volumes to target volumes that are within the same logical storage subsystem (LSS) or ESS (depending on the ESS hardware). Since copying your data is done very quickly, your applications need only be stopped for a very short time period.

SnapShot:
SnapShot is available on the RAMAC Virtual Array (RVA). This function quickly copies data from source volumes to target volumes that are within the same LSS. SnapShot allows you to copy a specific set of tracks, an entire volume, or a data set. The copy operation is completed with only a few I/Os to the device. There are two kinds of SnapShot copy. *Native SnapShot* copies data directly to the target device. *Virtual-concurrent-copy SnapShot* copies data to an intermediate location (where it is logically complete) and then to the target device (where it is physically complete).

Concurrent Copy:
Concurrent copy is a hardware and software solution that allows you to back up a database or any collection of data at a point in time and with minimum down time for the database. The database is unavailable only long enough for DFSMSdss to initialize a concurrent copy session for the data, which is a very small fraction of the time that the complete backup will take. The copy that is made does not include any of the update activity; it is as if the backup were made instantaneously when it was requested. After initialization, DFSMSdss releases all the serialization it holds on the data, informs the user that the initialization is complete so that update activity can resume, and begins reading the data. Concurrent copy is supported on 3990 storage subsystems. Concurrent copy is supported on ESS subsystems in situations where FlashCopy is not available, such as when source and target volumes are in different ESS subsystems.

Related reading: For more information on point-in-time copy functions, see *z/OS DFSMSdss Storage Administration* and *z/OS DFSMS Advanced Copy Services*.

Backup-While-Open

The backup-while-open (BWO) function performs a backup of CICS VSAM-file control data sets and other VSAM data sets that are in-use and open for update. While the data set is being backed up, an application program can update the data set in a manner that invalidates the backup data set. Concurrent copy improves BWO processing by significantly reducing the chances of invalidating a BWO dump because of updates to the data sets.

Data Access During Operating System or Hardware Outages

DFSMS can provide you with access to your data even during operating system or hardware outages. For example, you can restore data from tapes when the operating system is not functional. You can also maintain access to critical data sets on DASD in case of device problems by allowing DFSMS to maintain dual copies of your key DASD volumes.

Managing Your Data with Copy Services

DFSMS provides Advanced Copy Services that includes a hardware and software solution to help you manage and protect your data. These solutions help ensure that your data remains available 24 hours a day, seven days a week. Advanced Copy Services provide solutions to disaster recovery, data migration, and data duplication. Many of these functions run on the IBM TotalStorage Enterprise Storage Server (ESS).

With DFSMS, you can perform the following data management functions:

- Use remote copy to prepare for disaster recovery.
- Move your PPRC data more easily.

Disaster Recovery with Remote Copy

Remote copy provides two options for disaster recovery and workload migration: extended remote copy (XRC) and peer-to-peer remote copy (PPRC). These options enable you to maintain a current copy of your data at a remote site.

Related reading: For more information about XRC and PPRC, see *z/OS DFSMS Advanced Copy Services*.

Extended Remote Copy (XRC)

XRC combines hardware and software to provide continuous data availability in a disaster recovery or workload movement environment. XRC provides an asynchronous remote copy solution for both system-managed and non-system-managed data to a second, remote location.

XRC relies on the IBM TotalStorage Enterprise Storage Server, IBM 3990, RAMAC Storage Subsystems, and DFSMSdfp. The 9393 RAMAC Virtual Array (RVA) does not support XRC for source volume capability.

XRC relies on the system data mover, which is part of DFSMSdfp. The system data mover is a high-speed data movement program that efficiently and reliably moves large amounts of data between storage devices. XRC is a continuous copy operation, and it is capable of operating over long distances (with channel extenders). It runs unattended, without involvement from the application users. If an unrecoverable error occurs at your primary site, the only data that is lost is data that is in transit between the time when the primary system fails and the recovery at the recovery site.

Coupled Extended Remote Copy (CXRC) allows XRC sessions to be coupled together to guarantee that all volumes are consistent across all coupled XRC sessions. CXRC can manage thousands of volumes.

IBM TotalStorage XRC Performance Monitor provides the ability to monitor and evaluate the performance of a running XRC configuration.

Related reading: For more information about the XRC Performance Monitor, see *IBM TotalStorage XRC Performance Monitor Installation and User's Guide*.

Peer-to-Peer Remote Copy (PPRC)

PPRC is a hardware solution which provides rapid and accurate disaster recovery as well as a solution to workload movement and device migration. Updates made on the primary DASD volumes are synchronously shadowed to the secondary DASD volumes. The local storage subsystem and the remote storage subsystem are connected through a communications link called a PPRC path. You can use one of the following protocols to copy data using PPRC:

- ESCON
- Fibre Channel Protocol

Requirement: Fibre Channel Protocol is supported only on ESS Model 800 with the appropriate licensed internal code (LIC) level and the PPRC Version 2 feature enabled.

Peer-to-Peer Remote Copy Dynamic Address Switching

PPRC dynamic address switching (P/DAS) provides the ability to redirect all application I/O from one PPRC volume to another PPRC volume, using P/DAS commands. P/DAS operations are based on the PPRC functions of the 3990 Model 6 Storage Control, and can be used in shared-DASD environments, including Parallel Sysplex environments. P/DAS allows application-transparent switching of I/O to support the following tasks:

- Planned outages (device or subsystem)
- Device migration
- Workload movement

Related reading: Refer to *z/OS DFSMS Advanced Copy Services* for details on the P/DAS function.

Peer-to-Peer Extended Distance

When you enable the PPRC extended distance feature (PPRC-XD), the primary and recovery storage control sites can be separated by long distances. Updates made to a PPRC primary volume are sent to a secondary volume asynchronously, thus requiring less bandwidth.

If you are trying to decide whether to use synchronous or asynchronous PPRC, consider the differences between the two modes:

- When you use synchronous PPRC, no data loss occurs between the last update at the primary system and the recovery site, but it increases the impact to applications and uses more resources for copying data.
- Asynchronous PPRC using the extended distance feature reduces impact to applications that write to primary volumes and uses less resources for copying data, but data might be lost if a disaster occurs. To use PPRC-XD as a disaster recovery solution, customers need to periodically synchronize the recovery volumes with the primary site and make backups to other DASD volumes or tapes.

Related reading: Refer to *z/OS DFSMS Advanced Copy Services* for details on the PPRC extended distance feature.

Peer-to-Peer Remote Copy over Fibre Channel Protocol

You can use PPRC with Fibre Channel Protocol (FCP) connections. Using FCP significantly speeds up PPRC link transfer rates and eliminates the need for ESCON infrastructure where it is not needed for anything other than PPRC connections. However, the FCP protocol transfers data more efficiently than ESCON. FCP achieves a throughput greater than four times the fastest ESCON connections.

Related reading: For more information about PPRC over FCP, see the following publications:

- *z/OS DFSMS Advanced Copy Services*
- *IBM TotalStorage Enterprise System Server User's Guide*
- *Device Support Facilities (ICKDSF) User's Guide and Reference*

Metro/Global Mirror

Metro/Global Mirror is a three-site solution, which uses synchronous replication to mirror data between a local site and an intermediate site, and asynchronous replication to mirror data from an intermediate site to a remote site. In this configuration, a Metro Mirror pair is established between two nearby sites (local and intermediate) to protect from local site disasters. The Global Mirror volumes can be located thousands of miles away and would continue to be updated if the original local site has suffered a disaster and I/O has to be failed over to the intermediate site. In the case of a local-site-only disaster, Metro/Global Mirror can provide a zero-data-loss recovery at the remote site as well as at the intermediate site.

Related reading: For additional information about Metro/Global Mirror, see *z/OS DFSMS Advanced Copy Services*.

Managing Removable Media

This section describes how to manage your removable media with DFSMSrmm.

Shelf Locations for Your Removable Media

You can use DFSMSrmm to manage your removable media at the shelf, volume, and data set levels. Shelf locations can be assigned for volumes that are kept available and for volumes that are stored at a vault for vital records or backups.

Tape operators can locate and mount tapes more easily because the drive display and operator mount message tell the tape operator the shelf location of the volume to be mounted. DFSMSrmm automatically records information about data sets on tape volumes so that you can manage the data sets and volumes more efficiently. When all the data sets on a volume have expired, the volume can be reclaimed and reused. You can optionally move volumes that are to be retained to another location.

Reports for Removable Media

DFSMSrmm reporting facilities help you manage your removable media. These reports help you keep track of where your tape volumes are and where they should be moved, by providing you with information about your volumes, shelves, and tape data sets. You can also audit use of your tape data by running security reports that list all occasions when a protected tape data set has been accessed.

The DFSMSrmm Report Generator enables you to easily build customized reports.

Reuse of Removable Media

DFSMSrmm determines when a cartridge in an automated tape library holds only expired data, so that the volume can automatically be reused for new data.

Chapter 3. Managing Storage with DFSMSdfp

The storage management subsystem (SMS) is a DFSMSdfp facility designed for automating and centralizing storage management. SMS automatically assigns attributes to new data when that data is created. SMS automatically controls system storage and assigns data to the appropriate storage device. ISMF panels allow you to specify these data attributes.

This chapter describes the ways that you can assign attributes to data sets and objects through the SMS classes and storage groups, and automatic class selection (ACS) routines.

Interactive Storage Management Facility Menus

ISMF is an Interactive System Productivity Facility (ISPF) application. ISMF is the interface to DFSMS that helps you analyze and manage data and storage. ISMF provides access to the storage, data, space, and tape management, and backup and recovery services that DFSMSdfp, DFSMShsm, DFSMSdss, and DFSMSrmm provide.

In addition, ISMF works with the following products:

- ISPF/PDF, which provides the edit, browse, data set, and library utility functions
- Time Sharing Option/Extensions (TSO/E), which provides a way to issue commands instead of using the ISMF panels
- Data Facility Sort (DFSORT), which provides fast and efficient sorting, merging, copying, reporting, and analysis of your business information
- Resource Access Control Facility (RACF®), which provides security and access control for data sets and resources

Related reading:

- For more information on SMS and ISMF for general users, see *z/OS DFSMS Using the Interactive Storage Management Facility*.
- For more information on ISMF for storage administrators, see *z/OS DFSMSdfp Storage Administration*.
- For more information on ISPF/PDF, see *z/OS ISPF Reference Summary*.
- For more information on TSO/E, see *z/OS TSO/E Command Reference*.
- For more information on DFSORT, see *z/OS DFSORT Application Programming Guide* or the DFSORT Home Page (www.ibm.com/storage/dfsort).
- For more information on RACF, see *z/OS Security Server RACF Command Language Reference*.

NaviQuest Tool for Batch

You can access the NaviQuest tool by using the ISMF Primary Option Menu. Use NaviQuest to migrate to SMS and maintain your SMS configuration.

With NaviQuest you can perform the following tasks:

- Automatically create test cases.
- Run ISMF in batch mode using REXX EXECs.
- Create reports using ISMF saved lists.
- Create reports with DCOLLECT and volume mount analyzer data.
- Use ACS code fragments as models when creating your own ACS routines.
- Create, modify, list, display or delete SMS configurations.
- Add or delete volumes to storage groups.

Related reading: For more information on using NaviQuest, see *z/OS DFSMSdfp Storage Administration*.

SMS Configurations

SMS manages DASD, tape, and optical disk storage according to the information in the currently active configuration. An SMS configuration includes the following items:

- Base configuration
- SMS classes and groups
- ACS routines
- Tape library definitions
- Optical library and drive definitions

Related reading: For more information, see *z/OS DFSMSdfp Storage Administration* and *z/OS DFSMS Implementing System-Managed Storage*.

Base Configuration

The *base configuration* identifies the systems that the SMS configuration manages. The base configuration also contains installation defaults, such as a default management class and default device geometry.

Source Control Data Set

The *source control data set* (SCDS) contains the information that comprises an SMS configuration. You can define multiple SCDSs to describe different configurations. You can use ISMF to define, alter, and display the contents of an SCDS, or to activate a new SMS configuration.

Active Control Data Set

The *active control data set* (ACDS) controls the storage management policy for the installation. When you activate SMS, the system copies information in the SCDS into the ACDS. You can activate an SMS configuration manually, or automatically at IPL.

SMS Classes and Groups

SMS classes and groups are lists of characteristics that are associated with or assigned to data sets, objects, and volumes. Automatic class selection (ACS) routines assign classes to data, based on its requirements and attributes, and select target storage groups. An SMS configuration can contain the following types of classes and groups:

Data Class
Data definition parameters

Storage Class
Availability and accessibility requirements

Management Class
Data migration, backup, and retention attributes

Storage Group
List of storage volumes with common properties

Aggregate Group
Backup or recovery of all data sets in a group in a single operation

Copy Pool
A pool of storage groups to be processed collectively for fast replication operations

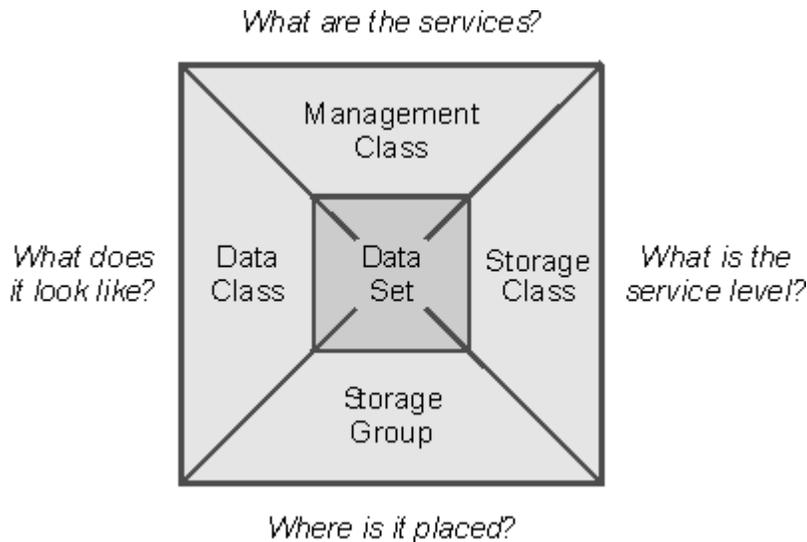shows the SMS classes and groups that are assigned to data.

Figure 2: SMS and its relationship to a data set

The following example shows the different ways in which you can use SMS services. The storage administrator might define one storage class for DASD data sets requiring high performance and another for DASD data sets requiring standard performance. The administrator might define one storage class for objects requiring fast initial access response, which would place the objects on DASD, or another with lower initial access response, which would place objects on tape or optical media.

SMS automatically assigns new data to classes based on criteria that the storage administrator selects, such as data naming conventions. SMS assigns the data to an appropriate storage group of devices. SMS uses the values from the assigned classes to determine the space management, performance, and availability requirements of the data.

Related reading: For further information about SMS classes, groups, and ACS routines, see the following documents:

• *z/OS DFSMSdfp Storage Administration*
• *z/OS DFSMS Implementing System-Managed Storage*

Data Class

A *data class* is a collection of allocation and space attributes that the system uses when creating a data set. If SMS is active, a data class can be assigned to any data set. For non-SMS-managed DASD data sets, the system uses the allocation attribute values of the data class, but it does not save the data class name. If you alter a data class definition, SMS applies the changes to all new data sets and certain existing data sets that use the data class.

For example, you can specify the record organization and space allocation values for a data set. You can specify compaction for tape data sets to make more efficient use of your tape cartridges.

You can associate the following types of data class attributes with a data class:

• Record, key, and space attributes
• Volume attributes
• Data set attributes
• VSAM attributes
• Backup-while-open attributes
• Tape media attributes

  – compaction
  – media type

- recording technology
- performance scaling.

Related reading: For more information about defining data classes, see the *z/OS DFSMSdfp Storage Administration*.

Storage Class

A *storage class* is a collection of performance goals and device availability requirements that the storage administrator defines. Storage class is a required part of an SMS-managed environment. A storage class provides the criteria that SMS uses in determining an appropriate location to place a data set or object. SMS uses the storage class to select a device according to the following criteria: device performance, the amount of space available on the volume, and how available a data set or object can be on that device.

The RAMAC Virtual Array, the 3990 Storage Control, and the 2105 Enterprise Storage Server use storage class availability and accessibility attributes to provide a required level of service. SMS ensures that a data set associated with that storage class is allocated to the required hardware.

The storage class coupling facility (CF) caching attributes work with the CF to provide record-level sharing (RLS) for VSAM data sets in a Parallel Sysplex. When you define a storage class with a CF cache set name, data sets are cached in specific CF cache structures, or buffer pools.

You can associate the following types of attributes with a storage class:

- Performance objectives for direct and sequential access
- Sustained data rate
- Availability and accessibility objectives
- Multi-tiered storage group
- Coupling facility caching attributes.

Related reading: For more information on record-level sharing for VSAM data sets, see "VSAM Record-Level Sharing" on page 29. For more information about defining storage classes, see the *z/OS DFSMSdfp Storage Administration*.

Management Class

A *management class* is a collection of management attributes that the storage administrator defines. For data sets, these attributes control retention, migration, backup, and release of allocated but unused space. For OAM objects, the attributes control retention, backup, and class transition. You can also use management classes to define how your aggregate groups should be managed. DFSMShsm acts on many management class attributes.

Management class is optional for SMS-managed data sets and does not apply to non-SMS-managed data sets.

You can associate the following types of attributes with a management class:

- Space management attributes
- Expiration and retention attributes
- Migration attributes
- Generation data group attributes
- Backup attributes
- Object class transition attributes
- Aggregate backup attributes.

Related reading: For more information on how DFSMShsm implements your management class policies, see "Managing Space" on page 56 and "Managing Availability" on page 61. For more information about defining management classes, see the *z/OS DFSMSdfp Storage Administration*.

Storage Group

A *Storage group* represents the physical storage that SMS manages. This storage can be collections of DASD volumes, volumes in tape libraries or in optical libraries, or virtual input/output (VIO) storage. The storage administrator groups the volumes to meet a specific service or business strategy.

The storage administrator uses the storage group ACS routine to assign a new data set or object to a storage group.

You can associate the following types of attributes with a storage group:

- Pool storage group attributes (allocation, backup, dump, migrate, extend, overflow, volume, status)
- VIO storage group attributes (size, unit, status)
- Dummy storage group attributes
- Object storage group attributes (library, cycle, volume)
- Object backup storage group attributes (library, cycle, volume)
- Tape storage group attributes (library, status)
- Copy pool backup storage group attributes.

Related reading: For more information about defining storage groups, see the *z/OS DFSMSdfp Storage Administration* and *z/OS DFSMShsm Storage Administration*.

Aggregate Groups for Disaster Backup and Recovery

The storage administrator defines aggregate groups of data sets to prepare for disaster recovery, application transfer, application archiving, or data migration among new business partners. An *aggregate group* is a collection of user-defined data sets that are processed together such as the collection of data sets used for a payroll program. DFSMShsm can back up and recover aggregate groups. A storage administrator might group related data sets in this manner:

- must be processed as a group
- are required to run a user application
- can be used for disaster recovery
- are being transferred from one site to another

Use the aggregate backup and recovery support (ABARS) tool to identify the critical applications in your enterprise and back up the applications as an entity.

Related reading: For more information on managing your aggregate groups with DFSMShsm, see "Protecting Critical User Applications" on page 61 and *z/OS DFSMShsm Storage Administration*.

Copy Pools

A *copy pool* is a defined set of storage groups that contain data that DFSMShsm can back up and recover collectively using fast replication. A copy pool backup storage group contains eligible target volumes that DFSMShsm can select for fast replication backup versions. After you define a copy pool and create a copy pool backup storage group, you can use DFSMShsm commands to prepare, create, recover, and delete fast replication backup versions. For more information, see "Using Fast Replication for Backup Versions" on page 66.

Related reading:

- For more information about using copy pools, see *z/OS DFSMSdfp Storage Administration*.
- For more information about ways to copy a volume, see *z/OS DFSMShsm Storage Administration* and *z/OS DFSMSdfp Storage Administration*.

Automatic Class Selection Routines

ACS routines automatically determine SMS classes and storage groups for all new data set allocations and for data set allocations that occur from converting, copying, recalling, restoring, or moving data sets.

ACS routines determine the storage group when storing objects; or storage class and management class, when storing or changing objects.

The storage administrator writes ACS routines in the ACS programming language, one routine for each type of SMS class and one for your storage groups. These four routines, used together with the data class, storage class, management class, and storage group definitions, and the base configuration, define the SMS configuration for your site. The storage administrator stores the information on this configuration in a source control data set (SCDS). ACS routines do not apply to copy pools or aggregate groups. For more information, see "SMS Configurations" on page 18.

After writing the ACS routines, the storage administrator must translate them into an object form that SMS understands. A successful translation places the ACS object in a specified SCDS. After you activate the configuration contained in that SCDS, all new data set allocations use the ACS routines. You can test ACS routines in batch through NaviQuest or ISMF. For more information, see "NaviQuest Tool for Batch" on page 17.

Figure 3 on page 22 shows a portion of a sample storage group ACS routine used to assign large data sets to the LARGE storage group and assign all other data sets to the PRIMARY storage group. Individual installations can determine their own standards for classifying data sets according to size. In Figure 3 on page 22, data sets greater than 200 000 KB are classified as large data sets; data sets smaller than those are placed in the primary storage group.
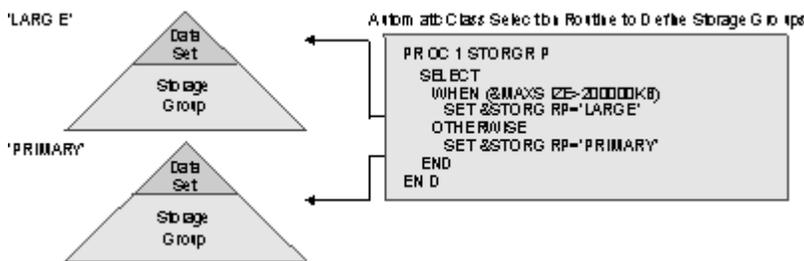

Figure 3: Example of a storage group automatic class selection routine

For example, you can implement the tape mount management methodology, using ACS routines to redirect small tape allocations to a pool storage group on DASD. DFSMShsm can automatically move the data sets together to tape at a later time and compact them. This methodology improves tape usage and significantly reduces tape mounts.

Related reading: For more information on using ACS routines, see *z/OS DFSMSdfp Storage Administration*.

Using SMS in a Multisystem Environment

You can use system-managed storage in a multisystem environment in which the z/OS systems share a common SMS configuration. Shared SMS control data sets contain a common set of classes and storage groups, ACS routines, and a base configuration that you can use across the systems. You can use DFSMShsm and DFSMSdss functions, such as restore, recover, and recall, in a multisystem environment.

You can use SMS in the following types of multisystem environments:

sysplex
A sysplex is a multiple z/OS-system environment that allows multiple-console support consoles to receive console messages and send operator commands across the system. These systems simultaneously process different types of work. For example, you could define different storage group

and volume connectivity for systems handling online transaction processing, and for systems handling batch processing.

