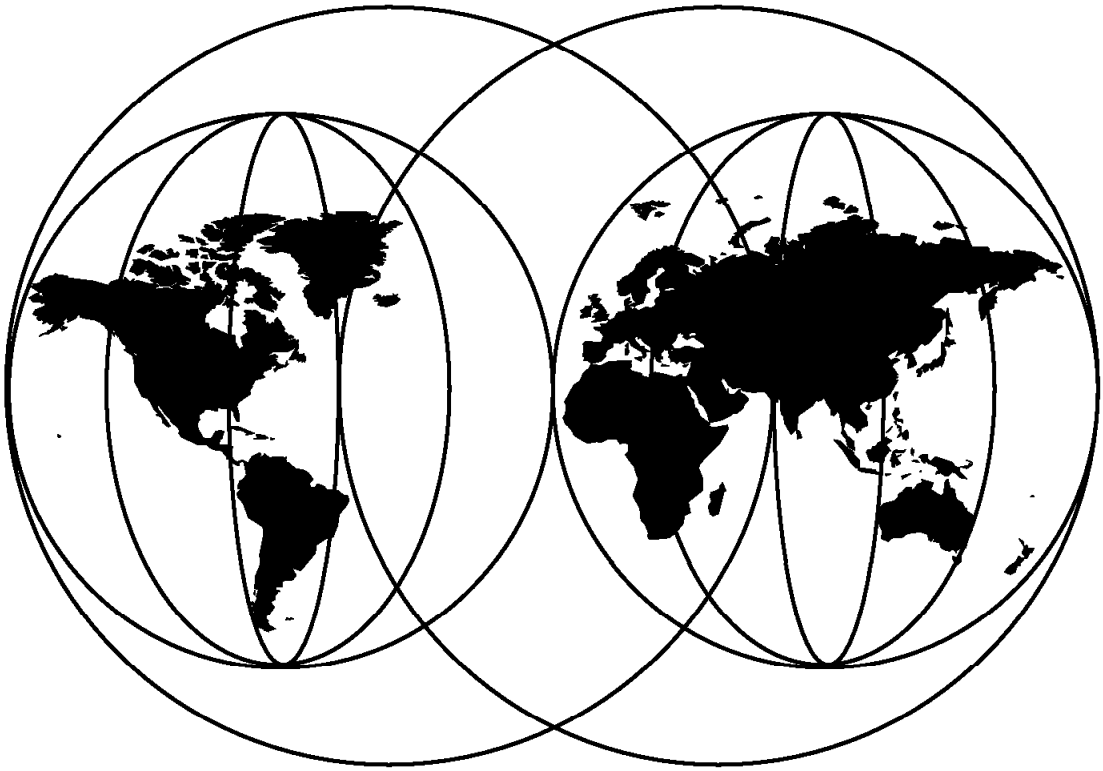
DFSMShsm Primer

Mary Lovelace, Jose Dovidauskas, Jon Tate, Andrew Bradley



International Technical Support Organization

http://www.redbooks.ibm.com

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see "How to Get ITSO Redbooks" at the back of this book for ordering instructions.

SG24-5272-00

International Technical Support Organization

SG24-5272-00

DFSMShsm Primer

December 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 261.

First Edition (December 1998)

This edition applies to Version 1, Release 4 of DFSMS/MVS, Program Number 5695-DF1 for use with the OS/390 Operating System

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

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

Contents

Figures

Tables

Preface

DFSMShsm provides storage management, and availability management in a storage device hierarchy in both system managed, and non-system managed storage environments. This redbook provides storage administrators with DFSMShsm implementation and customization information at the DFSMShsm Version 1 Release 4 level. Hints and tips on the daily operation, monitoring, and tuning of DFSMShsm are included.

Those implementing DFSMShsm for the first time will find valuable information for customizing and exploiting the functions of DFSMShsm. Experienced persons will find this book can be used as an update to the latest DFSMShsm functions, and how those functions can be implemented in an existing DFSMShsm environment.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization San Jose Center.

Mary Lovelace is a Software Engineer at the International Technical Support Organization, San Jose Center. Before joining the ITSO, she worked in the Education Support Center in Poughkeepsie, New York developing large systems and storage product education for EMEA. Mary has more than 20 years of experience with IBM in large systems and storage product education, system engineering and consultancy, marketing support, and system programming.

Andrew Bradley is a Storage Systems Specialist at IBM Global Services in Australia. He has 8 years of experience in providing management and first level support for IBM hardware and software products. He works at the Southern Region Data Centre in Ballarat, Victoria, and specialises in the Storage Services Group project support team providing hardware and software implementation for OS/390 delivery. His expertise is in DFSMS/MVS and storage related hardware.