Parallel Sysplex
A Parallel Sysplex consists of multiple z/OS images coupled together by hardware elements and software services. When multiple z/OS images are coupled using the cross-system coupling facility (XCF), they can be viewed as a single entity. XCF services allow applications on one system to communicate with applications on the same system or on other systems. The systems in a Parallel Sysplex can simultaneously process one type of work. In a Parallel Sysplex, you can share data, SMS control data sets, catalogs, and other system resources between the applications on these systems. For example, you can use VSAM record-level sharing on a Parallel Sysplex.

You can specify up to 32 unique system names, system group names, or a combination of both in your SMS configuration. DFSMS supports the following configurations:

- More than eight systems in a JES3 SMS complex, where system group names cannot be used
- System-level unique operations, such as VARY SMS commands, for more than eight unique systems, system groups, or a combination of unique systems and system groups
- More than eight unique system variations in connectivity, status and workload.

Requirement: You must have global resource serialization (GRS) or an equivalent product to serialize system resources on the multiple systems.

Related reading: For more information on the preceding topics, see:

- "VSAM Record-Level Sharing" on page 29
- *z/OS MVS Setting Up a Sysplex*
- *z/OS MVS Planning: Global Resource Serialization*

Chapter 4. Managing Data with DFSMSdfp

DFSMSdfp organizes, identifies, stores, catalogs, shares, and retrieves all the data that your installation uses. You can store data on DASD, magnetic tape volumes, or optical volumes. Using data management you can complete the following tasks:

- Allocate space on DASD and optical volumes.
- Automatically locate cataloged data sets.
- Control access to data.
- Transfer data between the application program and the medium.
- Mount magnetic tape volumes in the drive.

This chapter describes the different types of data sets and catalogs that DFSMS uses. Chapter 5, "Using Access Methods, Commands, and Utilities," on page 31 describes the access methods, macros, and commands that you can use to process data.

DFSMSdfp Data Set Organizations

The data organization that you choose depends on your applications and the operating environment. z/OS allows you to use temporary data sets and several ways to organize files for data to be stored on permanent media.

- VSAM data sets
- Non-VSAM data sets
- Extended-format data sets
- Objects
- z/OS UNIX files

VSAM Data Sets

VSAM data sets are formatted differently than non-VSAM data sets. Except for linear data sets, VSAM data sets are collections of records, grouped into control intervals. The control interval is a fixed area of storage space in which VSAM stores records. The control intervals are grouped into contiguous areas of storage called control areas. To access VSAM data sets, use the VSAM access method.

VSAM arranges records by an index key, by a relative byte address, or by a relative record number. VSAM data sets are cataloged for easy retrieval. The following are the different types of VSAM data sets:

Key-sequenced data set (KSDS)
Contains records in order by a key field and can be accessed by the key or by a relative byte address. The key contains a unique value, such as an employee number or part number.

Entry-sequenced data set (ESDS)
Contains records in the order in which they were entered and can only be accessed by relative byte address. An ESDS is similar to a sequential data set.

Relative-record data set (RRDS)
Contains records in order by relative-record number and can only be accessed by this number. Relative records can be fixed-length or variable-length.

Linear data set (LDS)
Contains data that can be accessed as byte-addressable strings in virtual storage. A linear data set does not have imbedded control information that other VSAM data sets hold. You also can access linear data sets by using the z/OS data in virtual (DIV) access method or window services. VSAM record-level sharing (RLS) is not available for VSAM linear data sets.

Related reading: For more information on VSAM data sets, see *z/OS DFSMS Using Data Sets*. For more information on DIV and window services, see *z/OS MVS Programming: Assembler Services Guide*.

Non-VSAM Data Sets

Non-VSAM data sets are collections of fixed-length or variable-length records, grouped into blocks. To access non-VSAM data sets, use BSAM, QSAM, or BPAM. There are several types of non-VSAM data sets:

Sequential data set (PS)
Contains records that are stored in physical order. New records are appended to the end of the data set. You can specify a sequential data set in extended format.

Partitioned data set (PDS)
Contains a directory of sequentially organized members, each of which can contain a program or data. After opening the data set, you can retrieve any individual member without searching the entire data set.

Partitioned data set extended (PDSE)
Contains an indexed, expandable directory of sequentially organized members, each of which can contain a program or data. You can use a PDSE instead of a PDS. The main advantage of using a PDSE over a PDS is that a PDSE automatically reuses space within the data set.

Related reading: For more information on non-VSAM data sets, see *z/OS DFSMS Using Data Sets*.

Extended-Format Data Sets

You can allocate both sequential and VSAM data sets in extended format on a system-managed DASD. The DASD is attached to a controller that supports Extended Platform.

An extended-format data set supports the following formats:

- Compression, which reduces the space for storing data and improves I/O, caching, and buffering performance.
- Data striping, which distributes data for one data set across multiple SMS-managed DASD volumes, which improves I/O performance and reduces the batch window. For example, a data set with 28 stripes is distributed across 28 volumes.
- Extended-addressability, which enables you to create a VSAM data set that is larger than 4 GB.
- Encryption, which enables you to protect sensitive data. A data set is encrypted when it is first allocated with an encryption key label.

  When a data set key label is specified through more than one interface, the order of precedence is as follows:

  1. RACF data set profile
  2. JCL, dynamic allocation, TSO ALLOCATE, IDCAMS DEFINE
  3. SMS data class

Extended-format VSAM data sets allow you to release partial space and to use system-managed buffering for VSAM batch programs. You can select whether to use the primary or secondary space amount when extending VSAM data sets to multiple volumes.

Related reading: For more information on extended-format data sets, see *z/OS DFSMS Using Data Sets*.

Large Format Data Sets

Large format data sets are sequential data sets that can grow beyond the size limit of 65 535 tracks (4369 cylinders) per volume that applies to other sequential data sets. Large format data sets can be system-managed or not. They can be accessed using QSAM, BSAM or EXCP.

Large format data sets reduce the need to use multiple volumes for single data sets, especially very large ones like spool data sets, dumps, logs, and traces. Unlike extended-format data sets, which also support greater than 65 535 tracks per volume, large format data sets are compatible with EXCP and don't need to be SMS-managed.

You can allocate a large format data set using the DSNTYPE=LARGE parameter on the DD statement, dynamic allocation (SVC 99), TSO/E ALLOCATE, or the access method services ALLOCATE command.

Related reading: For more information on large format data sets, see *z/OS DFSMS Using Data Sets*.

Basic format data sets

Basic format data sets are sequential data sets that are specified as neither extended-format nor large-format. Basic format data sets have a size limit of 65 535 tracks (4369 cylinders) per volume. Basic format data sets can be system-managed or not. They can be accessed using QSAM, BSAM or EXCP.

You can allocate a basic format data set using the DSNTYPE=BASIC parameter on the DD statement, dynamic allocation (SVC 99), TSO/E ALLOCATE or the access method services ALLOCATE command, or the data class. If no DSNTYPE value is specified from any of these sources, then its default is BASIC.

Related reading: For more information on basic format data sets, see *z/OS DFSMS Using Data Sets*.

Objects

Objects are named streams of bytes that have no specific format or record orientation. Use the object access method (OAM) to store, access, and manage object data. You can use any type of data in an object because OAM does not recognize the content, format, or structure of the data. For example, an object can be a scanned image of a document, an engineering drawing, or a digital video. OAM objects are stored either on DASD in a DB2® database or in a UNIX file system (zFS or NFS), or on an optical or tape storage volume.

The storage administrator assigns objects to object storage groups and object backup storage groups. The object storage groups direct the objects to specific disk, optical, or tape locations, depending on their performance requirements. You can have one primary copy of an object and up to two backup copies of an object.

A Parallel Sysplex allows you to access objects from all instances of OAM and from optical hardware within the sysplex. See "Optical and Tape Volume Support" on page 40 for further information on support for objects stored on optical and tape volumes.

Related reading: For more information on objects, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

z/OS UNIX Files

z/OS UNIX System Services (z/OS UNIX) enables z/OS to access UNIX files. UNIX applications also can access z/OS data sets. You can use the hierarchical file system (HFS), Network File System (NFS), z/OS File System (zFS), and temporary file system (TFS) with z/OS UNIX. You can use the BSAM, QSAM, BPAM, and VSAM access methods to access data in UNIX files and directories. The object access method (OAM) can also store object data in z/OS UNIX files in a zFS or NFS file system. z/OS UNIX files are byte oriented, similar to objects.

Hierarchical file system
You can define an HFS data set on the z/OS system. Each HFS data set contains a hierarchical file system. Each hierarchical file system is structured like a tree with subtrees, which consists of directories and all its related files. HFS data sets must reside on DASD volumes.

Network File System
An NFS is a distributed file system that enables users to access UNIX files and directories that are located on remote computers as if they were local. NFS is independent of machine types, operating systems, and network architectures.

z/OS File System
A zFS is a UNIX file system that supports multiple file systems in a data set.

Temporary file system
A TFS is stored in memory and delivers high-speed I/O. A system programmer can use a TFS for storing temporary files.

Related reading: For more information on z/OS UNIX files and HFS data sets, refer to the following publications:

- *z/OS UNIX System Services User's Guide*
- *z/OS DFSMS Using Data Sets*
- *z/OS DFSMS Macro Instructions for Data Sets*

Virtual Input/Output Data Sets

You can manage temporary data sets with a function called virtual input/output (VIO). VIO uses DASD space and system I/O more efficiently than other temporary data sets.

You can use the BPAM, BSAM, QSAM, BDAM, and EXCP access methods with VIO data sets. SMS can direct SMS-managed temporary data sets to VIO storage groups.

Related reading: For more information on VIO data sets, see *z/OS DFSMS Using Data Sets*.

Catalogs

To manage the storage and placement of data sets, DFSMSdfp uses catalogs and volume tables of contents (VTOCs). A catalog describes data set attributes and the location of a data set. You can retrieve the data set without having to specify the location of the data set. Cataloging data sets allows you to move data sets from one device to another without having to change the JCL. For maximum reliability and efficiency, you should catalog all permanent data sets.

You can use the catalog search interface to access the catalog. You can use the access method services utility to create, maintain, back up, and recover catalogs.

Master catalog
A single master catalog contains entries for system data sets and user catalogs. DFSMS uses the master catalog to locate data sets on volumes. All systems in a sysplex can use the same master catalog.

You can use the coupling facility to share catalogs in a sysplex with the use of enhanced catalog sharing. You can use system symbols in an alias entry, which permits each member of the sysplex to resolve the alias to a different data set name or catalog.

User catalogs
Multiple user catalogs contain information about user data sets. To simplify catalog selection, you can define aliases of up to four qualifiers for catalogs.

Volume catalogs
System-managed tape libraries and the volumes they contain must be cataloged in a volume catalog. A volume catalog (VOLCAT) is a user catalog that contains entries for tape volumes and tape libraries.

There are two categories of volume catalogs: general and specific. Each system can have access to only one *general volume catalog,* but can have access to many *specific volume catalogs*. A general volume catalog contains all the entries that define tape libraries as well as entries for tape volumes that are not cataloged in a specific volume catalog. Specific volume catalogs contain a specific group of volume entries based on the tape volume serial number.

VSAM volume data set
A VSAM volume data set (VVDS) resides on every volume that contains a cataloged VSAM data set or any other SMS-managed data set. The VVDS contains information about data sets on the volume. For more information, see *z/OS DFSMS Using Data Sets*.

Related reading: For more information, see *z/OS DFSMS Managing Catalogs* and *z/OS DFSMS Access Method Services Commands*.

Volume Table of Contents

The volume table of contents (VTOC) is a data set that describes the contents of the DASD volume on which it resides. The VTOC locates data sets on that volume. The VTOC is composed of data set control blocks (DSCBs). The DSCBs describe the type and location of data sets on that volume, unused records in the VTOC, and the unallocated space on the volume.

A VTOC index decreases search time for DSCBs. If the system detects a logical or physical error in a VTOC index, the system disables further access to the index from all systems that might be sharing the volume. If a VTOC index becomes disabled, the VTOC remains usable but with possibly degraded performance.

If a VTOC index becomes disabled, you can rebuild the index without taking the volume offline to any system. All systems can continue to use that volume without interruption to other applications, except for a brief pause during the index rebuild. After the system rebuilds the VTOC index, it automatically reenables the index on each system that has access to it.

Related reading: For more information, see *z/OS DFSMSdfp Advanced Services* and *Device Support Facilities (ICKDSF) User's Guide and Reference*.

Data Set Sharing

DFSMS allows sharing of VSAM and non-VSAM data sets within different jobs in a single operating system or within different operating systems. To share between different operating systems, you need global resource serialization or an equivalent product.

Enhanced data integrity prevents accidental data loss when concurrent users are writing to a shared sequential data set.

Related reading: For more information, see *z/OS DFSMS Using Data Sets* and *z/OS MVS Planning: Global Resource Serialization*.

PDSE Sharing Among z/OS Systems

DFSMS allows users on different z/OS systems to access the same PDSE simultaneously. A user who is sharing a PDSE can read or copy existing members in the PDSE, or create new members concurrently with other users on the same or other z/OS system and on other z/OS systems. Data integrity is maintained because only one user at a time can modify a data set member without copying it. With a PDS, you cannot copy or create members concurrently with other users of the PDS.

Related reading: For more information, see *z/OS DFSMS Using Data Sets*.

Hiperbatch for QSAM and VSAM

Hiperbatch provides an alternate method of making data available to many batch jobs. Hiperbatch can significantly reduce the execution time of batch job streams that access QSAM or VSAM data sets.

Related reading: For more information on Hiperbatch, see *z/OS DFSMS Using Data Sets* and *MVS Hiperbatch Guide*.

VSAM Hiperspaces

When you use the local shared resources facility to obtain VSAM buffers, VSAM Hiperspace™ provides a high performance method for accessing expanded storage.

Related reading: For more information, see *z/OS DFSMS Using Data Sets*.

VSAM Record-Level Sharing

VSAM record-level sharing (RLS) enables data sharing across multiple systems in a Parallel Sysplex. VSAM RLS is a data set access mode that allows CICS, batch, and non-CICS applications that run in multiple address spaces and on multiple z/OS systems to access data at the same time. VSAM RLS is designed for

online transaction processing (OLTP) with CICS Transaction Server and applications that use VSAM data sets.

With VSAM RLS, multiple CICS systems can directly access a shared VSAM data set. CICS provides the logging, commit, and rollback functions for VSAM recoverable files. VSAM uses the coupling facility (CF) to provide record-level serialization and cross-system caching. IBM CICS VSAM Recovery (CICSVR) recovers lost or damaged VSAM data sets.

CICSVR is for organizations where the availability and integrity of VSAM data is vital. CICSVR provides automated complete recovery, forward recovery, and backout functions, as well as logging for batch applications.

You can use VSAM RLS with DFSMStvs to run concurrent batch VSAM jobs and CICS online transactions against the same recoverable VSAM data set. DFSMStvs also provides forward recovery logging for recoverable data sets. If data is lost or damaged, you can restore it from a backup and use CICSVR to reapply the changes that were made since the last backup.

Related reading:

- For more information on VSAM record-level sharing, see *z/OS DFSMS Using Data Sets*.
- For more information on defining coupling facility resource management (CFRM) policies, see *z/OS DFSMSdfp Storage Administration*.
- For more information on CICSVR, see *IBM CICS VSAM Recovery Implementation Guide*.
- *z/OS DFSMStvs Planning and Operating Guide*

Chapter 5. Using Access Methods, Commands, and Utilities

DFSMSdfp manages the organization and storage of data in the z/OS environment. You can use access methods with macro instructions to organize and process a data set or object. Access method services commands manage data sets, volumes, and catalogs. Utilities perform tasks such as copying or moving data. You can use system commands to display and set SMS configuration parameters, DFSMSdfp callable services to write advanced application programs, and installation exits to customize DFSMS.

This chapter provides an overview of DFSMSdfp access methods, commands, utilities, installation exits, and callable services.

DFSMSdfp Access Methods

DFSMSdfp provides several access methods for formatting and accessing data. An access method defines the organization of the data in a data set and the technique by which the data is stored and retrieved. DFSMSdfp access methods have their own data set structures to organize data, macro instructions to process data sets, and utility programs to manipulate data sets.

Table 1 on page 31 describes the access methods that DFSMSdfp uses.

Table 1: DFSMSdfp access methods

Access method	Description	Data set organization
Basic partitioned access method (BPAM)	Use BPAM to create and retrieve program and data libraries on DASD. BPAM arranges records as members of PDSs, PDSEs, or z/OS UNIX directories.	• PDS • PDSE • z/OS UNIX
Basic sequential access method (BSAM)	Use BSAM to process data sets sequentially. You organize the records into blocks for retrieval.	• Sequential data sets • Extended-format data sets • PDS members • PDSE members • z/OS UNIX files
Object access method (OAM)— OSREQ interface	Use OAM to store, back up, and retrieve objects on disk, optical, and tape storage.	• Objects
Queued sequential access method (QSAM)	Use QSAM to process data sets sequentially. QSAM collects the records into blocks.	• Sequential data sets • Extended-format data sets • PDS members • PDSE members • z/OS UNIX files
Virtual storage access method (VSAM)	Use VSAM for direct or sequential processing of records on DASD. VSAM arranges records by an index key, by relative byte address, or by relative record number. VSAM catalogs data sets for easy retrieval.	• Entry-sequenced data sets • Key-sequenced data sets • Linear data sets • Relative record data sets • HFS files

DFSMS also supports the basic direct access method (BDAM) for coexistence with previous operating systems.

Related reading:

- For more information on BPAM, BSAM, QSAM, VSAM, and BDAM, see *z/OS DFSMS Using Data Sets*.
- For more information on OAM objects, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Application Programmer's Reference*.

Data Management Macro Instructions

You can use assembler language macro instructions to create, maintain, and process all the data set types supported by the access methods described in "DFSMSdfp Access Methods" on page 31. Macro instructions control data set allocation, input and output, and data security.

Each compiler provides facilities to create, read, and write data sets. Your compiler documentation describes how to use the access method facilities.

The following are a few of the functions that the macro instructions perform:

- Control block macros generate information that the access method needs to process the data sets.
- Request macros retrieve, update, delete, or insert logical records into data sets.
- Checkpoint/restart functions establish checkpoints during a program and restart the job at a checkpoint or at the beginning of a job step.

Related reading:

- For more information about DFSMSdfp macro instructions, see the following publications:
  - *z/OS DFSMS Macro Instructions for Data Sets*
  - *z/OS DFSMSdfp Advanced Services*
- For more information about OAM macro instructions, see the following publications:
  - *z/OS DFSMS OAM Application Programmer's Reference*
  - *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*
- For more information about checkpoints, see *z/OS DFSMSdfp Checkpoint/Restart*.

Access Method Services Commands

Access method services, also known as IDCAMS, creates and maintains VSAM data sets. With access method services, you can perform the following tasks:

- Define VSAM data sets.
- Define and build alternate indexes.
- Back up and restore VSAM data sets.
- Copy data sets.
- Print the contents of data sets.
- Delete data sets.
- Collect information about data sets.
- Examine the structural consistency of VSAM key-sequenced data sets.
- Control DASD cache.
- List tape volume (VOLCAT) catalog entries.
- Diagnose catalog errors.
- Recover from catalog errors.

- Define system-managed libraries and volumes.
- Define extended addressability for an extended-format VSAM data set to support a data set size greater than 4 GB.
- Encrypt and decrypt data sets.

You also can define VSAM data sets using JCL or dynamic allocation macros.

Related reading: For more information, see *z/OS DFSMS Access Method Services Commands*.

MVS System Commands

The MVS system console commands control the z/OS operating system. You can use MVS system commands to display device status and channel paths, and display or change the SMS configuration.

Related reading: For more information, see *z/OS MVS System Commands*. For the new and changed system commands, see *z/OS DFSMS Using the New Functions*.

Data Management Utilities

You can use the following data management utility programs to perform a variety of tasks, such as moving or copying data.

IEBCOMPR
Compares data sets to verify the accuracy of backup copies.

IEBCOPY
Copies and merges PDSs and PDSEs, or loads and unloads these data sets to a sequential data set.

IEBDG
Creates a pattern of test data for debugging programs.

IEBEDIT
Edits jobs and job steps into a single output data set.

IEBGENER
Copies records from a sequential data set or converts sequential data sets into members of PDSs or PDSEs.

IEBPTPCH
Punches or prints records from data sets.

IEBUPDTE
Updates PDSs with fixed-length records.

IEHINITT
Labels tapes in stand-alone devices and automated tape libraries.

IEHLIST
Lists entries in catalogs, PDS or PDSE directories, or volume tables of contents (VTOCs).

IFHSTATR
Formats and prints information about tape volume errors from system management facilities (SMF) records.

Related reading: For more information on the utilities, see *z/OS DFSMSdfp Utilities*.

DFSMSdfp Callable Services

User programs written in assembler language and in high-level languages can call the DFSMSdfp callable services.

IGWABWO
 Retrieves or sets data set indicators. For example, your program can determine if a data set can be backed up while it is open for update.

IGWARLS
 Obtains information about the record-level sharing attributes for a VSAM data set.

IGWASMS
 Determines if a data set is system-managed, and returns the SMS class names and data set type.

IGWASYS
 Determines the version, release, and modification level of DFSMS and the status of the SMS subsystem.

IGWLSHR
 Determines the DFSMSdfp share attributes currently in use on the system.

Related reading: For more information, see *z/OS DFSMSdfp Advanced Services*.

Installation Exits

You can customize DFSMS with installation exit routines and modules that extend or replace IBM-supplied functions at your installation. DFSMS provides the following types of installation exits:

- User installation exits
- Tape label processing installation exits
- System-managed tape library installation exits
- Pre-ACS installation exit for tape management systems
- ACS installation exits that extend the capabilities of ACS routines
- IEHINITT dynamic exits that place volume labels onto magnetic tapes
- DFSMShsm installation exits for customizing DFSMShsm and ABARS functions
- DFSMSdss installation exits

Related reading: For more information, see *z/OS DFSMS Installation Exits*, *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*, and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

System Data Administration

You can use special macros to modify the data management capabilities of the operating system:

Direct Access Device Space Management (DADSM)
 DADSM provides exit routines that control space allocation on DASD volumes. However, you can improve storage management by using ACS routines instead of the exits.

Execute Channel Program (EXCP)
 You use the EXCP access method to establish your own system for organizing, storing, and retrieving data based on device characteristics.

Related reading: For more information on DADSM exit routines and EXCP macro instructions, see *z/OS DFSMSdfp Advanced Services*.

Data Security

Security is the ability to protect data processing resources from unauthorized access, alteration, or destruction. DFSMSdfp supports several methods of controlling access to sensitive data.

Resource Access Control Facility (RACF)
RACF, a component of the z/OS Security Server, protects the system by giving access to those individuals who have authority to use the resource. You can use RACF to protect data sets, authorize access to ISMF functions, and authorize use of DFSMShsm, DFSMSdss, and DFSMSrmm commands.

Authorized Program Facility (APF)
APF is a facility that permits the identification of programs that are authorized to use restricted functions.

Integrated Cryptographic Service Facility (ICSF)
ICSF provides access to the hardware cryptographic feature for programming applications. The combination of the hardware cryptographic feature and ICSF provides secure high-speed cryptographic services.

Related reading:

• For more information on RACF, see *z/OS Security Server RACF Security Administrator's Guide*.

• For more information on APF, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

• For more information on ICSF, see *z/OS Cryptographic Services ICSF Overview*.

Chapter 6. Managing Devices with DFSMSdfp

Use the Hardware Configuration Definition (HCD) to define I/O devices to the operating system and control these devices. DFSMS manages DASD, storage control units, magnetic tape devices, optical devices, and printers. You can use DFSMS functions to manage many different device types, but most functions apply specifically to one type or one family of devices.

DFSMS Device Setup

This section introduces the operating modes in which devices can operate and discusses one way to define new devices to DFSMS.

For a list of the most common IBM I/O devices supported by DFSMS, see *z/OS Planning for Installation*.

Device Operating Modes

Most devices attached to z/OS operate in *full-function* mode; that is, all features on the device are compatible with and usable on the operating system. Some of these features include:

- DASD devices:
  - Dynamic path reconnection
  - Extended count-key-data operation
  - Caching and cache related facilities
- Tape devices:
  - Cartridge stack loading
  - Data compaction

Some devices also operate in *compatibility* mode, which lets you simulate the function of another device or model. Compatibility mode causes the device to function like a different device of the same type, ignoring some or all of the additional features that the device might have. This allows you to migrate between devices with minimal impact on programs that have device dependencies.

Hardware Configuration Definition

Use the Hardware Configuration Definition (HCD) element of z/OS to dynamically define an I/O configuration for your system. You can define the following hardware resources with HCD:

- Devices
- Channels
- Storage control units
- ESCON and FICON®™ switches

The number of devices that you can attach to your system depends on the hardware configuration of your processor and I/O devices, and the amount of virtual storage.

Related reading: For information on the Hardware Configuration Definition, see *z/OS HCD Planning*.

Managing Devices

This section describes the different tools for managing DASD, storage control units, magnetic tape volumes, optical volumes and libraries, and printers.

DASD

Use the following functions to manage DASD devices:

Interactive Storage Management Facility
ISMF enables both storage administrators and end-users to perform tasks related to storage, data, space, object, and tape management in DFSMS and other programs. For more information, see "Interactive Storage Management Facility Menus" on page 17.

MVS system commands
MVS system commands display the status of DASD devices. You also can use these commands to activate SMS or check, display, and change the status of the SMS configuration. For more information, see *z/OS MVS System Commands*.

Storage Control Units

You can use DFSMS to manage cache devices.

Continuous availability
Using the continuous availability function, SMS allocates the data set to a dual copy device or to a RAID protected device (support by all RAMAC subsystems). (The 3990 Model 3 or 6 supports dual copy.) For more information, see *z/OS DFSMSdfp Storage Administration*.

Extended-format data sets
SMS allocates extended-format data sets to a system-managed DASD volume on a 3990 Model 3, Model 6, any RAMAC subsystem, or Enterprise Storage Server. The data can be allocated on a DASD that supports extended-format data sets. If a device supporting extended-format data sets is not available, the data can be allocated on another device. For more information, see *z/OS DFSMS Using Data Sets*.

Caching on the IBM 3990 Model 3 or 6
SMS can select the 3990 Model 3 or 6 to provide caching for both read and write operations. The DASD fast write feature of the 3990 Model 3 or 6 provides caching for write operations. Caching enables applications to retrieve data sets more quickly and increases DASD capacity by mixing cached and noncached data on the same volume.

You can use access method services or the ISMF volume application to enable caching and DASD fast write at the volume level.

Caching on the IBM 3880 Model 23
SMS can select the 3880 Model 23 to provide caching for read operations. Caching for a system-managed volume on a 3880 Model 23 is enabled at the volume level, using access method services or the ISMF Volume Application.

Caching on RAMAC subsystems
Caching is always active on the 9393 RAMAC Virtual Array Storage and 9396 RAMAC Scalable Array Storage (RSA). The 9397 RAMAC Electronic Array Storage (REA) is a cache-only subsystem.

Related reading: For information on cache devices, Enterprise Storage Server, and RAMAC Storage Subsystems, see the *3990 Planning, Installation, and Storage Administration Guide*.

Point-in-Time Copy Services

Concurrent copy, SnapShot, and FlashCopy are the point-in-time copy functions. The point-in-time copy functions allow you to back up or copy data concurrently with normal applications. For an overview of these copy functions, see "Point-in-Time Copy Functions" on page 12.

Related reading: For more information on using these copy services, see the following publications:

- *z/OS DFSMSdfp Storage Administration*
- *z/OS DFSMSdss Storage Administration*
- *z/OS DFSMShsm Storage Administration*
- *z/OS DFSMS Advanced Copy Services*

Dynamic Copy Services

Extended remote copy (XRC) and peer-to-peer remote copy (PPRC) are dynamic copy functions. The dynamic copy functions constantly update the secondary copy as applications make changes to the primary data source. If an unrecoverable error occurs at the primary location, you can recover data from the recovery storage system.

Related reading: For more information on using XRC and PPRC, see the following material:

- "Managing Your Data with Copy Services" on page 13
- *z/OS DFSMS Advanced Copy Services*

Magnetic Tape Volumes and Libraries

DFSMS provides the following functions for magnetic tape devices and tape libraries:

Software support
DFSMS supports many types of magnetic tape drives. You can use IBM tape drives in the stand-alone environment, an automated tape library, or manual tape library. Each type of magnetic tape cartridge has a unique media type and recording format. For more information, see the publication for your IBM tape drive.

Tape label support
DFSMS can process magnetic tapes with IBM standard labels, nonstandard labels, ISO/ANSI labels, and tapes with no labels. For more information on tape labels and ISO/ANSI standards for tapes, see *z/OS DFSMS Using Magnetic Tapes*.

IEHINITT utility
IEHINITT writes IBM standard or ISO/ANSI labels on tape volumes. For more information on IEHINITT, see *z/OS DFSMSdfp Utilities*.

EDGINERS utility
You can use the EDGINERS utility in DFSMSrmm to initialize your tape volumes instead of using IEHINITT. EDGINERS provides several functions that IEHINITT does not. For more information on EDGINERS, see *z/OS DFSMSrmm Implementation and Customization Guide*.

IFHSTATR utility
IFHSTATR formats and prints information about tape volume errors from SMF records. For more information on IFHSTATR, see *z/OS DFSMSdfp Utilities*.

MSGDISP, SYNCDEV, NOTE, and POINT macros
These macros control operation of the IBM magnetic tape subsystems that use tape cartridges. For more information on these macros, see *z/OS DFSMS Macro Instructions for Data Sets* and *z/OS DFSMSdfp Advanced Services*.

MVS system commands
MVS system commands display the status of tape devices. For example, the DEVSERV command tests all paths to a device and displays information about the logical status of each path on the system console. The operator can use this information to diagnose problems in the I/O subsystem. For more information on console commands, see *z/OS MVS System Commands*.

Tape block count checking
Block count checking compares the number of tape blocks calculated and detects discrepancies between them. Magnetic tape subsystems maintain block count information for both reading and writing to improve tape reliability. For more information about block count checking, see *z/OS DFSMS Using Magnetic Tapes*.

Interactive Storage Management Facility
You can use ISMF to define attributes for system-managed tape libraries and storage groups interactively. You can also use ISMF to create, save, and restore lists of tape volumes in libraries.

DFSMSrmm TSO/E subcommands
Use the DFSMSrmm TSO/E subcommands to manage removable media. For more information, see *z/OS DFSMSrmm Managing and Using Removable Media*.

Volume mount analyzer
The volume mount analyzer produces reports that show the mounts and tape usage at your installation. For more information, see *z/OS DFSMS Using the Volume Mount Analyzer*.

Virtual Tape Server
The virtual tape server (VTS) system combines the random access and high performance characteristics of DASD with outboard hierarchical storage management and virtual tape devices and volumes. The VTS provides significant reductions in the number of physical cartridges, devices, and automated libraries needed to store data. For more detailed information on VTSs, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Peer-to-Peer Virtual Tape Server
A Peer-to-Peer VTS (PtP VTS) automatically copies the data stored on newly-created or updated tape volumes in each of two interconnected VTSs. This dual-volume copy function improves data availability and data recovery, while being transparent to applications and host processor resources. For a more detailed description of the PtP VTS, see "Virtual Tape Servers, Devices, and Volumes" on page 9.

Outboard Policy Management
Outboard policy management functions use the existing SMS management constructs to effectively manage the VTS physical and logical tape volumes. You assign construct names using ACS routines. The VTS manages the tape volumes through the construct names that are passed to the library during mount processing. See "Outboard Policy Management" on page 10 for a more detailed description of outboard policy management.

For further information about system-managed tape libraries and tape drives, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Optical and Tape Volume Support

DFSMS provides the following functions to help you more easily manage data residing on optical and tape volumes:

Interactive Storage Management Facility
ISMF lets you define attributes for optical libraries and optical drives interactively.

MVS system commands
OAM and SMS allow you to display the following information:

- Object access method status (the number, type, and availability of defined optical drives in the active configuration)
- A summary of the name, type, and status of OAM control tasks, and information about object processing status
- Details about the number, type, and status of object processing for a specified control task
- Optical and tape library, optical drive status, and tape drive status (online, offline, connectivity to the system).

Object Access Method
The object access method (OAM) lets you create, access, and delete objects on optical, tape, or disk storage. You define SMS groups and classes to establish the following operating conditions:

- Object storage medium hierarchies and requests for a backup copy of an object
- Whether the object resides on optical, tape, or disk storage
- When to migrate an object from one storage medium to another
- The number of backup copies of an object assigned to an object or object backup storage group
- Object expiration sequences

Parallel Sysplex
A Parallel Sysplex allows you to link multiple systems together to communicate and share data through the cross-system coupling facility. XCF services allow authorized applications on one system to cooperate with and share data with applications on the same system or on other systems within the sysplex. Because a sysplex facilitates parallel processing on multiple systems, objects can be accessed from all instances of OAM and from optical hardware within the sysplex. This allows transactions to be processed more efficiently.

For further information about OAM object support for optical and tape libraries and drives, see:

z/OS DFSMS OAM Application Programmer's Reference
z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support

Printers

Although most printers require the Print Services Facility™, older printers use the IEBIMAGE utility and the IGGUCSIT and SETPRT macros.

For more information on:	See the following publication:
IEBIMAGE	*z/OS DFSMSdfp Utilities*
IGGUCSIT	*z/OS DFSMSdfp Advanced Services*
SETPRT	*z/OS DFSMS Macro Instructions for Data Sets*
PSF for z/OS	*Advanced Function Printing: Printer Summary*

Chapter 7. Distributed Data Access with DFSMSdfp

In the distributed computing environment, applications must often access data residing on different computers in a network. Often, the most effective data access services occur when applications can access remote data as if it were local data.

Distributed FileManager/MVS is a DFSMSdfp client/server product that enables remote clients in a network to access data on z/OS or OS/390® systems. Distributed FileManager/MVS provides workstations with access to z/OS data. Users and applications on heterogeneous client computers in your network can take advantage of system-managed storage on z/OS, data sharing, and data security with RACF.

You should understand the following terms used in this chapter:

Client/server
A distributed application design model in which the front-end transaction (the one that initiates the conversation) is called the *client* and controls the course of the conversation. The *server* receives a request from the client, processes it, and returns the results.

User
Any person or application that issues or receives commands and messages to or from the information processing system.

File
A collection of related data that is stored and retrieved by an assigned name. Data set in z/OS is equivalent to file in other environments.

Distributed FileManager/MVS

Distributed FileManager/MVS uses the Distributed Data Management (DDM) and Advanced Program to Program Communication (APPC) protocols. DDM enables like and unlike computer systems to share file systems across a network. APPC manages peer-to-peer network conversations between application programs.

Distributed FileManager/MVS enables your z/OS system to act as a server to remote client systems. Distributed FileManager/MVS is designed to work with operating system platforms that support DDM client requests.

With Distributed FileManager/MVS, you can remotely access data on z/OS systems from the following IBM systems, or their equivalent:

- Workstations running the OS/2 operating system. Distributed FileManager/MVS on the OS/2 is called DFM/2.
- AS/400 midrange computers running the OS/400® operating system. Distributed FileManager/MVS on the OS/400 is called Source OS/400 DDM.
- Computers running the AIX® operating system.

Related reading: For more information about DDM, see *DDM Architecture: General Information*.

shows the Distributed FileManager/MVS client/server relationship.
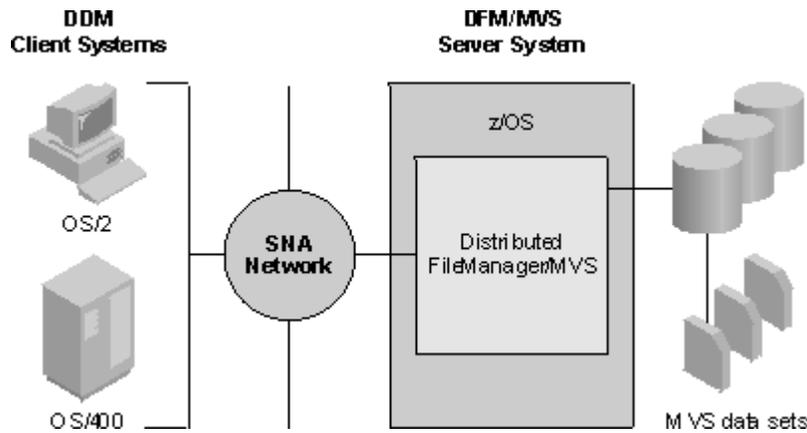
Figure 4: Distributed FileManager/MVS client/server relationship

Related reading: For more information on Distributed FileManager/MVS, see *z/OS DFSMS DFM Guide and Reference*.

Accessing z/OS Data with Distributed FileManager/MVS

Distributed FileManager/MVS enables users on authorized client systems to remotely access data on z/OS server systems as if the data were local to the users. Users can remotely perform such tasks as creating, reading, updating, and deleting z/OS data sets using local commands. Because DDM clients and servers communicate through common DDM commands, users on the client systems do not need to use z/OS command language.

Distributed FileManager/MVS allows client systems to access the following data set types on z/OS server systems:

- Physical sequential data sets
- VSAM data sets
- PDSE and PDS members

Using Distributed FileManager/MVS DataAgent

The Distributed FileManager/MVS Data Agent enables workstation users on Windows, OS/2, and AIX to invoke DataAgent routines. Through DataAgent routines, users can issue TSO/E commands, CLISTS, or Restructured Extended Executor (REXX) execs to access otherwise unsupported z/OS data sets or databases. For example, a DataAgent routine can extract data from z/OS files and databases, for the client to retrieve. DataAgent routines can be user-written, IBM-written, or vendor-written.

Chapter 8. Understanding DFSMSdss

DFSMSdss copies and moves data to help manage storage, data, and space more efficiently. You can use DFSMSdss to perform the following tasks:

Copy data and move data
DFSMSdss enables you to copy and move data between like and unlike devices and also copy backup data. See "Copying and Moving Data" on page 45 for more information.

Manage space
DFSMSdss helps you to manage DASD space efficiently. See "Managing Space" on page 47 for more information.

Back up and restore data
DFSMSdss provides host system backup and recovery functions. See "Backing Up and Restoring Data" on page 47 for more information.

DFSMSdss also includes a stand-alone restore program that restores data without a host operating system. See "DFSMSdss Stand-Alone Services" on page 49 for more information.

Convert data
DFSMSdss converts non-system-managed data to system-managed data and system-managed data to non-system-managed data. See "Converting to System-Managed Data" on page 49.

This chapter provides an overview of these DFSMSdss capabilities and describes additional DFSMSdss facilities that allow you to perform the following tasks:

- Choose between logical and physical processing
- Select data sets by filtering on criteria that you specify
- Invoke DFSMSdss from a variety of sources
- Control run-time options
- Print DASD data
- Control access to DFSMSdss tasks and data
- Customize DFSMSdss with installation exits

Copying and Moving Data

DFSMSdss is the primary data mover for DFSMS. It can efficiently move multiple data sets from old to new DASD. The data movement capability that is provided by DFSMSdss is useful for many other operations, as well.

DFSMSdss saves distributed data management (DDM) attributes that are associated with a specific data set and preserves those attributes during copy and move operations. DFSMSdss also offers the FlashCopy feature with Enterprise Storage Server (ESS) and the SnapShot feature with RAMAC Virtual Array (RVA). FlashCopy and SnapShot function automatically, work much faster than traditional data movement methods, and are well suited for handling large amounts of data.

When it is used with supporting hardware, DFSMSdss also provides concurrent copy capability. Concurrent copy lets you copy or back up data while that data is being used. The user or application program determines when to start the processing, and the data is copied as if no updates have occurred.

Copying and Moving Data to Meet Changing Storage Requirements

Moving data is an essential part of storage management. You must move data in order to replace storage devices, add storage capacity, and meet storage requirements. DFSMSdss has the following move capabilities:

- It moves data sets between DASD types and models.
- It moves data sets between system-managed and non-system-managed volumes.
- It moves data sets off of a volume when hardware maintenance is required.
- It moves or copies data sets for other purposes.

DFSMSdss can move data sets from one DASD volume to another volume of the same or different device type. Devices are of the same type if both their track capacities and the number of tracks per cylinder are the same. If the device types are the same, DFSMSdss can copy ranges of tracks from a volume or can copy a full volume.

DFSMSdss allows you to perform the following tasks:

- Move an entire VSAM sphere (the base cluster and all associated alternate index components and paths) with one copy. You need not move each component individually.
- Copy or move multivolume data sets.
- Retrieve readable data from undamaged portions of a volume that has sustained damage.
- Delete data sets from the source volume after a successful copy.
- Rename the copied data sets.
- Control data set placement on the target volume by copying data to preallocated data sets. In this way, you can place data sets on specific target volumes and tracks.
- Redefine the block size of one or more physical sequential, partitioned data sets (PDS), or partitioned data sets extended (PDSE) while copying those data sets.
- Copy a PDSE to a PDS, or a PDS to a PDSE.
- Copy data sets that have an undefined data set organization (DSORG) to an unlike device of larger capacity. DFSMSdss uses track-for-track replacement to copy or restore an undefined DSORG data set to a target device whose track size is greater than or equal to the source device.
- Move system data sets.
- Specify the target status of system-managed generation data sets.

You can move data sets between system-managed volumes and non-system-managed volumes. Moving data sets or copying data sets invokes the Automatic Class Selection (ACS) routines. These routines determine the Storage Management Subsystem (SMS) classes of the target data set. If you have RACF authorization, you can specify that ACS routines be bypassed. For more information, see "SMS Classes and Groups" on page 18 (ACS routines) and "Controlling Access to DFSMSdss Tasks and Data" on page 52 (RACF).

Moving Data with a Fast Replication Method

FlashCopy is much faster than traditional methods of data movement, especially for moving large amounts of data. DFSMS can use the FlashCopy feature of the ESS to quickly move the data from the source location to the target location. The source devices and the target devices must be in the same ESS, and the data to be moved must not need manipulation.

SnapShot can also be used to quickly move data from the source location to the target location. For additional information about SnapShot, see "Moving Data with SnapShot" on page 48.

Copying DFSMSdss-Produced Backup Data

You can make up to 255 copies of DFSMSdss-produced backup data. This is helpful when you need extra backup tapes for disaster recovery or for the distribution of dumped data.

You can copy the DFSMSdss-produced backup data, a sequential data set, to a tape or a DASD volume. The sequential data set that you are copying can reside on one or more tape volumes or DASD volumes. You can selectively copy the data from one or more of the DASD volumes when the backup data is produced from multiple DASD volumes.

Managing Space

Allocation algorithms and the frequent creation, extension, and deletion of data sets cause free space fragmentation on DASD volumes. The result is an inefficient use of DASD space, an increase in space-related abends (abnormal endings), and degraded performance.

With DFSMSdss, you can effectively manage space by reducing or eliminating free-space fragmentation. DFSMSdss provides features that help you perform the following space management tasks:

- Compress partitioned data sets.
- Release unused space in physical sequential, PDSs, PDSEs, and extended-format VSAM data sets.
- Consolidate free space on DASD volumes.

Compressing Partitioned Data Sets

You can selectively compress partitioned data sets on specified volumes to remove unused space between members in a partitioned data set. The recovered space is then available for reuse at the end of the data set. You can also use DFSMSdss to release this recovered space.

Releasing Unused Space

DFSMSdss releases allocated, but unused, space from sequential data sets, PDSs, PDSEs, and extended-format VSAM data sets using either volume or catalog filtering. DFSMSdss allows you to perform the following tasks:

- Release space only if the unused space is larger than the number of tracks that you specify.
- Release unused space in system data sets.
- Retain a specified secondary space allocation, leaving space in order to add records to the data set after DFSMSdss releases the unused space.

Consolidating DASD Space

DFSMSdss relocates data set extents on a DASD volume to reduce or eliminate free-space fragmentation, and reports about free space and other volume statistics. DFSMSdss allows you to perform the following tasks:

- Find the free space and volume statistics without defragmenting a volume.
- Specify that a volume be defragmented only if the space that is wasted by fragmentation exceeds a specified threshold.
- Use filtering to exclude selected data sets from defragmentation.

When protected data sets are relocated, DFSMSdss erases the data in the old location for security reasons.

Backing Up and Restoring Data

You can use DFSMSdss to back up data from DASD to tape, other DASD, or object storage clouds. You can also use DFSMSdss to restore data from the backup if the original is lost, damaged, or inadvertently changed. You can also use DFSMSdss to back up application data for vital records protection and disaster recovery. DFSMSdss backs up and restores data sets, specific tracks on a volume, or a full volume.

Backing Up Data

You can select data sets for backup by searching catalogs and the VTOC. You can back up data selectively, using either ISMF or DFSMSdss filtering to choose only the data sets that you want. For a description of DFSMSdss filtering, see .

DFSMSdss allows you to perform the following backup tasks:

- Back up an entire VSAM sphere by invoking one dump.
- Reset the data-set-changed flags of successfully dumped data sets in a full dump or in a data set dump. The next time you do a data set dump, you can specify that only data sets that have been updated since the last reset be dumped.
- Specify whether DFSMSdss continues or cancels a dump when a permanent read error occurs.
- Dump either all of the allocated space or only the used space in physical sequential and PDS, PDSEs, or data sets with no indicated data set organization. Users typically create the latter with the EXCP macro.
- Dump multivolume data sets.
- Dump and delete system data sets.
- Save space in the dump output data set by compressing the data. You can also optimize the data transfer rate by specifying the number of DASD tracks read at one time.
- Retrieve readable data from undamaged portions of a volume that has sustained partial damage.
- Filter on SMS class names. For example, you can filter on a management class to back up only data sets that are assigned to that class.

Restoring Data

You can use DFSMSdss to restore data to DASD volumes from DFSMSdss-produced dump volumes and to the same or different device types.

DFSMSdss allows you to perform the following restore-related tasks:

- Restore an entire VSAM sphere with a single operation.
- Specify whether DFSMSdss continues or cancels a restore when a permanent read error occurs.
- Lock data sets on shared DASD volumes using dynamic allocation.
- Rename a data set during a restore.
- Redefine the block size of physical sequential, PDS, or PDSEs while recovering those data sets.
- Restore data sets that have an undefined data set organization to an unlike device of larger capacity.
- Specify the target status of system-managed generation data sets.
- Restore a user catalog without bringing down your application program. This allows continuous applications such as Db2 and IMS™ to remain active while a damaged user catalog is replaced.
- Restore data sets to system-managed volumes and non-system-managed volumes. If you have RACF authorization, you can specify that the ACS routines be bypassed. Otherwise the ACS routines are invoked to assign SMS classes to the data sets.