Jose Dovidauskas is a Senior Products and Systems Specialist in Brazil. He has 14 years of experience in IBM storage software products. He has worked at IBM for 22 years. His areas of expertise include DFSMS/MVS, DASD and tape.

Jon Tate is a Senior Support Specialist in the United Kingdom. He has 12 years of experience in IBM storage software and management, services, and support. Jon′s areas of expertise include DFSMShsm, DFSMSdss, DFSMSdfp, DFSMSopt, DASD, and tape. Jon currently works in the IBM Technical Support Center providing MVS and Storage Level 2 Support for PSS North Region.

Thanks to the following people for their invaluable contributions to this project:

From the Storage Systems Division - Tucson

Ed Baker

Tony Pearson

Jerry Pence

Heather Otto

From the International Technical Support Organization, Raleigh Center

Bob Haimowitz

Thanks are also due to the many other collaborators and reviewers who contributed to this book. I especially want to acknowledge our technical editor, Maggie Cutler, whose work substantially improved the quality of this book.

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 271 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Chapter 1. DFSMShsm Introduction

DFSMShsm is a component of the DFSMS/MVS product that automatically performs space management and availability management in a storage device hierarchy. DFSMShsm ensures that space is available on your DASD volumes so you can extend data sets and allocate new ones. DFSMShsm makes sure that backup copies of your data sets are always available in case your working copies are lost or corrupted. The aggregate backup and recovery support (ABARS) function of DFSMShsm is designed for use in disaster recovery. ABARS performs data backup and recovery processes on a predefined set of data called an *aggregate*.

DFSMShsm is one of five components that, when added together, create a single, integrated software package for all of your installation storage management needs.

The five components of DFSMS/MVS and their functions are:

- DFSMSdfp - Provides storage, data, program, and device management functions through the storage management subsystem (SMS)
- DFSMSrmm - Provides tape management functions for removable media such as tape cartridges and 3420 reels
- DFSMSdss - Provides data movement, copy, backup, and space management functions in a batch environment
- DFSMSopt - Provides monitoring and reporting capabilities for multiple or single system environments
- DFSMShsm - Provides backup, recovery, migration, and space management functions with optimum automation capabilities

DFSMShsm is a policy-driven solution to storage management, removing the requirement for batch jobs to perform backup, migration, or space retrieval functions. DFSMShsm works by rules that you can apply and have set but which are also dynamically adjustable to allow the flexibility required in today′s constantly changing environments.

Its flexibility allows you to manage your storage at the data set level, the device level, or even the device pool level. DFSMShsm provides the means to manage every data set from the time of its creation until the time its last backup is no longer required.

DFSMShsm works closely with SMS, which is part of DFSMS/MVS. SMS changes the storage management approach from user-managed volumes to SMS-managed data sets residing in SMS-managed storage groups. The system, rather than the user, determines data placement and handles data backup, movement, space, and security.

In this redbook we describe how DFSMShsm interacts with all of the DFSMS/MVS components, including SMS, and provide details and techniques to guide you to a successful implementation.

1.1 SMS and DFSMShsm Relationship

SMS allows you to centrally control and direct data set allocation within your system. With SMS you can define performance goals and data availability requirements, create model data definitions for data sets, and automate data backup, using classes and groups. The SMS classes and groups are customized by the storage administrator based on the installation environment and storage policies. The SMS classes and group are:

Data class A list of allocation attributes for data sets (for example, logical record length and record format). The data class simplifies and standardizes data set creation.

Storage class A list of storage performance and availability service requests. The storage class contains space, availability, and performance attributes, such as response time and cache requirements, for data sets.

Management class A list of backup, retention, and migration attributes for data sets.

Storage group A group of one or more DASD volumes that SMS uses for data set allocation.

SMS and DFSMShsm work together, using the attributes specified in the management class, to provide space and availability management at the data set level.

Automatic class selection (ACS) routines are written by the storage administrator. They automatically assign SMS classes and groups to data sets, database data, and objects. Data allocations are processed through ACS routines. ACS routines allow you to enforce installation standards for data allocation and override user specifications for data, storage, and management classes and requests for specific DASD volumes.

You can create up to four ACS routines in an SMS configuration, one for each type of SMS class and one for storage groups. For system-managed data sets the storage group ACS routine is required because it is not possible to explicitly specify storage groups. The other ACS routines are optional. For objects, the storage group, storage class, and management class ACS routines are required. For tape, the storage group, storage class, and data class ACS routines are required.