Moving Data with SnapShot

SnapShot is a function of the RAMAC Virtual Array (RVA). SnapShot allows you to make a very quick copy of a set of tracks (an entire volume, a data set, or a random set of tracks). The copy operation is completed with only a few I/Os to the device.

DFSMSdss provides two types of SnapShot services:

Native SnapShot
Data is snapped (quickly copied) directly from the source location to the target location. This function occurs when you issue a DFSMSdss COPY command to copy volumes, tracks, or data sets from one DASD volume to another. DFSMSdss uses this method whenever the source and target data are on like devices in the same partition on the same RVA subsystem, and no reblocking is required. DFSMSdss can use this method whether the CONCURRENT keyword is specified. With SnapShot, the copy of the data is logically and physically complete as soon as the snap is complete.

Virtual Concurrent Copy
Data is snapped from the source location to an intermediate location and then gradually copied to the target location using normal I/O methods. This method operates in a fashion almost identical to the

existing concurrent copy (CC) operation. All DFSMSdss users and callers of the DFSMSdss application programming interface (API) (such as DFSMShsm, Db2, and IMS) can continue to use and receive the services of the functionally identical concurrent copy operation. Using virtual concurrent copy, the copy or dump of the data is logically complete after the source data is snapped to the intermediate location. The processing is physically complete after the data has been moved to the target media. It is also possible to perform concurrent copy on VM minivolumes using virtual concurrent copy.

DFSMSdss Stand-Alone Services

The Stand-Alone Services function is available to all supported releases of DFSMSdss. This latest version of IBM's Stand-Alone restore function is a single-purpose program. It is designed to allow the system programmer to restore vital system packs during disaster recovery without needing to rely on a z/OS environment. Stand-Alone Services runs independently of a system environment either as a "true" stand-alone system or under a VM system.

Stand-Alone Services can perform either a full-volume or a tracks restore from dump tapes that are produced by DFSMSdss or DFDSS.

Related reading: For additional information about the DFSMSdss Stand-Alone Services function, see *z/OS DFSMSdfp Storage Administration*.

Converting to System-Managed Data

DFSMSdss is the primary tool for converting system-managed data to non-system-managed data or non-system-managed data to system-managed data. DFSMSdss allows conversion with or without data movement.

Converting by Moving Data

DFSMSdss commands can move and convert data sets between system-managed and non-system-managed volumes. Using DFSMSdss, you can perform the following data conversion tasks:

- Recognize and preserve SMS classes. Each system-managed data set has a set of associated class names. These names can identify the data set's data class, management class, and storage class. DFSMSdss recognizes and, if specified, preserves these names when copying, dumping, and restoring data.
- Copy, dump, and restore system-managed data sets. DFSMSdss moves data sets between system-managed volumes and non-system-managed volumes. The SMS class names are either added to or taken away from the data set.
- Filtering by SMS class names. You can filter both system-managed and non-system-managed data sets by class names.

Converting Volumes without Moving Data

DFSMSdss provides a command that allows you to convert both non-SMS-managed volumes to SMS-managed volumes or SMS-managed volumes to non-SMS-managed volumes without moving any data. Before attempting to convert volumes, you can verify that DFSMSdss can convert all data sets on that volume.

Using Additional DFSMSdss Facilities

The sections that follow describe additional facilities that you can use with DFSMSdss.

Choosing between Two Processing Methods—Logical and Physical

DFSMSdss can perform either logical or physical processing. If you perform a logical dump on a data set, DFSMSdss restores it logically; if you perform a physical dump, DFSMSdss restores it physically. Logical processing operates against data sets independent of physical device format. Physical processing moves

data at the track-image level and operates against volumes, tracks, and data sets. Each type of processing offers different capabilities and advantages.

Logical Processing

DFSMSdss treats each data set and its associated information as a logical entity during logical processing. DFSMSdss processes an entire data set before beginning the next one. DFSMSdss moves each data set as a set of data records, allowing data movement between devices with different track and cylinder configurations. DFSMSdss uses the catalog and the VTOCs to select data sets for logical processing.

Physical Processing

DFSMSdss moves data based on physical track images during physical processing. Because DFSMSdss moves data at the track level, the target device's track sizes must equal the source device's track sizes. Therefore, you can use physical processing only between like devices. Physical processing operates on volumes, ranges of tracks, or data sets. For data sets, physical processing relies only on volume information in the VTOC and VVDS for data set selection. Only that part of a data set that resides on the specified input volume is processed.

Selecting Data Sets by Name or by Characteristic

You can select data sets for DFSMSdss processing by filtering on criteria that you specify. DFSMSdss can filter on fully or partially qualified data set names and on various data set characteristics.

Filtering by Data Set Names

You can select data sets for inclusion in or exclusion from DFSMSdss processing based on data set name. Data set names for DFSMSdss filtering can be fully qualified or partially qualified and can consist of one or more qualifiers.

Filtering by Data Set Characteristics

Besides filtering on data set names to process groups of data sets, you can filter on data set characteristics, such as the following:

- Allocation type (cylinder, track, block, absolute track, or movable)
- Creation date (absolute or relative)
- Expiration date (absolute or relative)
- Last-referenced date (absolute or relative)
- Data set organization
- Data set size (number of allocated or used tracks)
- Number of extents
- Volume type (single-volume or multivolume)
- Data-set-changed flag (on or off)
- SMS class names

Invoking DFSMSdss

You can invoke DFSMSdss processing through job control language (JCL), Interactive Storage Management Facility (ISMF), or an application program.

Invoking DFSMSdss with JCL

You can use JCL statements to invoke DFSMSdss and to define the data sets that DFSMSdss needs.

Invoking DFSMSdss with ISMF

You can use ISMF menu-driven panels to submit commands to DFSMSdss. If you fill in the panels with the values that you want, ISMF generates the JCL.

Related reading:

- For additional information about ISMF, see "Interactive Storage Management Facility Menus" on page 17.
- For additional information about using DFSMSdss with ISMF see the ISMF online help panels and *z/OS DFSMS Using the Interactive Storage Management Facility*.

Invoking DFSMSdss from Your Application Program

You can invoke DFSMSdss from an application program using the application interface. You can also invoke DFSMSdss to run in a separate address space by using the cross-memory application interface. This interface allows you, for example, to specify control variables, and to gather information on free and used space on a volume or in a data set. Any program that calls DFSMSdss must either be authorized by Authorized Program Facility (APF) or must invoke DFSMSdss using the cross-memory application interface. For more information on APF protection and authorization, see *z/OS MVS Programming: Assembler Services Guide*.

When you invoke DFSMSdss from an application program, you can provide a user interaction module (UIM) to interact with DFSMSdss during processing at various points. The UIM lets you perform the following tasks:

- Replace, insert, delete, or modify a SYSIN record after DFSMSdss reads it, or a SYSPRINT record when DFSMSdss is ready to print it.
- Replace, insert, delete, or modify a write-to-operator (WTO) message before DFSMSdss writes it.
- Control processing and gather auditing information for an individual data set during a logical copy, dump, or restore.
- Control processing for concurrent copy operations.
- Insert statistical records during a logical dump.
- Intercept records being dumped or supply records to be restored.

Using Auxiliary Commands to Control Run-Time Options

Auxiliary commands can control the way DFSMSdss runs. For example, you can use auxiliary commands to perform the following tasks:

- Start serial task scheduling (the default). DFSMSdss executes only one task at a time.
- Start parallel task scheduling. DFSMSdss executes two or more tasks concurrently if the required system resources (such as virtual storage, DASD, or tape volumes) are available.
- Control how DFSMSdss runs, based on the return codes of completed operations.
- Write a message to the system console.
- End your DFSMSdss job after the currently running operations and scheduled tasks complete.

Printing DASD Data

You can use DFSMSdss to print DASD data to the SYSPRINT data set or to a sequential data set in print format. The records in the output data set are blocked to better use DASD space. For data set printing, tracks are printed in the logical sequence of the data set on the volume, not in the physical cylinder and head sequence.

You can print the following data:

- A non-VSAM data set that is specified by a fully qualified name. You need not identify the data set location.
- VSAM data sets, including key-range data sets at the component level but not at the cluster level.
- Ranges of tracks.
- All or part of the VTOC. You need not identify the VTOC location.

You can optionally print only the tracks on which read errors occur for any of the preceding items.

Controlling Access to DFSMSdss Tasks and Data

You can use RACF to protect resources such as DASD volumes, tape volumes, and data sets against unauthorized access. You can also use RACF to limit the use of certain DFSMSdss functions to privileged users.

Protecting Data

You can grant access to a data set at the volume level or at the data set level. DFSMSdss uses RACF to check the user's access authority to the volume.

A data set can be either RACF-protected, password-protected, or both. When the data set is protected by RACF, DFSMSdss ignores the data set password.

When copying or restoring data, DFSMSdss checks that users have sufficient authority to create or overlay the target data sets. In addition, DFSMSdss tries to protect the target data sets with the same RACF protection as the source data sets.

Related reading: For additional information about the RACF profile, see *z/OS Security Server RACF Security Administrator's Guide*.

Protecting DFSMSdss Commands and Keywords

You can protect certain DFSMSdss commands and keywords against unauthorized use by defining the appropriate RACF FACILITY class profile. DFSMSdss invokes RACF checking to ensure that users have sufficient authority to perform the function.

Related reading: For additional information about the FACILITY class profiles used by DFSMSdss, see the *z/OS DFSMSdfp Storage Administration*.

Customizing DFSMSdss with Installation Exits

Installation exits provide a way for your installation to tailor DFSMSdss. Replace these exit routines with your own routines to perform the following tasks:

- Control or override the authorization checking of protected data sets
- Control the duration of the enqueue of the VTOC
- Verify or change the block size of a reblockable data set
- Specify installation options and defaults

Related reading: For additional information about DFSMSdss installation exits, see *z/OS DFSMS Installation Exits*.

Chapter 9. Understanding DFSMShsm

DFSMShsm provides automatic management of low activity and inactive data, and automatic backup and recovery of active data in both system-managed and non-system-managed environments. DFSMShsm accomplishes this by providing the following types of management:

Automatic Storage Management
DFSMShsm is a tool that improves productivity by effectively managing storage. DFSMShsm uses a hierarchy of storage devices in its automatic management of data that relieves users from manual storage management tasks and improves DASD utilization. For more information, see "Device Hierarchy" on page 53.

Space Management
DFSMShsm automatically manages DASD space by enabling active data sets to be kept on fast access storage devices. DFSMShsm frees available space on user volumes by releasing over allocated space, deleting eligible data sets, and by moving low activity data sets to lower cost-per-byte devices.

By using DFSMShsm in conjunction with your automatic class selection (ACS) routines to implement the tape mount management methodology, you can write multiple output data sets to a single tape, with a single tape mount. This not only improves tape use, but also greatly reduces the number of tape mounts required by your installation.

For more information, see "Managing Space" on page 56.

Availability Management
DFSMShsm makes data available by automatically copying new and changed data sets to backup volumes. If the original data sets are damaged or accidentally deleted, having backup versions ensures that the data can be made available.

DFSMShsm provides the ability to back up aggregate groups of critical data sets and programs that can be taken to remote locations and used for disaster recovery or other business needs.

DFSMShsm creates and manages backup versions of storage groups defined within copy pools. DFSMShsm recovers a fast replication backup version from the volume or copy pool level.

DFSMShsm also provides the secondary host promotion function to improve data availability in instances where a DFSMShsm host processor experiences a failure in a multisystem environment. Using the secondary host promotion function, with a cross-system coupling facility (XCF) in either a basic or Parallel Sysplex environment, a secondary host can take over the unique functions that were performed by a failed primary host. The secondary host promotion function also enables other DFSMShsm hosts to take over secondary space management functions from either a primary or secondary host that has failed. If a promoted host also fails, any remaining host that is eligible for promotion will take over.

For more information, see "Managing Availability" on page 61.

DFSMShsm provides a number of functions to support space and availability management. Full exploitation of DFSMShsm services in DFSMS environments requires the use of DFSMSdss for certain functions.

Device Hierarchy

Device hierarchy is defined as levels of storage devices with each level having its different access speeds, costs-per-byte, and storage capacities.

At the top of the device hierarchy is the DASD storage control with its optional, high speed, cache storage.

In the middle of the device hierarchy is DASD. Data that needs to be accessed more frequently is typically stored on DASD.

Removable media is the lowest level in the device hierarchy. It can be used with optical and tape library data servers for the fastest response time, as well as with stand alone optical and tape drives with shelf storage. Removable media has the slowest initial access time; however, it has the lowest cost per megabyte. The storage capacity of removable media is limited only by the number of volumes your installation is willing to use and manage. In addition, volumes can be removed to a remote site to protect them from damage in the event of a disaster.

Functioning Independently of Device Type

DFSMShsm provides the capability to function independently of the device type to which data is being transferred. That is, when you define a volume to DFSMShsm, you specify its device type. DFSMShsm then does all the work of converting for different track capacities and different allocation specifications necessary to accommodate the data sets sent to various volumes. For most access methods, this device independence allows migrating or backing up a data set from one kind of device and recalling or recovering it to another kind of device.

Data Hierarchy

DFSMShsm uses advanced concepts in data hierarchy not always found in similar products. The data hierarchy is comprised of the following types of data:

Active data
Active data is frequently used and seldom migrated data. The data normally resides on DASD, for fast access, and is in user format. The DASD containing active data are called level 0 volumes, and they can be managed by DFSMShsm.

Low activity data
Low activity data is sometimes used, but not often. This type of data is eligible to be migrated by DFSMShsm, or has already been migrated by DFSMShsm, to a more cost efficient type of storage.

Inactive data
Inactive data is data that was dumped or is at the incremental backup level. Dumps and incremental backups are taken of the active data and of the DFSMShsm control data sets and journal. These dumps and incremental backups can then be used to recover data sets, or entire volumes, if they are lost due to damage or accidental deletion.

Inactive data also includes backups of aggregate groups. These backups can be transported to a remote site for disaster recovery, or they can be used for other business purposes.

Preserving Data Set Integrity

DFSMShsm protects a data set while it is processing. Storage administrators can use management class attributes to prevent migration and backup of data sets. These attributes ensure data integrity by disallowing the data sets to be moved or copied with DFSMShsm migration or backup functions.

If data is moved or copied, DFSMShsm ensures that the migration or backup operation is performed correctly before it erases any data set or resets its changed indicator. DFSMShsm ensures the integrity by using either volume reserves or global resource serialization capability. For more information about global resource serialization, see "Using SMS in a Multisystem Environment" on page 22.

Using DFSMShsm Control Data Sets

DFSMShsm control data sets (CDS) are system-type data sets that DFSMShsm uses to keep track of all DFSMShsm-owned data. They consist of migration, backup, and offline control data sets. The control data

sets are an inventory of low activity and inactive data that was stored by DFSMShsm and used to manage its environment. DFSMShsm logs its transactions and maintains multiple backup versions of its CDSs for recovery purposes as specified by the user.

DFSMShsm can use VSAM KSDS extended addressability (EA) capabilities that use any of the following serialization techniques for accessing its CDSs:

- Record-level sharing (RLS) access mode
- CDSQ serialization
- CDSR serialization

VSAM EA capabilities allow each migration control data set (MCDS), backup control data set (BCDS), and offline control data set (OCDS) cluster to exceed the 4 GB size. The MCDS and BCDS can span up to four unique KSDS clusters. The OCDS is limited to a single cluster. The same serialization technique must be used to access all CDSs.

DFSMShsm-Owned Volumes

DFSMShsm-owned volumes contain low activity and inactive user data sets that were moved (migrated data) or copied (dumped and backed up data) into the DFSMShsm-owned portion of the storage hierarchy. The data is stored in a space-saving format. The data stored in a space-saving format is not directly accessible by users, but must be returned to a level 0 volume in user format before users can access it. Information about DFSMShsm-owned data is recorded in the inventory maintained in the DFSMShsm control data sets.

Migration level 1 and migration level 2 volumes
Low activity data is stored on migration level 1 (ML1) and migration level 2 (ML2) volumes. The DFSMShsm automatic management of ML1 and ML2 volumes can be supplemented by DFSMShsm commands to manage them manually.

Backup tapes
Backup tapes contain the backup copies of data sets placed there on a daily basis.

Spill backup tapes
Spill backup tapes contain data moved from backup volumes as a result of recycle processing. The data is written in a DFSMShsm space saving format. The recycle function is a tape consolidation process performed on ML2 and backup tapes that become sparsely populated with valid data.

Dump tapes
Dump tapes contain all the data of a physical volume and are written in a DFSMSdss space saving format. Several dumps from different DASD volumes can be automatically stacked onto a single tape.

Aggregate data tapes
Aggregate data tapes contain user defined groups of data sets that were copied by aggregate backup for recovery at a remote site. Aggregate control tapes can contain control information, data from level 0 DASD, data from ML2 and user tape, and instruction and log data.

Alternate tapes
ML2 and backup tapes can have alternate tapes. The alternate tapes are produced by either duplicating the original tapes, or by using the duplex tape function to concurrently create a copy when the original ML2 or backup tapes are created. Alternate tapes provide site disaster protection, supplementing the ABARS (aggregate backup and recovery support) aggregates and protecting against media damage.

Fast replication target volumes
Fast replication target volumes contain the fast replication backup copies created during DFSMShsm fast replication processing. Fast replication target volumes are defined within copy pool backup storage groups.

Managing Space

You can use DFSMShsm to keep DASD space available. DFSMShsm can automatically and periodically move low activity data sets from user DASD volumes to DFSMShsm-owned volumes. DFSMShsm also reduces the space occupied by data on both the user DASD volumes and the DFSMShsm-owned volumes.

To provide you with more control, daily space management is divided into automatic primary and automatic secondary space management functions. Each of these functions can have a processing cycle. Each can be started at a specified time of day and end when all data that is managed by DFSMShsm is processed, or end when the ending time of day (storage administrator specified) is reached.

Utilizing Automatic Primary Space Management

During automatic primary space management, DFSMShsm can process a maximum of 15 volume migration tasks concurrently. This activity consists of the deletion of temporary data sets, deletion of expired data sets, and the release of unused and overallocated space, and migration. Each task processes its own separate user DASD volume. The storage administrator selects the maximum number of tasks that can run simultaneously, and specifies which days and the time of day the tasks are to be performed.

If a volume is system-managed, DFSMShsm uses the storage group threshold of occupancy as a measure of how much free space to provide. The volumes processed are those in storage groups having the automatic migration attribute. Storage groups can also be used to restrict the processing to a single system or a system group.

DFSMShsm manages each data set on a volume according to the management class attributes associated with the data set.

If a volume is non-system-managed, DFSMShsm manages each data set on a volume according to the attributes of the volume on which the data set resides.

Figure 5 on page 57 shows the migration paths of data sets managed in system-managed and non-system-managed environments.
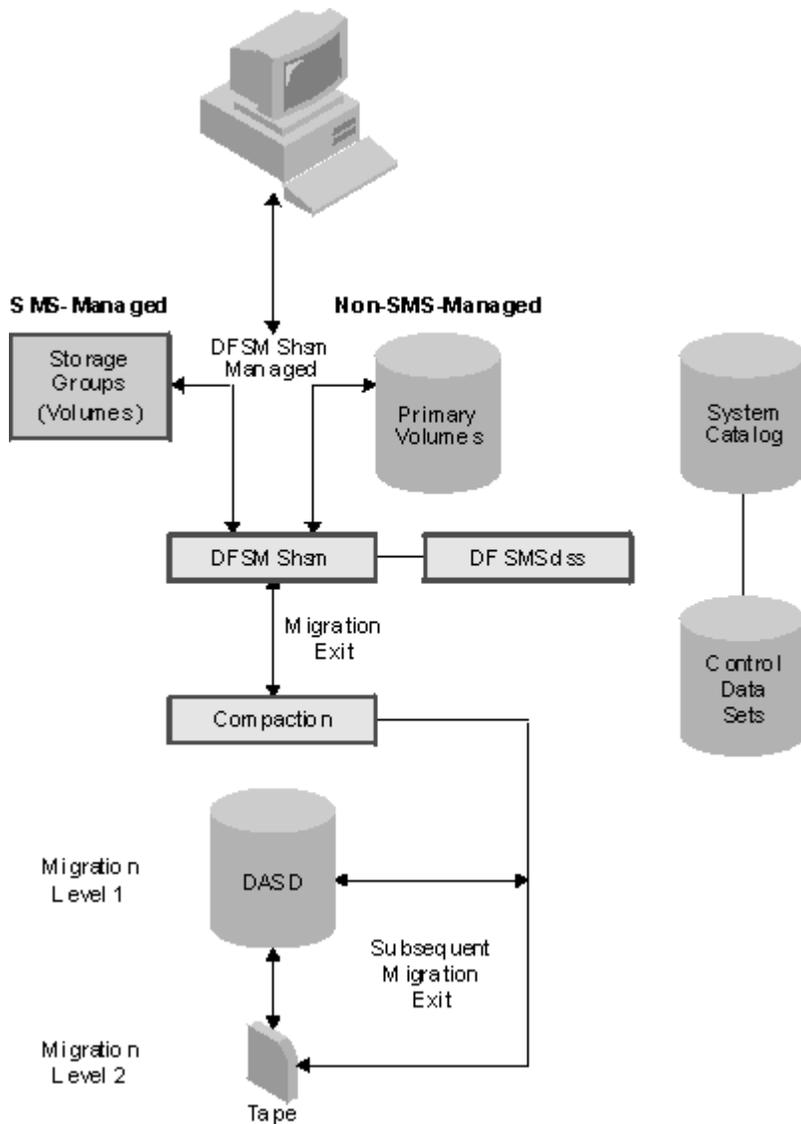
Figure 5: Automatic volume and secondary space management

Using Automatic Secondary Space Management

Automatic secondary space management prepares the computing system for the automatic primary space management work load by freeing space on DFSMShsm-owned volumes. Secondary space management deletes expired migrated data sets and migrates eligible data sets from ML1 volumes to ML2 volumes. The storage administrator specifies the day in the cycle and the time of day that automatic secondary space management is to be performed.

Invoking Automatic Interval Migration

Automatic interval migration is an option that invokes migration when DFSMShsm-managed volumes become full during high activity periods. If the storage administrator chooses this option, DFSMShsm automatically checks the level of occupancy of all DFSMShsm-managed volumes periodically. If the level of occupancy for any volume exceeds a given threshold, DFSMShsm automatically performs a subset of the space management functions on the volume. The threshold you select should be one that would be exceeded only when your installation's activity exceeds its usual peak. For those volumes requiring interval migration, DFSMShsm can process up to 15 volume migration tasks concurrently.

During automatic interval migration on a volume, the expired data sets are deleted, then the largest eligible data sets are moved first so that the level of occupancy threshold can be reached sooner. Data sets are not migrated from ML1 to ML2 volumes during interval migration.