Figure 1 on page 3 shows the order in which ACS routines are processed. Data becomes system-managed if the storage class routine assigns a storage class to the data. If the routine does not assign a storage class to the data, the data cannot reside on a system-managed volume. The exception is when a specific system-managed tape is specified; then the data set is allocated on system-managed tape.

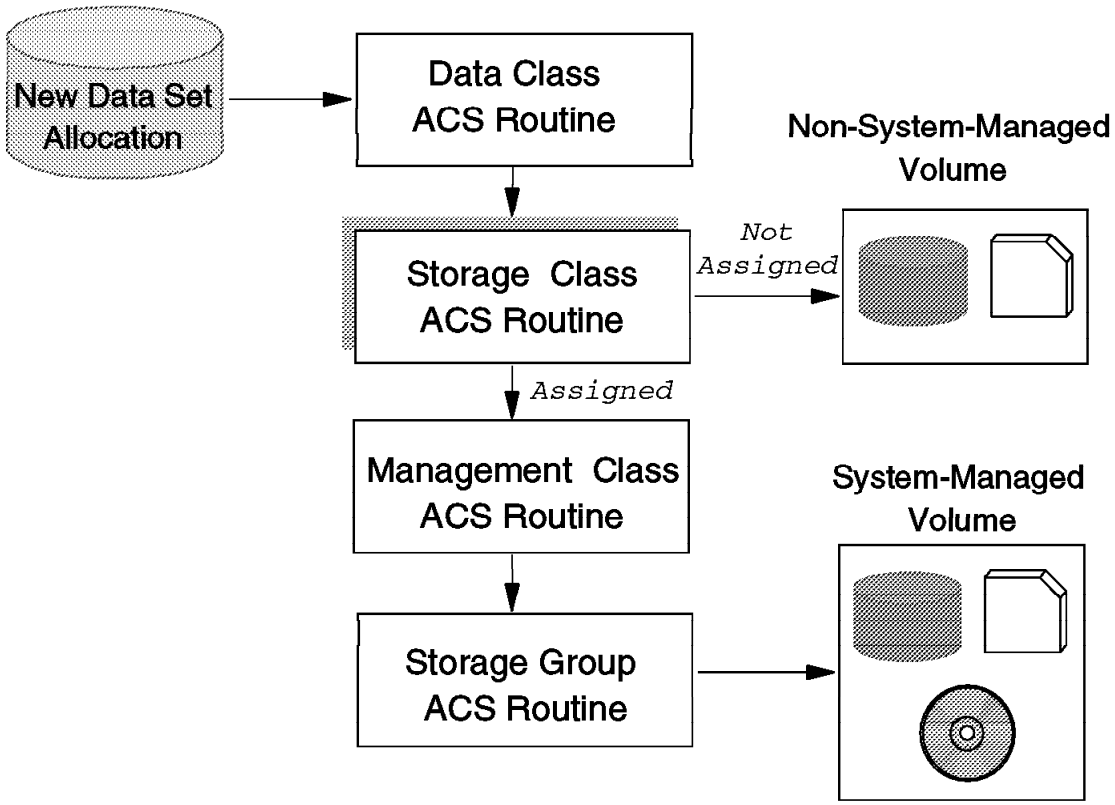


Figure 1. ACS Routine Processing Order

1.2 Space Management

A major function of DFSMShsm is space management. It ensures that only active data occupies the space on a DASD. The major activities of space management include automatic deletion of obsolete data from the DASD, and moving data that is not used frequently from expensive DASD to less expensive DASD and tape volumes.

Data movement from expensive volumes to less expensive storage is called *migration*. Data can be migrated from level 0 volumes that users can access to migration level one (ML1) volumes. The ML1 volumes must be DASD volumes. Data can also be migrated to migration level two (ML2) volumes, which are usually tape volumes. Although ML2 volumes can be DASD, SMS only supports ML2 tape.

When the data is needed again, DFSMShsm returns it to the user's control. This process is called *recall* and is usually processed automatically by DFSMShsm Extra recall is possible by issuing an HRECALL command from your time sharing option (TSO) session or batch job. The ACS routines ultimately determine the volume to which the data set will be recalled.

Space management parameters control how DFSMShsm makes space available on level 0 volumes. Space management is specified as a combination of parameters in the management classes and storage groups. In planning for space management, you want to strike a balance between having enough DASD space available for new data set allocations and having frequent recalls of

migrated data sets. The management class attributes apply on a data set basis so that different data sets on a volume can be migrated based on different criteria.

There are two windows for automatic space management processes: primary and secondary.

1.2.1 Automatic Primary Space Management

Automatic primary space management migrates data sets from level 0 volumes to ML1 volumes or directly to ML2 tapes.