For system-managed volumes, DFSMShsm uses the level of occupancy thresholds established for the storage group to perform this function.

For non-system-managed volumes, DFSMShsm uses the level of occupancy thresholds established for the DFSMShsm-managed volume to perform this function.

Using Automatic Recall

This recall process returns a migrated data set from a ML1 or ML2 volume to a DFSMShsm-managed volume. When a user refers to the data set, DFSMShsm reads the system catalog for the volume serial number. If the volume serial number is MIGRAT, DFSMShsm finds the migrated data set, recalls it to a DFSMShsm-managed volume, and updates the catalog. The result of the recall process is a data set that resides on a user volume in a user readable format. The recall can also be requested by a DFSMShsm command.

Figure 6 on page 58 shows the recall paths of data sets managed with and without system-managed storage.
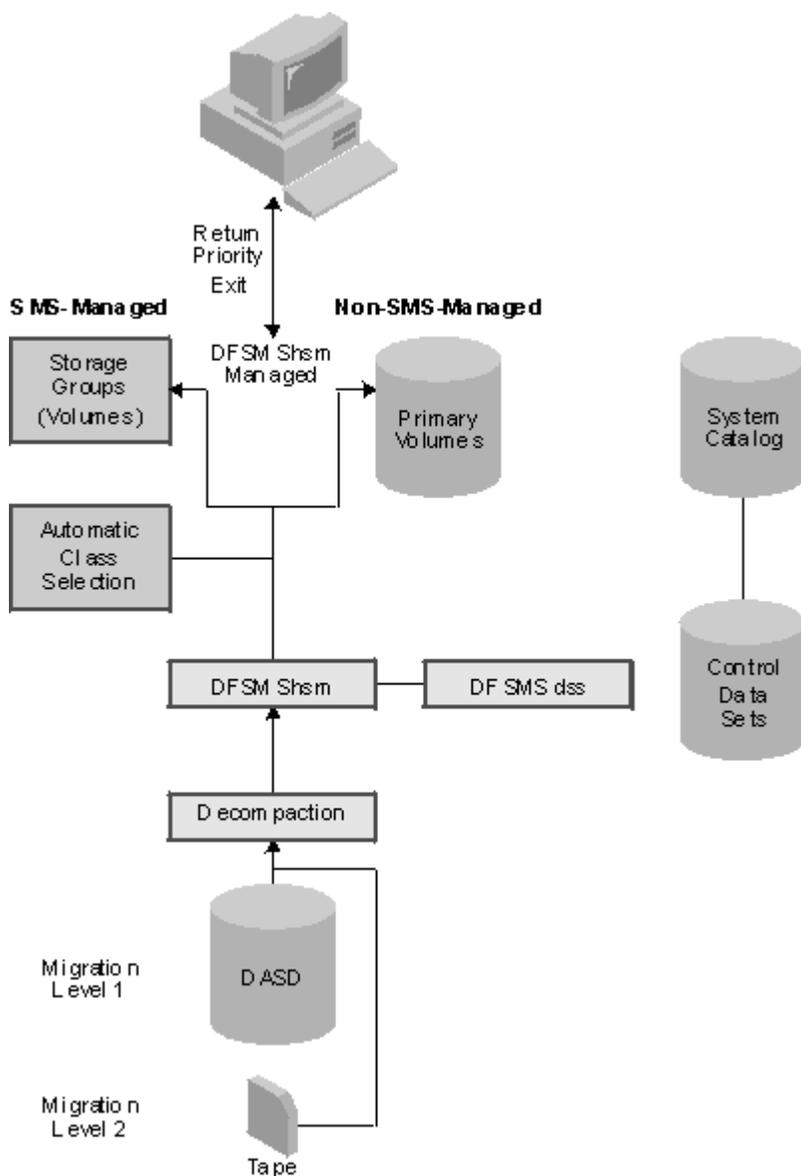
Figure 6: Automatic recall

For system-managed environments, DFSMShsm invokes the automatic class selection routines to determine whether a data set should be recalled as managed with or without system-managed storage. Automatic class selection chooses the storage group to which the data set is recalled. The system chooses the specific volumes where the data set will reside, from the storage groups.

For non-system-managed environments, DFSMShsm returns a migrated data set to the DFSMShsm-managed volume with the most free space or to a specified target volume.

Using the Common Recall Queue to Recall Data Sets

One or more processors with DFSMShsm installed and running that share a common BCDS, MCDS, OCDS, and journal is called an HSMplex. Each host in an HSMplex can add, scan, select, and remove recall requests when utilizing the common recall queue (CRQ). When a host selects a request from the CRQ, it processes it as if the request were its own.

Another advantage of the CRQ is the reduction in the number of tape mounts. One DFSMShsm host can process recalls that originate from other DFSMShsm hosts (recalls are on the same tape). The CRQ provides the following enhancements to recall processing:

- Workload balancing
- Priority optimization
- Tape mount optimization
- Request persistence
- Flexible configurations
- HSMplex information gathering

Figure 7 on page 59 shows an overview of the common recall queue. All hosts can process requests from the shared queue where they have been placed.
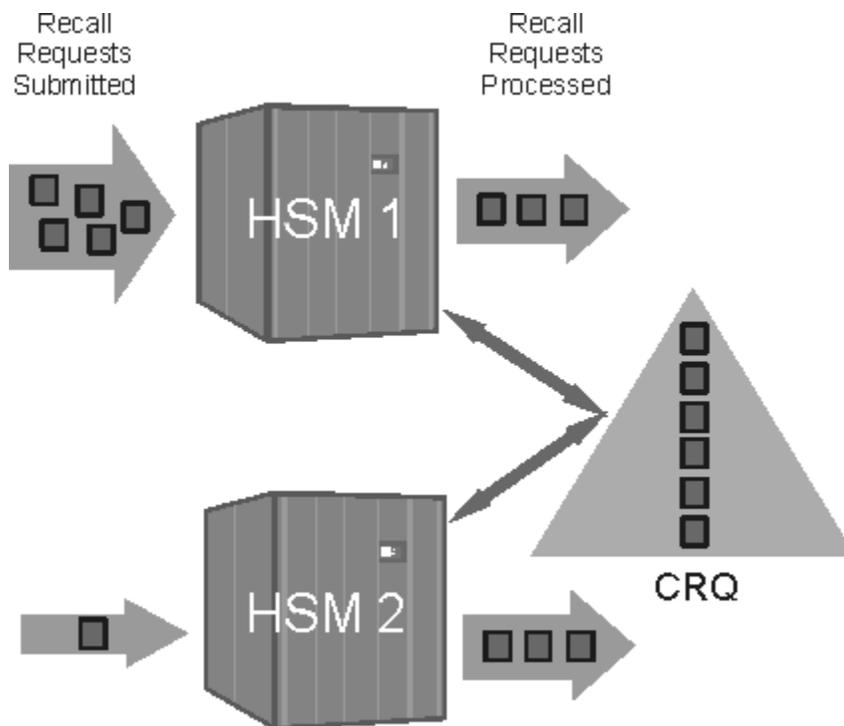
Figure 7: The common recall queue

See *z/OS DFSMShsm Storage Administration* for more information on the common recall queue.

Using Fast Subsequent Migration to Reconnect Data Sets

The fast subsequent migration function allows unchanged data sets that are recalled from ML2 tape to be reconnected to the same ML2 tape. Data sets must meet standard migration eligibility checks as well as a reconnectability check to be eligible for reconnection. The fast subsequent migration function occurs for non-VSAM data sets and for VSAM data sets that have no defined alternate indexes or paths.

Once a data set that is recalled from tape meets normal eligibility criteria, the fast subsequent migration (a fast remigrate) occurs during volume migration. Fast subsequent migrations occur ahead of migrations that cause data movement. Because the data is still on tape from the previous migrate, the fast subsequent migration involves both creating and updating DFSMShsm control records. It does not involve moving the data to tape.

Reconnection can occur during individual data set migration or during volume migration. See *z/OS DFSMShsm Storage Administration* for more information on fast subsequent migration.

Saving Space on DFSMShsm-Owned Volumes

DFSMShsm can save space on migration volumes with its space saving functions. DFSMShsm generally moves and copies only used, not overallocated, space to its migration volumes. These DFSMShsm functions are available in system-managed and non-system-managed environments.

The following list represents the space saving functions:

Data compaction
DFSMShsm can save space on DFSMShsm-owned volumes by compacting each data set as it is processed. The data compaction functions are available with or without using the DFSMSdss component of DFSMS as the data set mover. DFSMShsm can compact data during data set backup and migration, or by using DFSMSdss compression during physical volume dumps and aggregate backup.

The IBM RAMAC Virtual Array Subsystem also provides its own data compaction, making compaction by either DFSMShsm or DFSMSdss unnecessary.

Small data set packing
Small data set packing (SDSP) saves space on ML1 volumes by writing small data sets (fewer than 800 KB) as records in a VSAM data set called an SDSP data set. The space saving occurs because the DASD data set allocation requires allocation in full tracks. SDSP lets you give those data sets allocation in 2093-byte units.

Storage administrators can specify the size of data sets eligible to migrate to SDSP data sets in kilobytes or tracks. The storage administrator must specify which ML1 volumes have SDSP data sets.

Partitioned data set compression
Partitioned data set compression reduces the allocated space for a data set as members are deleted or replaced. DFSMShsm provides an automatic method of compressing partitioned data sets during space management by doing migration and recall when the partitioned data set exceeds a certain number of extents.

DFSMShsm maintains user information that is in partitioned data set directories when it performs partitioned data set compression.

User data set reblocking
During recall and recovery, the process of reblocking user data sets changes the number of records in a physical block and thereby uses the space on the DASD user volume more efficiently. Data movement using DFSMShsm reblocks physical sequential data sets. Data movement using DFSMSdss reblocks physical sequential and partitioned data sets.

Optimum DASD blocking
Optimum blocking of data is an option that saves space on DFSMShsm-owned DASD volumes by using an optimum block size for storing the maximum data on each track. The optimum block size is determined, based on the device type of the DASD device.

Tape recycle
Data on tapes is invalidated over time by the expiration or recall of migrated data sets or the generation of more recent backup data sets. DFSMShsm provides the capability of recycling backup or

migration tapes when they contain less than an installation defined percentage of valid data. Recycling transfers the valid data sets from these tapes and consolidates the data onto fewer tapes, thus leaving the recycled tapes available for reuse. Tape recycle can also be used to move data from one tape technology type to another. Options are available to select input tapes by volume serial ranges, which can be used to identify tapes of a given technology.

Tapes eligible for recycle processing are logically sorted, with those containing the least amount of valid data (per cartridge type, standard or enhanced capacity) processing first.

To facilitate rapid recycle processing, up to 15 tape processing tasks can run simultaneously on a processor, with multiple tape buffers being used for improved throughput. The number of recycle tape processing tasks can be changed dynamically, even during recycle processing, to any number from one to 15.

Dump tapes and ABARS tapes can be reused automatically without recycle; dump tapes when they reach the end of their retention period, ABARS tapes when they reach an expiration date, or when aggregate roll off occurs.

If data in single-file format on a backup or ML2 tape is overwritten or becomes unreadable, DFSMShsm can retrieve all the undamaged data from that tape.

Managing Availability

Availability management must ensure that a current backup copy of a data set, a volume, or a set of application data sets is available for recovery at a point in time. To accomplish this objective effectively requires that planning be done from the viewpoint of recovery.

Protecting Critical User Applications

To protect your application programs and data, use aggregate backup and aggregate recovery (ABARS) processing as your disaster survival solution. Rather than managing disaster survival by dumping and restoring volumes, ABARS manages disaster survival by copying the data sets belonging to an application, along with relevant control information, on a set of portable device independent tape data and control files. Thus, critical applications can be recovered at one or more remote sites.

Backing Up Aggregate Groups

Aggregate backup and recovery is a process to back up and recover any user defined group of data sets that are vital to your business. Up to 64 ABARS commands can run concurrently.

You can use ISMF panels to identify these data sets as an aggregate group. DFSMShsm uses the data set lists, along with other control information stored in the associated aggregate group and management class, to manage the aggregate backup process. For more information about aggregate groups, see "Aggregate Groups for Disaster Backup and Recovery" on page 21.

DFSMShsm works with the other functional components of DFSMS to provide enhanced backup and recovery capabilities for aggregate groups. When you define the aggregate group, you specify characteristics of that group in its management class. You can specify, for example, how many copies of each backup version you want. The backup copies are managed according to the aggregate group attributes you define in the management class. You can use the same management class attributes for multiple aggregate groups whose backup copies have the same management needs. The attributes assigned to an aggregate group are tracked by the aggregate backup and recovery function and can be used to create the proper environment at the recovery site.

You can automate the aggregate backup process by using an DFSMShsm assembler macro to submit the ABACKUP command.

The DFSMS functional component DFSMSrmm can be used to track the movement of the aggregate backups, further automating your disaster backup process. See Chapter 10, "Managing Removable Media with DFSMSrmm," on page 73.

shows the movement of data sets during aggregate backup.
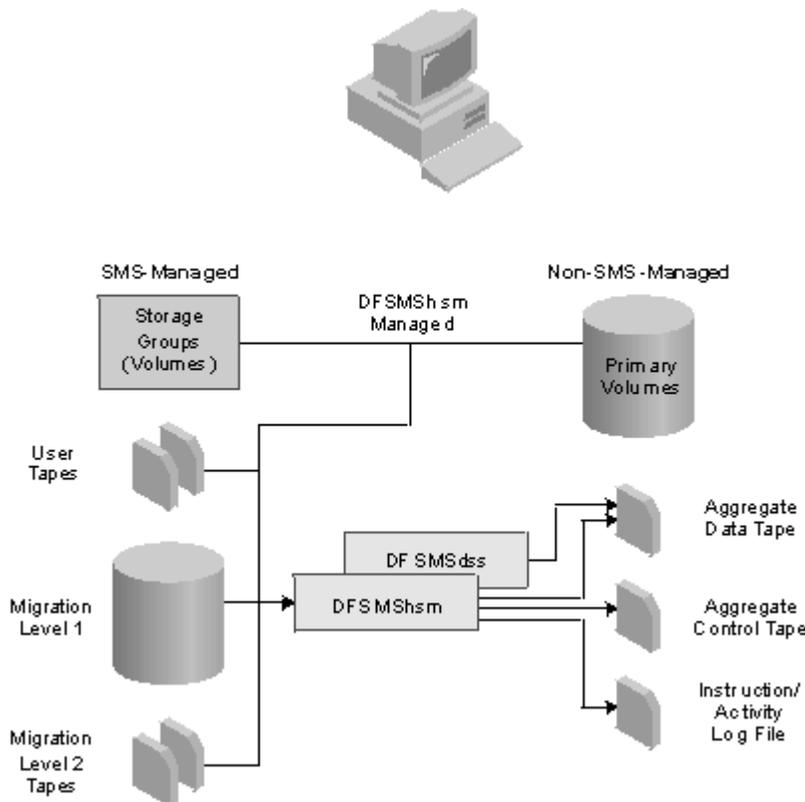
Figure 8: Aggregate backup

See *z/OS DFSMShsm Storage Administration* for more information on the ABARS function.

Recovering Aggregate Groups

During aggregate recovery, the backed up data sets are recreated at the recovery site. Data sets backed up as migrated data sets can be returned to the same level as when backed up, or they can all be recovered to ML1 DASD or to ML2 tape. SMS-managed generation data sets being recovered to level 0 DASD can be recovered with a status of deferred, active, rolled off, or source, where source means restoring the data set to the status it had when backed up. All data sets backed up from user volumes are returned to user volumes.

A generation data group (GDG) base can be defined at the recovery site without having to back up any of the generation data sets (GDSs). The GDG base name is defined, if one does not already exist, during ARECOVER processing, and before any GDSs are recovered.

During aggregate recovery processing, the aggregate backup volumes can be automatically defined to RACF if the appropriate DFSMShsm commands are specified and if RACF is active at the recovery site.

Recovering User Data Sets and Volumes

User data sets and volumes can be recovered for reasons other than disaster survival. DFSMShsm availability management uses the following functions as a means of ensuring that the latest version can be recovered:

- DFSMSdss full volume dump
- DFSMShsm incremental backup
- DFSMShsm inline backup
- Fast replication backup

Automatic Invocation of DFSMSdss Full Volume Dump

DFSMShsm automatically invokes DFSMSdss functions to dump DFSMShsm-managed user volumes and ML1 volumes. Each DFSMShsm-managed volume and each ML1 volume can be assigned a dump class that defines the following:

• When and how to dump the volume

• How often to dump the volume

• How long to keep the dump copy

DFSMShsm can dump up to five concurrent copies of a volume in five dump classes. At least two concurrent copies are required to provide an on site and an off site dump tape. Each group of up to five concurrent copies of a volume is known as a generation. DFSMShsm maintains up to 100 generations of each volume.

DFSMShsm can dump up to 15 volumes concurrently. Each dump task selects its own dump tapes. The storage administrator selects the maximum number of concurrent tasks allowed.

DFSMShsm creates and maintains an inventory of the dump copies created. Figure 9 on page 63 shows the dump paths of data sets managed with and without SMS.
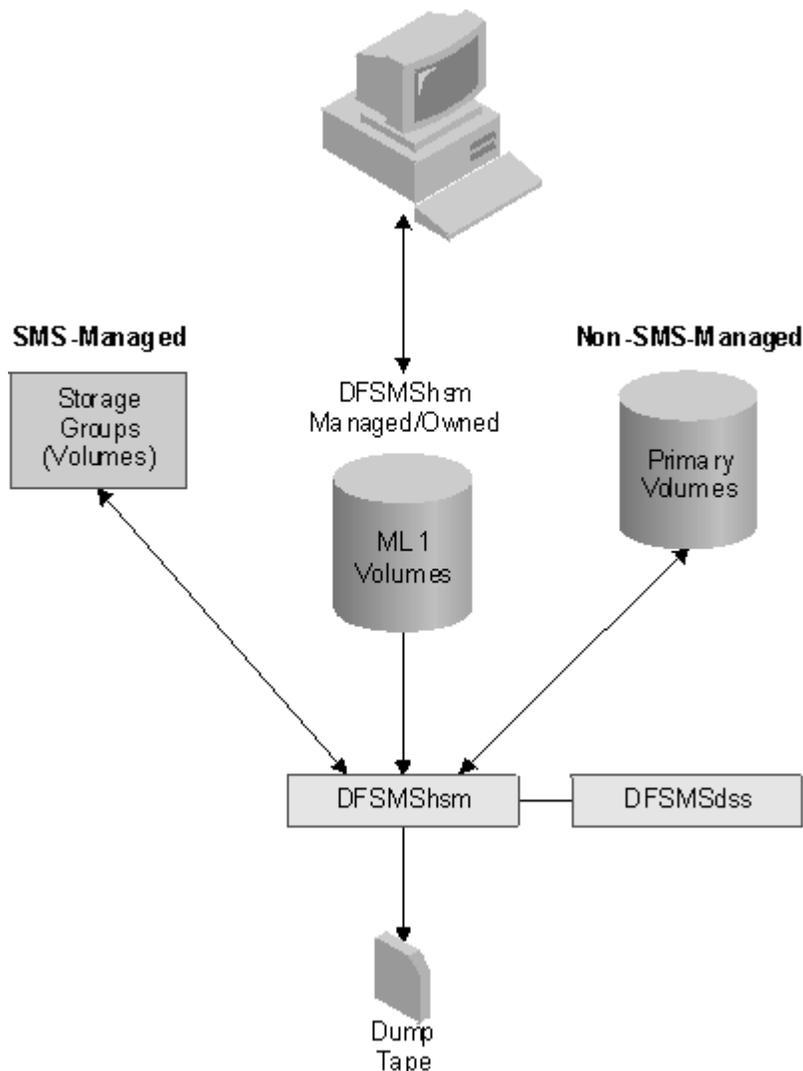
Figure 9: Automatic dump and restore

For system-managed volumes, DFSMShsm uses the automatic dump attributes and dump class names specified in the storage group definition.

For non-system-managed volumes, DFSMShsm uses the volume automatic dump attribute and dump class names specified for the volume.

Automatic DFSMShsm Incremental Backup

Backup of only new or changed user data sets is known as incremental backup. Automatic incremental backup ensures that up to date copies of new and changed data sets exist in case the original data sets are damaged or accidentally deleted. On the days and at the time specified by the storage administrator, DFSMShsm automatically copies new and changed data sets on DFSMShsm-managed volumes to DASD or tape backup volumes.

The DFSMShsm copy is a backup version that resides in an optionally compacted, device independent format on a lower cost-per-byte storage device. Users can recover their own data sets without intervention by a storage administrator.

During the backup process, DFSMShsm prevents anyone from updating the data while the data set is being copied, unless the data set is backup-while-open eligible. If the data set is backup-while-open eligible, DFSMShsm invokes DFSMSdss to perform the incremental backup, but the data set change indicator is not reset, so the data set gets backed up again the next time backup is performed. DFSMShsm records the location of the backup version in its backup control data set and maintains records of its tape volumes in the offline control data set.

DFSMShsm can back up a maximum of 15 user volumes concurrently. Each incremental backup task selects its own daily backup volume, either tape or DASD. The storage administrator selects the maximum number of concurrent tasks.

DFSMShsm allows automatic retention of up to 100 backup versions of each data set. DFSMShsm has the capability to determine and delete those backup versions that are no longer needed according to criteria supplied by the storage administrator.

To protect your inventory of backed up and migrated data, the three DFSMShsm control data sets and journal are backed up as a synchronized set at the start of automatic incremental backup. DFSMShsm maintains a user specified number of backup and journal generations. The journal, a record of all transactions posted to the control data sets between backups, is used in the recovery process to update the recovered control data sets to the point of failure.

shows the backup paths of data sets managed with SMS and without SMS.
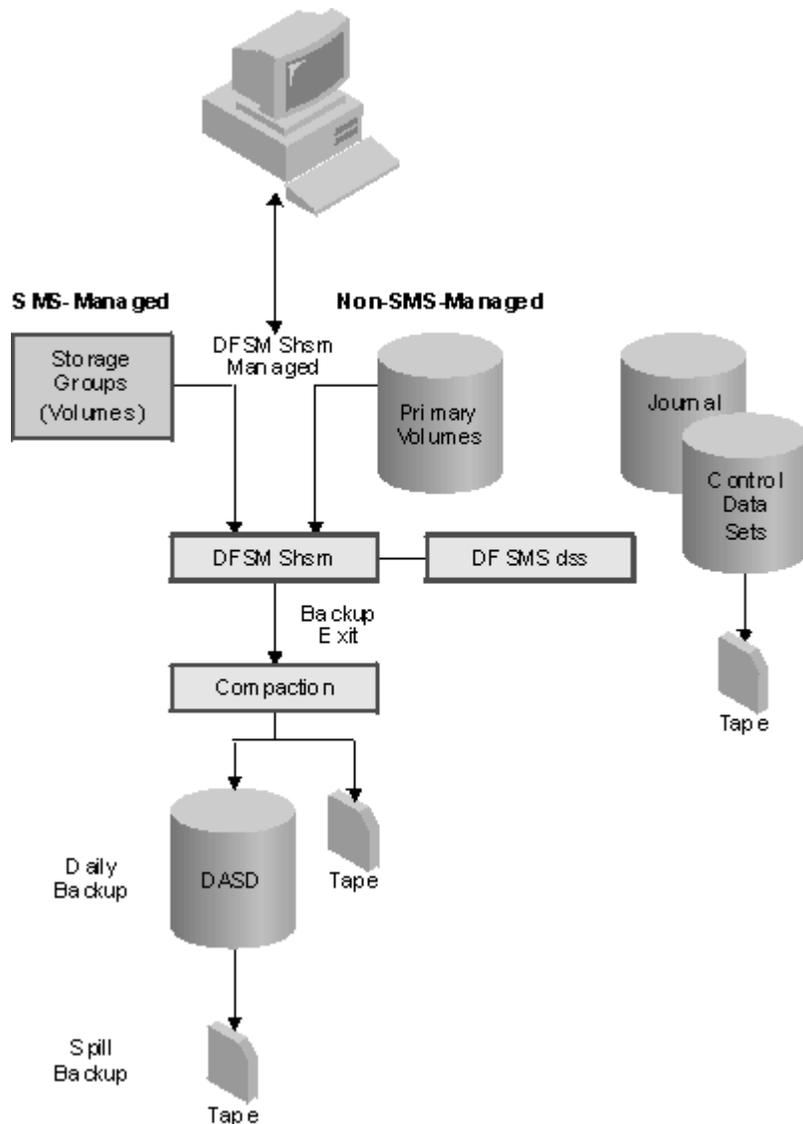
Figure 10: Automatic incremental backup

For system-managed volumes, DFSMShsm uses the storage group automatic backup attribute to determine what volumes to back up, and the management class attributes to determine whether to back up a data set and, if so, how often to back it up. After the data set is backed up, DFSMShsm determines from the management class attributes how many versions to maintain and how long to keep them.

For non-system-managed volumes, DFSMShsm uses the volume attributes to determine what volumes to back up. If DFSMShsm's automatic backup attribute is active for the volume, all data sets requiring backup are backed up.

Performing Inline Backup

You can back up a data set during job processing by invoking DFSMShsm to perform inline backup.

DFSMShsm provides two distinct methods to perform inline backup. Each has its benefits over the other. The first method uses the HBACKDS command in a TSO terminal monitor program (TMP) batch job step. This allows for data set filtering using % and * characters for masking, and provides an option to only perform the backup if the data set was modified since the last backup was taken by DFSMShsm. The second method is to use the ARCINBAK program in a batch job step. Each data set is identified by a unique DD statement and supports the use of JCL name substitution and relative data flow diagram names. The second method can be used while the job has exclusive use of the data set, for example, if it was created by an earlier job step.

You can request inline backup copies by placing the appropriate statements in a job step. The copies are created by DFSMShsm incremental backup during processing. Inline backup writes the backup version on a ML1 DASD volume to minimize unplanned tape mounts. The backup versions can be moved from ML1 to incremental backup volumes automatically during incremental backup processing, or by command.

Using Fast Replication for Backup Versions

Fast replication provides a valuable method for obtaining a point-in-time backup that is managed by DFSMShsm with minimal application outage. Fast replication uses FlashCopy on the Enterprise Storage Server (ESS) or SnapShot on the RAMAC Virtual Array (RVA) to copy the volumes. The use of this function allows for the following benefits:

- You can use an SMS copy pool construct that allows you to define a set of storage groups that are processed collectively for fast replication.
- DFSMShsm manages full volume fast replication backup versions.
- Recovery from a backup version is performed at the volume or copy pool level.
- You can use ISMF panels to implement this function.

Recovering Data Sets

Data set recovery is the process of copying a backup version back to a user DASD volume. Anywhere from one to 64 individual data set recovery tasks can be performed concurrently, as defined by the installation.

A storage administrator can enter a data set recover command to perform the following recovery tasks:

- Recover the latest data set version of any data set implicitly from a dump or an incremental backup volume
- Recover an explicit data set version from a dump or an incremental backup volume

Users can enter data set recover commands to recover only their own data sets, either the latest, or a specific version. The data recovered can come from either dump or incremental backup volumes.

Recovering DASD Volumes

Volume recovery is the process of copying all data back to a user DASD volume.

You can use volume recovery commands to perform the following recovery tasks:

- Restore a volume from the DFSMSdss dump copy with the option to recover individual data sets to the level of the most recent DFSMShsm incremental backup data set versions. This option scratches restored data sets that are now invalid because they:
  - Were already migrated or were scratched after the dump copy was made
  - Are uncataloged but have the same name as a cataloged data set
- Restore the volume only from the DFSMSdss dump copy
- Recover the volume only from the DFSMShsm incremental backup data set versions

The multitasking capabilities of individual data set recovery can speed volume recovery when incremental data set backup versions are used in the recovery process. The incremental backup data sets are scheduled to be processed by data set recovery tasks, rather than being processed by the volume recover task. If more than one volume is being recovered, DFSMShsm schedules the incremental data set recovery requests for one volume and the recovery process can immediately begin on the next volume, lessening the time needed for the total recovery process.

Managing Availability Manually

DFSMShsm provides commands to perform availability operations on volumes and on data sets. The following tasks can be accomplished by command:

Deleting expired backup versions
DFSMShsm can determine expired backup versions of data sets and delete them. DFSMShsm uses the management class attributes to determine which expired backup versions of system-managed data sets to delete. DFSMShsm uses the parameters entered with the command to determine which expired backup versions of non-system-managed data sets to delete.

Duplicating DFSMShsm tapes
Dumps by command and automatic dumps provide a way to store data at an offsite storage facility. The data can be dumped from level 0 and ML1 volumes, but not for backup and ML2 data that was written to tapes. Off site protection of data sets can be done with aggregate backup, volume dumping, TAPECOPY, or by using the duplex tape function.

Replacing DFSMShsm tapes with alternate tapes
If a DFSMShsm original backup or ML2 tape is damaged, the data from that tape can be retrieved from the alternate tape. You can request that DFSMShsm replace its inventory references to the original tape with the references to the alternate tape. The alternate tape becomes the new original tape. Thus, when DFSMShsm refers to the data that was on the original tape, the data is retrieved from the new original tape.

DFSMShsm also provides a function called disaster alternate. Disaster alternate allows for a much faster form of tape replacement than previously existed. Should a true disaster occur, you can enter a command that causes DFSMShsm to flag all existing alternate tapes as disaster alternates. Then, you can use another command to set DFSMShsm to run in disaster mode. When in disaster mode, each data set recall and recover causes an extra check of the volume record to see if a disaster alternate tape should be substituted for the original. The disaster alternate function also allows an installation to test their disaster plan much quicker than was previously possible.

Performing volume dump
During volume dump, DFSMShsm uses a DFSMSdss function to perform a full-volume dump of the entire allocated space on a level 0 or DFSMShsm-owned volume. Volume dumps can only be performed by authorized users.

Requesting Volume Backup
To request a backup of some or all data sets on a DFSMShsm-managed volume, any primary volume, or a user volume that is not managed by DFSMShsm, authorized users can enter a command that backs up an entire volume. Keywords in the command allow you to specify whether to back up every data set on the volume or whether to back up specific data sets. Although this command is available to back up every data set on the volume, using a command to perform a full-volume dump usually provides better performance. Only authorized users can enter a command to back up a level 0 volume.

Requesting data set backup
You can request a backup of a specific data set from a mounted volume. DFSMShsm backs up the data set and temporarily stores it on a ML1 volume. The data set does not have to reside on a volume managed by DFSMShsm.

For SMS-managed data sets, a management class backup attribute determines whether data sets in the management class are allowed to be backed up by command.

DFSMShsm allows up to 64 data set backups per DFSMShsm image. Data sets can be backed up directly to tape. Storage administrators can define whether tape, ML1 DASD, or both, are to be used as target output devices.

Tape takes longer from initial selection to first write than DASD, but is potentially faster in data throughput. DASD takes less time to become available because there is no mount delay, but the throughput is potentially less than it is for tapes.

Authorizing and Protecting DFSMShsm Commands

Because DFSMShsm operates as an MVS-authorized task, it can manage data sets automatically, regardless of their security protection. DFSMShsm allows an installation to control the authorization of commands through the use of either RACF FACILITY class profiles or the AUTH command.

If the RACF FACILITY class is active, DFSMShsm always uses it to protect all DFSMShsm commands. If the RACF FACILITY class is not active, DFSMShsm uses the AUTH command to protect storage administrator commands.

The RACF FACILITY Class Environment

DFSMShsm provides a way to protect all DFSMShsm command access through the use of RACF FACILITY class profiles. An active RACF FACILITY class establishes the security environment.

An individual, such as a security administrator, defines RACF FACILITY class profiles to grant or deny permission to issue individual DFSMShsm commands.

For more information about establishing the RACF FACILITY class environment, see *z/OS DFSMShsm Implementation and Customization Guide*.

The DFSMShsm AUTH Command Environment

Commands within DFSMShsm are classified as *authorized* and *nonauthorized*.

Authorized commands can be issued only by a user specifically authorized by a storage administrator. Generally, authorized commands can affect data sets not owned by the person issuing the command and should, therefore, be limited to only those whom you want to have that level of control.

Nonauthorized commands can be issued by any user, but they generally affect only those data sets for which the user has appropriate security access. Nonauthorized commands are usually issued by system users who want to manage their own data sets with DFSMShsm user commands.

For more information about establishing an AUTH command environment, see *z/OS DFSMShsm Implementation and Customization Guide*.

Controlling Access to Data

DFSMShsm protects data sets from unauthorized access by controlling access to data sets.

Access can be controlled by the use of both passwords and security programs at the same time. However, system-managed data sets are not password protected. If a data set is password protected and security-program protected, DFSMShsm allows access to the data set without checking the password if the security program authorizes the access.

DFSMShsm also provides protection against unauthorized use or deletion of its owned tape volumes.

To provide security program protection, DFSMShsm calls the system authorization facility (SAF) when any unauthorized user enters a command that manipulates a data set or its backup copies.

As an installation option, users can submit batch jobs containing DFSMShsm commands in secure systems without RACF. DFSMShsm retrieves the user ID from the time sharing option (TSO) protected step control block for a TSO batch request and associates it with the request so that authorization can be checked.

DFSMShsm optionally creates a backup profile for the most recent backup version of a cataloged data set if the data set is protected with a RACF discrete data set profile when it is backed up. DFSMShsm maintains only one backup profile for all the backup versions of the cataloged data set. The backup profile is used to recreate the discrete data set profile if it does not exist when the data set is recovered. When all backup versions of a data set are deleted, the related backup profile is also deleted.

DFSMShsm-owned data on DASD is named so that it can be protected by RACF generic profiles.

Protecting Data on Tapes

DFSMShsm provides three basic methods for protecting tape volumes under its control. You can choose more than one tape protection option.

RACF

DFSMShsm protects each tape backup volume, dump tape volume, and tape migration level 2 (ML2) volume with RACF. The tape volume is RACF protected when the tape is first written. DFSMShsm protects the tape volume with RACF by adding the volume to the RACF tape volume set for DFSMShsm. All tape volumes in the RACF tape volume set for DFSMShsm share the same access list and auditing controls. Protection of the tape volume is removed when the tape becomes empty and is returned to scratch status.

Expiration date

If you use expiration date protection, DFSMShsm protects each tape backup volume, dump tape volume, tape ML2 volume, and aggregate tape volume with an expiration date. DFSMShsm places an expiration date of 99365 (means that the date never expires) in the IBM standard data set label 1 unless specified otherwise by a DFSMShsm parameter or installation exit.

Password

If you use password protection, DFSMShsm causes each backup, dump, or migration tape to be password indicated. The password indication is placed in the security byte in the IBM standard data set label 1.

Recommendation: IBM does not recommend the use of data set password protection because it provides less protection than the use of RACF. See *z/OS DFSMSdfp Advanced Services* for a discussion of password protection and the reasons to not use it.

Using Tape

DFSMShsm takes advantage of the single-file format and the compaction algorithms available on cartridge type devices. The single-file format allows for better performance and utilization of cartridges since they can contain hundreds or thousands of data sets on a single tape cartridge. DFSMShsm uses the capacity of the cartridges and provides ways to handle contentions for different data sets residing on the same cartridge.

DFSMShsm creates backup and migration output for tapes in single-file format and supports only cartridge type tape devices for the output. The CDS backup, the dump, and the ABARS functions support both reel type and cartridge type tapes for their functions. Data sets that have been backed up or migrated to reel type tapes, or that are in multifile format, can still be recovered or recalled from those tapes. Reel type tapes can also be recycled by DFSMShsm, with the new tapes being created as cartridge type tapes.

Specifying Tape Utilization

DFSMShsm allows you to specify the desired level of tape utilization (fullness) for a backup or migration tape volume written in single-file format. You can use the full tape volume, meaning that DFSMShsm writes until reaching the physical end of volume (EOV). However, the recommended method is to specify a percentage to indicate how much of the tape should be used before considering it full. DFSMShsm then forces a logical end of volume with this method.

Marking Partially-Filled Tapes as Full

When a backup or migration task ends, the last tape is usually only partially filled. When new processing begins, DFSMShsm selects that last partially-filled tape on which to continue processing. This lets you completely fill your tapes, but it does require a specific tape mount.

DFSMShsm allows you to specify that your partially-filled tapes be marked as full. Then, when new processing begins, DFSMShsm issues a nonspecific tape mount, which is beneficial when using cartridge loaders.

If you are not using the duplex tape function, marking partially-filled tapes as full has another benefit. If you want to copy the migration and backup tapes for the purpose of disaster backup protection, marking a tape full allows it to be copied by the generic TAPECOPY command. A partially-filled tape that is not marked as full is not eligible to be copied. The data on the partially-filled tape would not have a disaster

backup copy until the next time tape migration and backup processing occurred, the tape was filled, and the TAPECOPY command was run.

If you *are* using the duplex tape function, even partially-filled migration and backup tapes have copies. Marking partially-filled tapes as full would still provide the benefit of nonspecific tape mounts.

Each installation should weigh the benefits of marking partially-filled tapes as full against the benefits of utilizing the full storage capacity of the tapes, and decide what works best for them.

Automating DFSMShsm Tape Processing

Both cartridge loaders and automated tape libraries offer automated tape processing with DFSMShsm. The cartridge loaders are manually loaded and can hold up to multiple tape cartridges. An automated tape library is a library where tape cartridges are stored, selected, and loaded onto tape drives without manual intervention. It manages the tape volume itself, not the data sets on the tape volumes.

An automated tape library allows DFSMShsm to perform operations such as recycle, backup, dump, or migration to tape without any manual intervention. Recalls or recovery of data sets or volumes that were migrated, backed up, or dumped to tape can also be done without manual intervention. When DFSMShsm issues a request for one or more tape volumes, the automated tape library locates the requested volumes and automatically loads them on a tape drive. When processing is complete, the cartridges are returned to storage.

Implementing Tape Mount Management

By implementing the tape mount management methodology, you can improve tape utilization and greatly reduce the number of tape mounts required for tape output data sets.

The volume mount analyzer reviews your tape mounts and creates reports that provide you with the information you need to effectively implement the recommended tape mount management strategy.

With tape mount management, you can have your ACS routines redirect the data sets that are targeted to tape and have them written initially to a designated DASD storage group. Later, the data sets can be migrated collectively to ML2 tape, enabling your installation to use the full storage capacity of your tape cartridges. At the same time, instead of having one tape mount for each tape output data set, you have one tape mount per quantity of data equal to the logical capacity of your tape cartridge. Not only is less time spent mounting tapes, or waiting for tape mounts, there is less tape drive contention, which can speed the processing of other jobs requiring tape resources.

For example, if your tape cartridge held an average of 100 output data sets and you were using tape mount management, you would have 10 output tape mounts per 1000 output data sets. If you were not using tape mount management, you would have 1000 output tape mounts per 1000 output data sets. The reduction in tape mounts for your installation could be significant.

Using Storage Groups, Copy Pools, and Volume Pools

DFSMShsm processes system-managed DASD volumes in SMS pool storage groups. A pool storage group is a set of volumes with free space threshold and paths to the systems within the complex. Requests of data sets that have common storage requirements can more easily be satisfied by placing volumes in a storage group.

In a multiple-processing-unit environment, storage groups can be defined to allow DFSMShsm automatic functions to be performed for specific storage groups by specific processing units. This is in contrast to allowing storage groups to be processed by all processing units in a multiple-processing-unit environment.

DFSMShsm processes SMS pool storage groups at the copy pool level. A copy pool is a named set of storage groups. DFSMShsm processes these storage groups collectively by using the fast replication function. You can maintain as many as 85 backup versions for each copy pool. DFSMShsm can recover all of the volumes in a copy pool or an individual volume.

DFSMShsm continues to support volume pools that are not managed by the Storage Management Subsystem. Volume pools depend on the volume from which the data set last migrated. If the volume from which the data set last migrated is a member of a pool, DFSMShsm recalls the data set to any online volume in the pool. If the volume from which the data set migrated is not a member of a pool, DFSMShsm uses its nonpooling algorithms for recalling the data set.

Volume pooling allows the inclusion of individual applications or group volumes under DFSMShsm control. It provides the capability to manage independent pools of volumes according to specific, installation defined requirements. Volume pooling also allows the gradual placement of more volumes into a larger pool, thus benefitting from the increased performance, improved DASD space utilization, and additional flexibility of the larger resource. This gives the installation the opportunity to reevaluate its current pooling structures for possible consolidation, while still taking advantage of the automated space management features of DFSMShsm.

Providing Multiple Address Spaces for DFSMShsm

Multiple address spaces for DFSMShsm allows you to start multiple DFSMShsm hosts in a single z/OS image.

An HSMplex can consist of both single and multiple DFSMShsm-host environments. There are several advantages to starting more than one DFSMShsm host in a z/OS image:

- Less work per address space and less contention between functions.
- Each address space that is doing some part of DFSMShsm's work can have an appropriate MVS dispatching priority for that type of work.
- A larger number of tasks are provided that perform any given DFSMShsm function; for example, migration.
- DFSMShsm functions that operate in more than one address space allow more MIPs that are allocated to DFSMShsm functions.

Chapter 10. Managing Removable Media with DFSMSrmm

DFSMSrmm is an optional element of z/OS and helps you manage your removable media, such as tape cartridges, reels, and optical volumes. DFSMSrmm provides a central online inventory of the resources in your removable media library and in storage locations outside your removable media library.

A removable media library contains all the tape and optical volumes that are available for immediate use and includes the shelves where they reside. A removable media library usually includes other libraries: system-managed libraries and non-system-managed libraries, either automated or manual.

Storage locations are locations outside your removable media library. You can use DFSMSrmm defined storage locations or define your own storage locations. Use the storage locations for managing disaster recovery or vital records or keeping track of volumes sent to other locations for processing.

DFSMSrmm manages your storage within your removable media library at several levels:

Managing Shelves
  Shelves are where you store your tape volumes when they are not in an automated tape library. Use DFSMSrmm to efficiently group your shelves and keep track of what volumes reside on them. For more information, see "Managing Shelves" on page 73.

Managing Volumes
  DFSMSrmm helps you manage the movement and retention of your tape volumes over their full life. For more information, see "Managing Volumes" on page 74.

Managing Data Sets
  DFSMSrmm records information about data sets on the volumes it manages, and then uses that information to validate volumes. DFSMSrmm can also control the retention of data sets. For more information, see "Managing Data Sets" on page 74.

See "Using DFSMSrmm" on page 75 to learn about the tasks that you can perform using DFSMSrmm.

Refer to the following publications for more information about DFSMSrmm:

- *z/OS DFSMSrmm Application Programming Interface*
- *z/OS DFSMSrmm Managing and Using Removable Media*
- *z/OS DFSMSrmm Implementation and Customization Guide*
- *z/OS DFSMSrmm Diagnosis Guide*
- *z/OS DFSMSrmm Reporting*

Managing Storage with DFSMSrmm

DFSMSrmm helps you manage your tape volumes and shelves at your primary site and storage locations by recording information in a DFSMSrmm control data set.

You can manage storage at the shelf, volume, and data set level.

Managing Shelves

DFSMSrmm helps you manage the shelves in your tape library and storage locations, simplifying the tasks of your tape librarian. When you define a new volume in your library, you can request that DFSMSrmm shelf-manage the volume by assigning the volume a place on the shelf. You also have the option to request a specific place for the volume. Your shelves are easier to use when DFSMSrmm manages them in pools. Pools allow you to divide your shelves into logical groups where you can store volumes. For example, you can have a different pool for each system that your installation uses. You can then store the volumes for each system together in the same pool.

You can define shelf space in storage locations. When you move volumes to a storage location where you have defined shelf space, DFSMSrmm checks for available shelf space and then assigns each volume a place on the shelf if you request it. You can also set up DFSMSrmm to reuse shelf space in storage locations.

Managing Volumes

DFSMSrmm manages your volumes over their full life, from initial use to the time they are retired from service. You can define all volumes to DFSMSrmm and use DFSMSrmm to manage them now that you can define volumes that have duplicate volume serial numbers. DFSMSrmm provides functions to control the use of undefined volumes and allows you to identify volumes that should be ignored.

You do not have to define a tape volume to DFSMSrmm to use the volume. Volumes that are used to satisfy nonspecific requests and all system-managed tape volumes must be managed by DFSMSrmm. Volumes in a system-managed tape library are automatically defined to DFSMSrmm upon entry into the system-managed tape library, if the volume is not previously defined.

To enhance the integrity of your volumes, DFSMSrmm identifies volumes with permanent errors and temporary errors. DFSMSrmm identifies tapes with permanent errors as requiring replacement. DFSMSrmm tracks and reports information about volumes with temporary errors. This information helps you identify volumes that should be replaced or removed from the library, and prevents their reuse as scratch volumes.

DFSMSrmm eases your operator's handling of volumes by updating drive displays and operator mount messages to indicate a volume's correct shelf location. DFSMSrmm supports the display of messages on drives that have displays. DFSMSrmm can update write-to-operator messages to show the volume's shelf location.

You can set up DFSMSrmm to issue operator messages to a specific console name associated with a system-managed library. DFSMSrmm then issues messages to the specific console and through standard message route codes.

For control of volume movements and retention, you can define policies using DFSMSrmm vital record specifications for both volumes and data sets. For example, you can define a vital record specification that ignores the volume expiration for the volume on which a data set resides. DFSMSrmm then uses the vital record specification retention criteria instead of the user specified retention period to retain the data set. This allows you to control the time a data set is retained before it is considered for release. Use DFSMSrmm vital record specifications to control data set retention including whether or not user specified values are honored.

Use DFSMSrmm with products like DFSMShsm, OAM, and Tivoli® Storage Manager to extend tape management functions and tape integrity checking to tape volumes that are owned by these products. DFSMSrmm ensures that the correct tape volume is always mounted and that only scratch tapes are used for nonspecific tape volume requests. DFSMSrmm can also control the movement of DFSMShsm tape volumes that are selected as vital records, such as alternate volumes that are required for disaster recovery.

DFSMSrmm supports all the tape label types supported by DFSMSdfp and records the tape label type for each DFSMSrmm-managed volume. DFSMSrmm does not support volumes with nonstandard labels. DFSMSrmm also supports the use of bypass label processing (BLP). See "Magnetic Tape Volumes and Libraries" on page 39 for more information on tape labels. DFSMSrmm provides support for ISO/ANSI Version 3 and 4 tape labels.

Managing Data Sets

When a tape volume is used on the system, DFSMSrmm automatically records information, such as expiration dates, about each data set on the volume. DFSMSrmm uses these expiration dates along with the retention and movement policies you define to determine when to release the volume.

DFSMSrmm provides options that you can use to retain data sets and volumes:

• By Cycles and By Days Cycle

- By Elapsed Days
- By Extra Days
- By Day Last Referenced
- While the data set is cataloged
- Until the expiration date is reached
- By Data Set Open or Abend Status
- By Job Name and Data Set Name
- By Generic Job Name Mask and Generic Data Set Name Mask
- By Specific Volume Serial Number
- By Generic Volume Serial Number
- By Specific Date

DFSMSrmm prevents volumes that contain unexpired data from being used as scratch volumes. With DFSMSrmm, you can manage volumes and data sets under the following conditions:

- Volumes that contain just one data set
- Volumes that contain multiple data sets
- Data sets that span more than one volume
- Data set collections that span multiple volumes

DFSMSrmm supports generic data set names as filter criteria for searching the control data set, making it easier to create lists of resources.

DFSMSrmm provides the same support for dates as DFSMSdfp. DFSMSrmm records all dates using a 4-digit year and also allows you to specify dates using the 4-digit year.

Using DFSMSrmm

Your tape librarian and storage administration group are the focal point for using DFSMSrmm. DFSMSrmm helps these and other users perform the tasks required for managing tape volumes. You can control which users can access the data defined to DFSMSrmm, as well as the level of access they have. DFSMSrmm also helps you create valuable reports and provides utilities to maintain your tape library.

Performing User Tasks

Although DFSMSrmm automates many of the tasks required to manage your removable media library, there are some tasks you can perform manually:

- Defining the resources in your removable media library, such as volumes, data sets, and shelves, to the DFSMSrmm control data set
- Updating this same information
- Registering software product volumes
- Obtaining information about these resources
- Requesting scratch volumes for private use
- Releasing volumes
- Requesting electronic notification when a volume expires
- Confirming volume movements and release actions

You can request these tasks using the DFSMSrmm ISPF dialog, the DFSMSrmm application programming interface, or the TSO/E command set that DFSMSrmm provides. Generally you use the DFSMSrmm ISPF dialog to perform DFSMSrmm tasks. You might want to use the TSO/E commands in procedures you write. You can use the dialog and command set in the foreground. You can also submit requests for batch processing.

Controlling Access to DFSMSrmm Tasks and Data

DFSMSrmm uses the system authorization facility (SAF) to authorize the use of DFSMSrmm commands and functions to further secure your DFSMSrmm resources. For DFSMSrmm command authorization decisions, you can use volume ownership, or volume and data set authorization. When RACF name hiding is in use, DFSMSrmm makes extra calls via SAF to ensure that users can only access DFSMSrmm resource information for volumes and data sets that they are authorized to use. You can use discrete Resource Access Control Facility (RACF) TAPEVOL profiles and the RACF TAPEDSN option to ensure that only authorized users can access data on volumes that are defined to DFSMSrmm.

Creating Reports

DFSMSrmm provides report facilities and sample reports to help you report on DFSMSrmm information. Reporting facilities include an ISPF application that enables quick and easy creation of several types of reports. You can use the system management facilities (SMF) to create records when information in DFSMSrmm has been updated or when confidential volumes have been accessed. Use the SMF records to create reports describing access to designated volumes and data sets.

You can create inventory and movement reports for volumes managed by DFSMSrmm. The inventory reports help you audit the content of your library and storage locations. The movement reports identify volume movement through your library and storage locations. You can use information from the DFSMSrmm control data set as input to create your own reports using DFSORT or DFSORT's ICETOOL utility. DFSORT lets you use symbols for fields and constants in DFSORT and ICETOOL control statements. This makes it easy to use DFSORT's ICETOOL for DFSMSrmm reporting. You can use DFSORT symbols for the DFSMSrmm SMF records, extract file records, and activity file records instead of using field offsets or data types to create reports.

Maintaining Your Removable Media Inventory

DFSMSrmm provides several utilities you can use to keep the information in your tape library current and to perform various tasks related to maintaining your tape media. Run these utilities using your existing scheduling facilities, such as the IBM Tivoli Workload Scheduler for z/OS, so you can choose the best time and sequence to perform regularly required tasks.

DFSMSrmm utilities perform the following tasks:

- Maintain the integrity of the DFSMSrmm control data set. You can verify the records in the control data set, back up the data set, and restore it.
- Control and track volumes and data sets that are managed by vital record specifications.
- Manage the release of volumes that have reached their expiration date and that are no longer retained by vital record specifications.
- Define and verify the volume movements between your library and storage locations.
- Erase and initialize volumes.

Appendix A. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

If you experience difficulty with the accessibility of any z/OS information, send a detailed email message to mhvrcfs@us.ibm.com.

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element
> The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element
> The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

* indicates an optional syntax element that is repeatable
> The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

> **Notes:**
>
> 1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
>
> 2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.

3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line `6.1+ data area`, you must include at least one data area. If you hear the lines `2+`, `2 HOST`, and `2 STATE`, you know that you must include `HOST`, `STATE`, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for the Knowledge Centers. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual.

This glossary includes 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, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- 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). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See:
> This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

3480
> The machine number of the IBM 3480 Magnetic Tape Subsystem, which is a group of magnetic tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A11 and A22, and two drive models, B11 and B22.

3490
> IBM 3490 Magnetic Tape Subsystem. A group of magnetic tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A01 and A02, and two drive models, D31 and D32, in which the control unit function and tape drives are integrated.

3490E
> IBM 3490E Magnetic Tape Subsystem. A group of enhanced capability tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A10 and A20, and two drive models, B10 and B20. In addition, there are two models, D41 and D42, in which the control unit function and tape drives are integrated.

3590
> The IBM TotalStorage Enterprise Tape System 3590 is capable of coexisting with 3490 and 3490E devices in the IBM TotalStorage Enterprise Automated Tape Library (3495) and with the 3490E devices in the IBM TotalStorage Enterprise Automated Tape Library (3494), or as a stand-alone tape drive. The 3590 has a built-in control unit. This device supports the IBM High Performance Cartridge System Tape and the IBM Extended High Performance Cartridge System Tape media.

ABARS
> Aggregate backup and recovery support in DFSMShsm.

ABEND
> Abnormal end of task. End of a task, a job, or a subsystem because of an error condition that cannot be resolved by recovery facilities while the task is performed.

access method
> A mainframe data management routine that moves data between storage and an I/O device in response to requests made by a program.

The part of the distributed data management architecture which accepts commands to access and process the records of a file.

access method services
A multifunction service program that manages VSAM and non-VSAM data sets, as well as catalogs. Access method services provides the following functions:

- defines and allocates space for data sets and catalogs
- converts indexed-sequential data sets to key-sequenced data sets
- modifies data set attributes in the catalog
- reorganizes data sets
- facilitates data portability among operating systems
- creates backup copies of data sets
- assists in making inaccessible data sets accessible
- lists the records of data sets and catalogs
- defines and builds alternate indexes

ACDS
See *active control data set.*

ACS
See *automatic class selection.*

activate
Establishing a new storage management policy for the Storage Management Subsystem (SMS) complex by loading the contents of a source control data set (SCDS) into SMS address space storage and into an active control data set (ACDS), or loading the contents of an existing ACDS into SMS address-space storage.

active configuration
The configuration currently used by SMS to control the managed storage in the installation.

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, communications data set*.

active data
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.

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*.

activity log
In DFSMShsm, a SYSOUT or DASD-type data set used to record activity and errors that occurred during DFSMShsm processing.

address
The unique identifier assigned to each device or workstation connected to a network.

address space
The complete range of addresses in memory available to a computer program.

Advanced Program-to-Program Communications (APPC)
An implementation of the Systems Network Architecture (SNA) logical unit (LU) 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

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 data sets
In aggregate backup and recovery processing, data sets that have been defined in an aggregate group as being related.

aggregate group
A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

aggregate recovery
The process of recovering a user-defined group of data sets that were backed up by aggregate backup.

AIX
Advanced Interactive Executive. In a VSAM environment, alternate index.

alias
An alternative name for a catalog, a non-VSAM data set, or a member of a partitioned data set (PDS) or partitioned data set extended (PDSE).

allocation
Generically, the entire process of obtaining a volume and unit of external storage, and setting aside space on that storage for a data set.

The process of connecting a program to a data set or devices.

alternate index
In systems with VSAM, a key-sequenced data set containing index entries organized by the alternate keys of its associated base data records. It provides an alternate means of locating records in the data component of a cluster on which the alternate index is based.

alternate tape volumes
In DFSMShsm, copies of original tape volumes created during tape copy processing. The volumes can either be stored on-site or off-site for use later in the event of a disaster. During the tape replace processing, these volumes can replace the original volumes that may be lost.

ANSI
American National Standards Institute.

ANSI
American National Standards Institute.

APF
Authorized program facility.

API
Application programming interface.

APPC
Advanced program-to-program communication.

application programming interface (API)
A formally defined programming language interface between an IBM system control program or a licensed program and the user of a program.

asynchronous operation
A type of operation in which the extended remote copy (XRC) function copies updates to the secondary volume of an XRC pair at some time after the primary volume is updated. See also *synchronous operation.*

AUTH
The DFSMShsm command used to identify an authorized user who can issue DFSMShsm system programmer and storage administrator commands.

automated tape library data server
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. Contrast with *manual tape library*. See also *tape library*.

automatic backup
In DFSMShsm, the process of automatically copying data sets from primary storage volumes or migration volumes to backup volumes.

In OAM, the process of automatically copying a primary copy of an object from disk, or an optical or tape volume to a backup volume contained in an object backup storage group.

automatic class selection (ACS) routine
A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

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 interval migration
In DFSMShsm, automatic migration that occurs periodically when a threshold level of occupancy is reached or exceeded on a DFSMShsm-managed volume during a specified time interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

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.

availability
For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

backup
The process of creating a copy of a data set or object to be used in case of accidental loss.

backup control data set (BCDS)
In DFSMShsm, a VSAM key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

backup profile
In DFSMShsm, a RACF discrete data set profile associated with the backup version of a cataloged data set that is protected by a RACF discrete data set profile.

backup volume
A volume managed by DFSMShsm to which backup versions of data sets are written.

backup-while-open (BWO)
This makes a backup copy of a data set while the data set is open for update. The backup copy can contain partial updates.

base cluster
In systems with VSAM, a key-sequenced or entry-sequenced data set over which one or more alternate indexes are built.

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 manages.

basic direct access method (BDAM)
An access method used to directly retrieve or update particular blocks of a data set on a direct access device.

basic partitioned access method (BPAM)
An access method that can be applied to create program libraries in direct access storage for convenient storage and retrieval of programs.

basic sequential access method (BSAM)
An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

BCDS
See *Backup control data set.*

BDAM
See *Basic direct access method.*

block count
The number of data blocks on a magnetic tape volume.

block size
The number of data elements in a block; usually specified in bytes.

A measure of the size of a block, usually specified in units such as records, words, computer words, or characters.

Synonymous with block length.

Synonymous with physical record size.

BLP
Bypass label processing.

BPAM
See *Basic partitioned access method.*

BSAM
See *Basic sequential access method.*

BTLS
Basic Tape Library Support.

buffer
A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another.

byte stream
A simple sequence of bytes stored in a stream file.

cache fast write
A storage control capability in which the data is written directly to cache without using nonvolatile 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*.

cache set
A parameter on storage class and defined in the base configuration information that maps a logical name to a set of CF cache structure names.

capacity planning
The process of forecasting and calculating the appropriate amount of physical computing resources required to accommodate an expected workload.

catalog
A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors.

A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics. (A) (ISO)

To enter information about a file or a library into a catalog. (A) (ISO)

The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set.

To include the volume identification of a data set in the catalog.

See *VSAM master catalog, VSAM user catalog.*

Catalog Search Interface
An application programming interface (API) to the catalog accessible from assembler and high-level languages. As an alternative to LISTCAT, it allows tailoring of output, provides additional information not provided by LISTCAT, while requiring less I/O than LISTCAT, because of using generic locates.

CDS
See *Control data set.*

CDSQ serialization
A technique that serializes DFSMShsm control data sets (CDSs) with a global enqueue product, such as global resource serialization (GRS).

CDSR serialization
A technique that serializes DFSMShsm CDSs with volume reserves.

CF
See *coupling facility*.

CFRM
Coupling facility resource management.

checkpoint
A designated point in the program at which information about the job is collected and recorded in a separate checkpoint data set.

CI
See *Control interval.*

CICS
Customer Information Control System.

CICSVR
Customer Information Control System VSAM Recovery, a forward recovery utility, which can perform forward recovery for DFSMStvs and others as well as for CICS.

class transition
An event that brings about change to an object's service-level criteria, causing OAM to invoke ACS routines to assign a new storage class or management class to the object.

client
A user.

A consumer of resources or services.

A functional unit that receives shared services from a server.

A system that is dependent on a server to provide it with programs or access to programs.

On a network, the computer requesting services or data from another computer.

CLIST
TSO command list.

cluster
A data component and an index component in a VSAM key-sequenced data set; or a data component alone in a VSAM entry-sequenced data set. See also *storage cluster.*

command line
On a display screen, a display line usually at the bottom of the screen in which only commands can be entered.

common recall queue (CRQ)
A single recall queue shared by multiple DFSMShsm hosts that enables the recall workload to be balanced across each of those hosts.

compaction
See *improved data recording capability.*

compatibility mode
The mode of running SMS in which no more than eight names—representing systems, system groups, or both—are supported in the SMS configuration.

compress

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.

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.

compressed format

A particular type of extended-format data set specified with the (COMPACTION) parameter of data class. VSAM can compress individual records in a compressed-format data set. SAM can compress individual blocks in a compressed-format data set. See *compress*.

concurrent copy

A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

configuration

The arrangement of a computer system as defined by the characteristics of its functional units. See also *SMS configuration*.

connectivity

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.

In a system-managed storage environment, the system status of volumes and storage groups.

construct

One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

control area (CA)

A group of control intervals used as a unit for formatting a data set before adding records to it.

In a key-sequenced data set, the set of control intervals, pointed to by a sequence-set index record, that is used by VSAM for distributing free space and for placing a sequence-set index record adjacent to its data.

control data set (CDS)

With respect to SMS, a VSAM linear data set containing configurational, operational, or communication information. SMS uses three types of control data sets: the source control data set (SCDS), the active control data set (ACDS), and the communication data set (COMMDS).

In DFSMSrmm, all changes made to the inventory, such as adding or deleting volumes, are recorded in the control data set.

control file

In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the catalog, allocation, volume, and related information necessary to perform aggregate recovery.

control interval (CI)

A fixed-length area of auxiliary storage space in which VSAM stores records. It is the unit of information (an integer multiple of block size) transmitted to or from auxiliary storage by VSAM.

control unit

A hardware device that controls the reading, writing, or displaying of data at one or more input/output devices. See also *storage control.*

copy pool

A collection of names of storage groups that are processed collectively for fast replication operations.

coupled extended remote copy (CXRC)

A technique that supports synchronous copy operations in large environments that have an expanded number of primary-storage controls and direct access storage device (DASD) volumes; this number is in excess of those supported by a single datamover configuration. Installations may have configurations consisting of thousands of volumes in multiple XRC sessions; coordination among the sessions ensures that all volumes can be recovered to a consistent time. CXRC greatly expands upon the ability of XRC to provide remote disaster-recovery protection across a sysplex.

coupling facility (CF)
    The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

Customer Information Control System (CICS)
    An IBM licensed program that provides online transaction processing services and management for critical business applications. CICS runs on many IBM and non-IBM platforms (from the desktop to the mainframe) and is used in various types of networks that range in size from a few terminals to many thousands of terminals. The CICS application programming interface (API) enables programmers to port applications among the hardware and software platforms on which CICS is available. Each product in the CICS family can interface with the other products in the CICS family, thus enabling interproduct communication.

CXRC
    See *coupled extended remote copy*.

DADSM
    See *direct access device space management*.

DASD
    See *Direct access storage device.*

DASD fast write
    An extended function of some models of the IBM 3990 Storage Control 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. Contrast with *cache fast write*. See also *dynamic cache management*.

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
    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.

Data Facility Storage Management Subsystem (DFSMS)
    An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

Data Facility Storage Management Subsystem Transactional VSAM Services (DFSMStvs)
    An optional feature of DFSMS for running batch VSAM processing concurrently with CICS online transactions. DFSMStvs users can run multiple batch jobs and online transactions against VSAM data, in data sets defined as recoverable, with concurrent updates.

data record
    A collection of items of information from the standpoint of its use in an application, as a user supplies it to the system storage. Contrast with *index record*.

data security
    Prevention of access to or use of data or programs without authorization. As used in this publication, the safety of data from unauthorized use, theft, or purposeful destruction.

data set
    In DFSMS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and

sometimes are used interchangeably. See also *file*. In z/OS UNIX environments, the terms *data set* and *file* have quite distinct meanings.

data set collection

A group of data sets which are intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking.

data set control block

A control block in the VTOC that describes data set characteristics.

data set organization

The type of arrangement of data in a data set. Examples are sequential organization or partitioned organization.

data set stacking

The function used to place several data sets on the same tape volume or set of tape volumes. It increases the efficiency of tape media usage and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data sets together on a minimum number of tape volumes, which is useful when sending data offsite.

Db2

DATABASE 2.

DDM

See *Distributed Data Management Architecture*.

Distributed Data Management Architecture (DDM)

Distributed Data Management Architecture (DDM) offers a vocabulary and a set of rules for sharing and accessing data among like and unlike computer systems. DDM includes a set of standardized file models for keyed, relative record, sequential, and stream data. It allows users and applications to access data without concern for the location or format of the data.

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 base configuration, it identifies an esoteric (such as SYSDA) or generic (such as 3390) device name. If a user omits the UNIT parameter on the JCL or the dynamic allocation equivalent, SMS applies the default unit if the data set has a disposition of MOD or NEW and is *not* system-managed.

device

This term is used interchangeably with unit. For a disk or tape, a unit on which a volume may be mounted. For example, a tape drive is a device; a tape cartridge is a volume. Device also applies to other types of equipment, such as a card reader or a channel-to-channel (CTC) adapter.

device category

A storage device classification used by SMS. The device categories are as follows SMS-managed DASD, SMS-managed tape, non-SMS-managed DASD non-SMS-managed tape.

device management

The task of defining input and output devices to the operating system, and then controlling the operation of these devices.

Device Support Facilities (ICKDSF)

A program used for initialization of DASD volumes and track recovery.

DFM

See *Distributed FileManager*.

DFSMS

See *Data Facility Storage Management Subsystem.*

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 z/OS, this function is provided by DFSMS, DFSORT, and RACF. See also *system-managed storage*.

DFSMSdfp
A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss
A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm
A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm control data set
In DFSMShsm, one of three VSAM key-sequenced data sets that contain records used in DFSMShsm processing. See also *backup control data set, migration control data set, offline control data set*.

DFSMShsm log
In DFSMShsm, a pair of sequential data sets, X and Y, containing a chronological list of transactions and statistics occurring in DFSMShsm.

DFSMShsm-managed volume
A primary storage volume, which is defined to DFSMShsm but which does not belong to a storage group.

A volume in a storage group, which is using DFSMShsm automatic dump, migration, or backup services. Contrast with *system-managed volume, DFSMSrmm-managed volume*.

DFSMS Optimizer Feature
A DFSMS feature that provides an analysis and reporting capability for SMS and non-SMS environments.

DFSMSrmm
A DFSMS functional component or base element of z/OS, that manages removable media.

DFSMSrmm-managed volume
A tape volume that is defined to DFSMSrmm. Contrast with *system-managed volume, DFSMShsm-managed volume*.

DFSMStvs
See *Data Facility Storage Management Subsystem Transactional VSAM Services.*

DFSORT
See *Data Facility Sort.*

direct access device space management (DADSM)
A collection of subroutines that manages space on disk volumes. The subroutines are: Create, Scratch, Extend, and Partial Release.

direct access storage device (DASD)
A device in which time is effectively independent of the location of the data.

directory
A file that maps the names of other directories and files to their locations.

disaster backup
A means to protect a computing system complex against data loss in the event of a disaster.

disaster recovery
A procedure for copying and storing an installation's essential business data in a secure location, and for recovering that data in the event of a catastrophic problem. Compare with *vital records*.

discrete profile
A RACF profile that contains security information about a specific data set on a particular volume.

Distributed FileManager

Distributed FileManager is an implementation of target (server) support as defined by Distributed Data Management Architecture (DDM). DDM permits systems in an extended enterprise that have DDM source (client) capability to access file data on a DDM target MVS system. See also *source, target, extended enterprise*.

Distributed Data Management Architecture (DDM)

Distributed Data Management Architecture (DDM) offers a vocabulary and a set of rules for sharing and accessing data among like and unlike computer systems. DDM includes a set of standardized file models for keyed, relative record, sequential, and stream data. It allows users and applications to access data without concern for the location or format of the data.

DIV

Data in Virtual.

DSCB

Data set control block.

DSORG

Data set organization. It is specified in the JCL as "DSORG=".

dual copy

A high availability function made possible by nonvolatile storage in some models of the IBM 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.

dump class

A set of characteristics that describes how volume dumps are managed by DFSMShsm.

dynamic allocation

Assignment of system resources to a program when the program is performed rather than when it is loaded main storage.

EA

See *extended addressability*.

Enterprise Storage Server (ESS)

An IBM product that provides an intelligent disk-storage subsystem across an enterprise. ESS provides data storage for SCSI and fibre-channel attached host systems and for System/390 and zSeries host systems.

EOV

End-of-volume.

erase-on-scratch

A RACF and DFP/XA function that overwrites the space occupied by a data set when the data set is scratched from a DASD device supported by z/OS/XA.

ESCON

Enterprise System Connection.

ESDS

See *Entry-sequenced data set*.

ESS

Enterprise Storage Server.

EXCP

See *Execute channel program.*

execute channel program (EXCP)

A macro used to access a data set without specifying the organization.

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.

In DFSMSrmm, all volumes have an expiration date or retention period set for them either by vital record specification policy, by user-specified JCL when writing a data set to the volume, or by an installation default. When a volume reaches its expiration date or retention period, it becomes eligible for release.

export
The operation to remove one or more logical volumes from a virtual tape server library. First, the list of logical volumes to export must be written on an export list volume and then the export operation itself must be initiated.

extended addressability
The ability to create and access a VSAM data set that is greater than 4 GB in size. Extended addressability data sets must be allocated with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

extended format
The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different. Data sets in extended format can be striped or compressed. Data in an extended format VSAM KSDS can be compressed. See also *striped data set, compressed format*.

extended remote copy
Extended Remote Copy (XRC) is a technique involving both the DFSMS host and the I/O Subsystem that keeps a "real time" copy of designated data at another location. Updates to the primary center are replicated at the secondary center asynchronously.

extents
A continuous space on a direct access storage volume, occupied by or reserved for a particular data set, data space, or file.

facility class profile
A way that RACF protects all DFSMShsm command access. An active RACF facility class establishes the security environment.

fast replication
The functions that create an instant data copy. The Enterprise Storage Server (ESS) FlashCopy function is an example of fast replication.

FCP
Fibre Channel Protocol.

Fibre Channel Protocol
An evolving ANSI interconnection standard for high-speed data transfer among computers and storage devices defined by the Fibre Channel Protocol (FCP). At the physical level, data is transmitted serially over pairs of point-to-point links between FC ports. With appropriate components, data can be transferred at 100 MB/sec up over a distance of 10 km between FC ports. The ports can be configured in ring and switched interconnect topologies which allow each port to exchange data with any other port that is part of the same fabric.

file
A collection of information treated as a unit. In z/OS non-UNIX and non-z/OS UNIX environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

file system
In the z/OS UNIX environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See *z/OS file system* and *HFS data set*.

file system
In the z/OS UNIX environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

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.

FlashCopy

A point-in-time copy services function that can quickly copy data from a source location to a target location.

free space

Space reserved within the control intervals of a key-sequenced data set for inserting new records into the data set in key sequence or for lengthening records already there; also, whole control intervals reserved in a control area for the same purpose.

GB

See *gigabyte*.

GDG

See *generation data group.*

GDS

See *generation data set*.

giga (G)

The information-industry meaning depends upon the context:

1. G = 1 073 741 824(2^{30}) for real and virtual storage

2. G = 1 000 000 000 for disk storage capacity (e.g. 4 GB fixed disk)

3. G = 1 000 000 000 for transmission rates

generation data group (GDG)

A collection of historically related non-VSAM data sets that are arranged in chronological order; each data set is a generation data set.

generation data set

One generation of a generation data group.

generic profile

A RACF profile that contains security information about multiple data sets, users, or resources that may have similar characteristics and require a similar level of protection.

global resource serialization (GRS)

A component of z/OS used for serializing use of system resources and for converting hardware reserves on DASD volumes to data set enqueues.

GRS complex (GRSplex)

One or more z/OS images that share a common global resource serialization policy in either a ring or star configuration.

group

With respect to partitioned data sets, a member and the member's aliases that exist in a PDS or PDSE, or in an unloaded PDSE.

A collection of users who can share access authorities for protected resources.

GRS

See *global resource serialization.*

guaranteed space

A storage class attribute indicating the space is to be preallocated when a data set is created. If you specify explicit volume serial numbers, SMS honors them. If space to satisfy the allocation is not available on the user-specified volumes, the allocation fails.

hardware configuration definition (HCD)

An interactive interface in z/OS that enables an installation to define hardware configurations from a single point of control.

HCD

See *hardware configuration definition.*

hierarchical file system (HFS) data set

A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using z/OS UNIX System Services. See also *file system.*

Hiperspace
A high performance virtual storage space of up to two gigabytes. Unlike an address space, a Hiperspace contains only user data and does not contain system control blocks or common areas; code does not execute in a Hiperspace. Unlike a data space, data in a Hiperspace cannot be referenced directly; data must be moved to an address space in blocks of 4KB before they can be processed. The 4K blocks can be backed by expanded storage or auxiliary storage, but never by virtual storage. The Hiperspace used by VSAM is only backed by expanded storage. See also *Hiperspace buffer*.

host
A computer connected to a network that provides an access method to that network. A host provides end-user services.

HSM complex (HSMplex)
One or more z/OS images running DFSMShsm that share a common set of control data sets (MCDS, BCDS, OCDS, and Journal).

ICETOOL
DFSORT's multipurpose data processing and reporting utility.

import
The operation to enter previously exported logical volumes residing on a stacked volume into a virtual tape server library. First, the list of logical volumes to import must be written on an import list volume and the stacked volumes must be entered, then, the import operation itself must be initiated.

IMS/VS
Information Management System/Virtual Storage.

inactive configuration
A configuration contained in an SCDS. A configuration that is not currently being used by the Storage Management Subsystem.

inactive data
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.

In tape mount management, data that is written once and never used again. The majority of this data is point-in-time backups.

Objects infrequently accessed by users and eligible to be moved to the optical library or shelf.

Contrast with *active data*.

incremental backup
A process in which data sets are backed up only if they have changed since their last backup.

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.

initial program load (IPL)
The initialization procedure that causes an operating system to commence operation.

The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction.

The process of leading system programs and preparing a system to run jobs.

Synonymous with system restart, system startup.

inline backup
The process of copying a specific data set to a migration level 1 volume from a batch environment. This process allows you to back up data sets in the middle of a job.

installation exit
The means specifically described in an IBM software products documentation by which an IBM software product may be modified a customers system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or

subroutines to an IBM software product, for the purpose of modifying (including extending) the functions of the IBM software product.

instruction data set
In aggregate backup and recovery processing, a data set that contains instructions, commands, or any data the aggregate backup site defines as needed for aggregate recovery at the recovery site.

Interactive Storage Management Facility (ISMF)
The interactive interface of DFSMS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF)
An interactive base for ISMF.

interval migration
In DFSMShsm, automatic migration that occurs when a threshold level of occupancy is reached or exceeded on a DFSMShsm-managed volume, during a specified time interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

I/O
Input/output.

IPL
See *initial program load.*

ISAM
Indexed sequential access method.

ISMF
See *Interactive Storage Management Facility.*

ISO
International Organization for Standardization.

ISO/ANSI
When referring to magnetic tape labels and file structure, any tape that conforms to certain standards established by the ISO and ANSI. Tapes are sometimes called:

- **Version 3 tapes** Conforms to ISO 1001–1979 level 4, and ANSI X3.27–1978 level 4

- **Version 4 tapes** Conforms to ISO 1001–1986(E) LEVEL 4, ANSI X3.27–1987 level 4

ISPF
See *Interactive System Productivity Facility.*

ISPF/PDF
Interactive Systems Productivity Facility/Program Development Facility.

JCL
See *job control language*.

JES3
A z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

KB
See *kilobyte.*

kilo (K)
The information-industry meaning depends upon the context:

1. K = 1024(2^{10}) for real and virtual storage

2. K = 1000 for disk storage capacity (e.g. 4 KB fixed disk)

3. K = 1000 for transmission rates

key-sequenced data set (KSDS)
A VSAM data set whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in the data set in key sequence because of free space allocated in the data set. Relative byte addresses of records can change because of control interval or control area splits.

KSDS
Key-sequenced data set.

LDS
See *linear data set*.

level 0 volume
A volume that contains data sets directly accessible by the user. The volume might be either DFSMShsm-managed or non-DFSMShsm-managed.

like device
Pertaining to (DASD) devices with identical geometry: that is, the same number of bytes per track, the same number of tracks per cylinder, and the same number of cylinders per actuator.

linear data set (LDS)
In VSAM, a named linear string of data, stored in such a way that it can be retrieved or updated in 4096-byte units.

logical storage
With respect to data, the attributes that describe the data and its usage, as opposed to the physical location of the data.

logical storage subsystem
A collection of addresses that are associated with the same logical subsystem. Used internally by ESS to manage a set of logical volumes which are associated with an individual device adapter. For example, a physical ESS subsystem might be partitioned into multiple logical storage subsystems.

LSR
See *local shared resources*.

management class
A collection of management attributes, defined by the storage administrator. Management class is used to control: the release of allocated but unused space; the retention, migration, and back up of data sets; the retention and back up of aggregate groups, and the retention, back up, and class transition of objects.

manual tape library
Installation-defined set of tape drives defined as a logical unit together with the set of system-managed volumes which can be mounted on the drives.

master catalog
A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

MB
Megabyte.

MDS
Main device scheduling.

mega (M)
The information-industry meaning depends upon the context:

1. M = 1,048,576(2^{20}) for real and virtual storage
2. M = 1,000,000 for disk storage capacity (e.g. 4 MB fixed disk)
3. M = 1,000,000 for transmission rates

migration

The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also *migration level 1, migration level 2*.

migration control data set (MCDS)

In DFSMShsm, a VSAM key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMShsm.

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, 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, migration level 1*.

ML1

See *Migration level 1*.

ML2

See *Migration level 2*.

multiple extended remote copy (MXRC)

An enhancement to XRC that allows you to run up to five XRC sessions within a single LPAR.

MVS

Multiple Virtual Storage.

name hiding

A characteristic of a security mechanism that prevents unauthorized users from obtaining names of objects that the user is not authorized to know about.

NaviQuest

A DFSMSdfp component for implementing, verifying, and maintaining your DFSMS SMS environment in batch mode. It provides batch testing and reporting capabilities that can be used to automatically create test cases in bulk, run many other storage management tasks in batch mode, and use supplied ACS code fragments as models when creating your own ACS routines.

Network File System

A protocol developed by Sun Microsystems, Incorporated, that allows users to directly access files on other systems in a network.

NFS

z/OS Network File System.

NL

No label.

NSL

Nonstandard label.

OAM

see *object access method*.

OAM-managed volumes

Optical or tape volumes controlled by the object access method (OAM).

OAM Storage Management Component (OSMC)

Where objects should be stored, manages object movement within the object storage hierarchy and manages expiration attributes based on the installation storage management policy.

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 storage group
A type of storage group that contains objects on DASD, tape, or optical volumes. See also *storage group*.

object storage hierarchy
A 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.*

OCDS
See *offline control data set.*

offline control data set (OCDS)
In DFSMShsm, a VSAM key-sequenced set that contains information about tape backup volumes and tape migration level 2 volumes.

OLTP
Online transaction processing.

optical disk drive
The mechanism used to seek, read, and write data on an optical disk. An optical disk drive can be operator-accessible or library-resident.

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.*

outboard
Indicates that a function or action is performed by the tape library, not by the host system software.

outboard policy management
A method of managing tape volumes that defines actions for volume policies at the library instead of at the host.

Parallel Sysplex
A sysplex that uses one or more coupling facilities.

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 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.

PDS
See *Partitioned data set.*

PDSE
See *partitioned data set extended.*

peer-to-peer remote copy (PPRC)
A hardware-based remote copy option that provides a synchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

peer-to-peer VTS library
Multiple VTS subsystems and their associated tape libraries coupled together to form one subsystem and library image to the host.

performance
A measurement of the amount of work a product can produce with a given amount of resources.

In a system-managed storage 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*.

physical storage
With respect to data, the actual space on a storage device that is to contain data.

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*.

PPRC
See *Peer-to-peer remote copy.*

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, migration level 2*.

program management
The task of preparing programs for execution, storing the programs, load modules, or program objects in program libraries, and executing them on the operating system.

PSF
Print Services Facility

PSP
Preventive service planning.

PTF
Program temporary fix.

QSAM
See *queued sequential access method*.

queued sequential access method
An extended version of the basic sequential access method (BSAM). Input data blocks awaiting processing or output data blocks awaiting transfer to auxiliary storage are queued on the system to minimize delays in I/O operations.

RACF
See *Resource Access Control Facility*.

RBA
See *relative byte address.*

REA
RAMAC Electronic Array Storage.

recall
The process of moving a migrated data set from a level 1 or level 2 volume to a DFSMShsm-managed volume or to a volume not managed by DFSMShsm.

record-level sharing (RLS)
See *VSAM Record Level Sharing (VSAM RLS)*.

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.

recycle process
A DFSMShsm process that, based on the percentage of valid data on a tape backup or migration level 2 volume, copies all valid data on the tape to a tape spill backup or migration level 2 volume.

Redundant Array of Independent Disks (RAID)
A disk subsystem architecture that combines two or more physical disk storage devices into a single logical device to achieve data redundancy.

relative byte address (RBA)
The displacement (expressed as a fullword binary integer) of a data record or a control interval from the beginning of the data set to which it belongs, independent of the manner in which the data set is stored.

relative record data set (RRDS)
A type of VSAM data set whose records have fixed or variable lengths, and are accessed by relative record number.

remote copy
A storage-based disaster recovery and workload migration function that can copy data in real time to remote location. Two options of remote copy are available. See *peer-to-peer remote copy* and *extended remote copy.*

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 is included in z/OS Security Server and is also available as a separate program for the z/OS and VM environments. RACF provides access control by identifying and verifying the users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

restore
Synonym for recover.

Restructured Extended Executor (REXX) Language
A general-purpose, high-level programming language, particularly suitable for EXEC procedures or programs for personal computing.

RETAIN®
REmote Technical Assistance and Information Network.

RLS
Record-level sharing. See *VSAM record-level sharing (VSAM RLS).*

RRDS
Relative record data set.

RSA
RAMAC Scalable Array Storage.

RVA
RAMAC Virtual Array Storage.

SAF
System authorization facility.

SCDS
See *Source control data set.*

SDSP
Small data set packing.

sequential data set
A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Contrast with *direct data set.*

server
A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server.

On a network, the computer that contains the data or provides the facilities to be accessed by other computers in the network.

A program that handles protocol, queuing, routing, and other tasks necessary for data transfer between devices in a computer system.

service level (Storage Management Subsystem)
A set of logical characteristics of storage required by a Storage Management Subsystem-managed data set (for example, performance, security, availability).

sharing control data set
A VSAM linear data set that contains information DFSMSdfp needs to ensure the integrity of the data sharing environment.

SHCDS
Sharing control data set.

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 for storage of removable media.

In DFSMSrmm, a shelf location is defined in the removable media library by a rack number, and in a storage location, it is defined by a bin number. See also *rack number, bin number*.

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.

SMF
See *System Management Facilities.*

SMS
See *Storage Management Subsystem.*

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 to SMS in the SMS base configuration.

SMS control data set
A VSAM linear data set containing configurational, operational, or communications information that guides the execution of the Storage Management Subsystem. See also *source control data set, active control data set, communications data set*.

SMS class
A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or backup and retention (management class) needs.

SMS-managed data set
A data set that has been assigned a storage class.

SNA
Systems Network Architecture.

SnapShot
A function of the RAMAC Virtual Array (RVA) that allows an instantaneous copy to be made of data sets using DFSMS software.

source
Source is the term used in Distributed Data Management Architecture (DDM) to refer to the platform that originates a request for remote data. Source is also known as client. Contrast with *target*.

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, communications data set*.

space management
In DFSMShsm, the process of managing aged data sets on DFSMShsm-managed and migration volumes. The three types of space management are: migration, deletion, and retirement.

sphere
A VSAM cluster with one or more associated alternate indexes and paths. The VSAM cluster (sometimes called the base cluster), alternate indexes, and paths are sometimes referred to as sphere components.

SQL
Structured Query Language.

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 collection 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 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, dummy storage group*.

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, 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 activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

Storage Management Subsystem (SMS)
A DFSMS 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
In DFSMS, the portion of a striped data set, such as an extended format data set, that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striped data set
In DFSMS, an extended-format data set consisting of two or more stripes. SMS determines the number of stripes to use based on the value of the SUSTAINED DATA RATE in the storage class.

Striped data sets can take advantage of the sequential data striping access technique. See*stripe, striping.*

striping

A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

synchronous operation

A type of operation in which the remote copy PPRC function copies updates to the secondary volume of a PPRC pair at the same time that the primary volume is updated. Contrast with *asynchronous operation.*

sysplex

A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

system data

The data sets required by z/OS or its subsystems for initialization and control.

system group

All systems that are part of the same Parallel Sysplex and are running the Storage Management Subsystem with the same configuration, minus any systems in the Parallel Sysplex that are explicitly defined in the SMS configuration.

system-managed buffering for VSAM

A facility available for system-managed extended-format VSAM data sets in which DFSMSdfp determines the type of buffer management technique along with the number of buffers to use, based on data set and application specifications.

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, and space to applications. See also *system-managed storage 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 *DFSMShsm-managed volume, DFSMSrmm-managed volume.*

system management facilities (SMF)

A component of z/OS 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.

Systems Network Architecture (SNA)

The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through the networks and also operational sequences for controlling the configuration and operation of networks.

TB

Terabyte.

tera (T)

The information-industry meaning depends upon the context:

1. $T = 1\ 099\ 511\ 627\ 776 (2^{40})$ for real and virtual storage

2. $T = 1\ 000\ 000\ 000\ 000$ for disk storage capacity (e.g. 4 TB of DASD storage)

3. $T = 1\ 000\ 000\ 000\ 000$ for transmission rates

tape librarian
The person who manages the tape library. This person is a specialized storage administrator.

tape library
A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See also *system-managed tape library, automated tape library data server*.

Tape Library Dataserver
A hardware device that maintains the tape inventory that is associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, 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 a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume
A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

target
Target is the term used in Distributed Data Management Architecture (DDM) to refer to the platform that fulfills a request for remote data. Target is also known as server. Contrast with *source*.

TCP/IP
See *Transmission Control Protocol/Internet Protocol*.

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, as a percentage of occupied tracks versus total tracks. The *low migration threshold* is used during primary space management and interval migration to determine when to stop processing data. The *high allocation threshold* is used to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high threshold are selected over volumes that meet or exceed the high threshold value.

TMM
See *tape mount management*.

TMP
Terminal Monitor Program.

transaction processing
Processing transactions as they are received by the computer. Also called online or realtime systems transaction processing. Data sets are updated as soon as transactions are entered at terminals or received over communications lines. It also implies that confirmations are returned to the sender. If you save receipts in a shoebox and add them up at the end of the year for taxes, that's batch processing. However, if you buy something and immediately add the amount to a running total, that's transaction processing.

Transmission Control Protocol/Internet Protocol (TCP/IP)
The two fundamental protocols of the Internet protocol suite. The abbreviation TCP/IP is frequently used to refer to this protocol suite. TCP/IP provides for the reliable transfer of data, while IP transmits the data through the network in the form of datagrams. Users can send mail, transfer files across the network, or execute commands on other systems.

TSO/E
   TSO/Extensions.

UCB
   See *unit control block*.

UIM
   User interaction module.

unit affinity
   Requests that the system allocate different data sets residing on different removable volumes to the same device during execution of the step to reduce the total number of tape drives required to execute the step. Explicit unit affinity is specified by coding the UNIT=AFF JCL keyword on a DD statement. Implicit unit affinity exists when a DD statement requests more volumes than devices.

UNIX
   A highly portable operating system originally developed by Bell Laboratories that features multiprogramming in a multi-user environment. UNIX is implemented in the C language. UNIX was originally developed for use on minicomputers but has been adapted on mainframes and microcomputers. It is especially suitable for multiprocessor, graphics, and vector-processing systems.

unlike device
   A DASD device with a different number of bytes per track and tracks per cylinder, or both.

user catalog
   An optional catalog used in the same way as the master catalog and pointed to by the master catalog. It lessens the contention for the master catalog and facilitates volume portability.

user group
   A group of users in an installation who represent a single department or function within the organization.

validate
   To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

VIO
   See *virtual input/output (VIO) storage group*.

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*.

virtual storage access method (VSAM)
   An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by the relative-record number.

virtual tape server (VTS)
   This subsystem, integrated into the IBM TotalStorage Enterprise Automated Tape Library (3494) or the IBM TotalStorage Enterprise Automated Tape Library (3495), combines the random access and high performance characteristics of DASD with outboard hierarchical storage management and virtual tape devices and tape volumes.

vital records
   A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare 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, tape volume*.

volume mount analyzer
A program that helps you analyze your current tape environment. With tape mount management, you can identify data sets that can be redirected to the DASD buffer for management using SMS facilities.

volume serial number (VOLSER)
An identification number in a volume label that is assigned when a volume is prepared for use on the system. For standard label volumes, the volume serial number is the VOL1 label of the volume. For no label volumes, the volume serial number is the name the user assigns to the volume.

In DFSMSrmm, volume serial numbers do not have to match rack numbers.

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.
- "Non-system-managed" indicates that the volume does not contain any system-managed data sets and has not been initialized as system-managed.

volume table of contents (VTOC)
A table on a direct access volume that describes the location, size and other characteristics of each data set on the volume. Each entry is a data set control block (DSCB).

VRS
Vital record specification.

VRRDS
Variable-length relative-record data set.

VSAM
See *virtual storage access method.*

VSAM record-level sharing (VSAM RLS)
An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the z/OS Coupling Facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM sphere
The base cluster of a VSAM data set and its associated alternate indexes.

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 a catalog. See also *basic catalog structure.*

VTOC
Volume table of contents.

VTS
See *virtual tape server.*

VVDS
VSAM volume data set.

WTO
Write-to-operator.

XRC
See *extended remote copy.*

zFS
See *z/OS File System.*

z/OS File System (zFS)
A type of file system that resides in a Virtual Storage Access Method (VSAM), linear, data set. The data and interfaces are compliant with the distributed file system (DFS), which is part of the distributed computing environment (DCE). DCE is defined by the Open Software Foundation (OSF).

z/OS
z/OS is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system.

z/OS Network File System
A base element of z/OS, that allows remote access to z/OS host processor data from workstations, personal computers, or any other system on a TCP/IP network that is using client software for the Network File System protocol.

z/OS UNIX System Services (z/OS UNIX)
The set of functions provided by the SHELL and UTILITIES, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the z/OS operating system that allow users to write and run application programs that conform to UNIX standards.

Index

D

Transactional VSAM Services *(continued)*
    VSAM record-level sharing 30
transactions 1
TSO/E
    DFSMSrmm subcommands 39
    ISMF 17

U

UIM (user interaction module) 51
UNIX files
    accessing data 2, 27
    hierarchical file system 27
    using access methods 31
unlike devices, moving data between 46
unused space, using DFSMSdss to reduce 47
user catalog 28
user data sets, reblocking 60
user interface
    ISPF 77
    TSO/E 77
using remote copy
    disaster recovery 13
utility
    data management 33
    removable media management 76

V

V2R3 xiv
variable-length records 25
VARY SMS command 23
VIO (virtual I/O) 28
virtual storage access method (VSAM)
    accessing UNIX files 27
    data set types 25
    Transactional VSAM Services 4
    VIO data sets 28
Virtual Tape Server
    description 9
    managing with DFSMSrmm 4
    peer-to-peer single library image 10
virtual tape servers, devices, and volumes 9
VOLCAT 28, 32
volume
    backup 55
    backup command 67
    conversion 3
    defining cache preference level 10
    fast subsequent migration 60
    level 0 54
    management with DFSMSrmm 4
    ML1 (migration level 1) 55
    ML2 (migration level 2) 55
    pooling 70
    reindexing while online 29
    restore and recover 66
    spill backup 55
    support 40
volume catalog 28
volume mount analyzer 8, 40, 70
VSAM (virtual storage access method)
    access method services 32

VSAM (virtual storage access method) *(continued)*
    accessing UNIX files 27
    backup-while-open 13
    CICS VSAM Recovery 29
    data set types 25
    description 31
    Hiperbatch 29
    Hiperspace 29
    record-level sharing (RLS) 25, 29
VSAM volume data set (VVDS) 28
VTOC (volume table of contents)
    catalogs 28
    description 29
VVDS (VSAM volume data set) 28

X

XRC function of remote copy 13
XRC Performance Monitor 13

Z

z/OS DataAgent 44
z/OS File System 27
z/OS Network File System 2
z/OS operating system, overview 1
z/OS Security Server 34
z/OS UNIX System Services
    description 27
    overview 2
    using access methods 31

IBM.