The main function of primary space management main function is to move your system and user data away from your DFSMShsm-managed volumes to tape, or temporarily to ML1 DASD, which provides faster recall capability. This process maintains sufficient capacity on your DFSMShsm-managed volumes to enable operations such as batch processing to continue without interruption or space failure.

Primary space management also performs other important functions on your DFSMShsm-managed volumes such as releasing unused or overallocated space and deleting old or temporary data sets.

DFSMShsm-managed volumes are usually defined in SMS storage groups. Non-SMS-managed volumes can be defined directly to DFSMShsm. DFSMShsm performs space management on both SMS-managed and non-SMS-managed volumes.

1.2.2 Automatic Secondary Space Management

DFSMShsm secondary space management is the function that moves data sets (under control of the management class) from ML1 to ML2 volumes and processes the migration cleanup function. Migration cleanup deletes expired data sets, erases deleted data from small data set packing (SDSP) data sets, and deletes unwanted records from the DFSMShsm migration control data set (MCDS).

Note: The MCDS is a VSAM key-sequenced data set (KSDS) containing all records for each migrated data set on your system.

Secondary space management is usually run before primary space management. It creates the available capacity on your ML1 DASD, which primary space management then uses.

1.2.3 Interval Space Management

DFSMShsm interval space management ensures on an hourly basis that a specified amount of space is available on DFSMShsm-managed volumes. Interval space management only runs against storage groups that have this function enabled through SMS. It is recommended for high usage pools whose volume contents are considered temporary and can be deleted or migrated to tape as soon as possible.

1.3 Availability Management

The other major function that DFSMShsm provides is availability management. Availability management enables a user to easily recover a lost or damaged data set and a storage administrator to easily recover a damaged volume at the most current level possible. This is done through the use of automatic full volume dump supplemented with automatic incremental backup.

Using this methodology, the storage administrator can recover a complete volume by restoring the volume from the full volume dump, then correcting the volume for later activity by the automatic inclusion of changes taken from incremental backups. A DFSMShsm-authorized user can issue one RECOVER command that is used to request both a volume restore and an incremental volume recovery.

The inline backup function allows you to back up data sets in a batch environment. You can back up data sets in the middle of a job in a batch environment. Inline backup writes the backup version on an ML1 volume to minimize unplanned tape mounts. The backup version is subsequently moved to the backup volume during secondary space management.

ABARS provides for disaster recovery. The ABARS functions require the specification of aggregate group and selection data set names to encompass all of the data sets required for running critical applications. You can then issue a single command to back up (ABACKUP) or recover (ARECOVER) all of the data sets in each aggregate.

1.4 DFSMShsm Tape Processing

DFSMShsm uses tapes for functions such as backup, migration, dump, tape copy, recycle of tape backup or ML2 volumes, and ABARS processing. You implement a DFSMShsm tape processing environment by specifying SETSYS commands in the DFSMShsm PARMLIB member ARCCMDxx.

The tape environment is determined by the definition of the library environment (tape library or nonlibrary), tape management policy, device management policy, and performance management policy. You can also define an SMS-managed tape environment (includes a tape management library) or a non-SMS-managed tape environment (a nonlibrary environment).

DFSMShsm uses tape scratch pools for output functions (for example, backup and dump). You can define a global scratch pool or a specific scratch pool. A global scratch pool is a repository of empty scratch tapes for use by anyone. A specific scratch pool is a repository of empty tapes whose use is restricted to a specific user or set of users. Global scratch pools are recommended because mount requests can be responded to more quickly and easily than when tapes reside in a specific scratch pool. Using a global scratch pool you can easily take advantage of automatic cartridge loaders, reducing the tape mount wait time. Global scratch pools also enable use of a tape management product like DFSMSrmm.

The life cycle of a tape is also determined by the SETSYS commands you specify in the DFSMShsm ARCCMDxx member of PARMLIB. The SETSYS commands manage the tape as it enters a scratch pool, is inventoried as active data by

DFSMShsm, is recycled (if it is a backup or migration tape), and is finally returned to the scratch pool.

In this redbook we describe how to define a tape processing environment including tape scratch pools.

Chapter 2. Implementation

The implementation of DFSMShsm involves several activities. Although some of the implementation activities are not directly related to the product, they are required in order to merge the new functions with the existing data management environment.

The purpose of the implementation process is to select and implement options in such a way that the resultant customized product can be used effectively. It is important to consider the modifications that must be applied to the existing system, for example, RACF and DFSMS/MVS as prerequisites before starting any implementation activity.

In this chapter we describe our approach to DFSMShsm implementation. The information is presented in the order that should be followed during this process. Details on setting up and executing test cases are also presented whenever applicable.

The main steps to complete a basic DFSMShsm implementation are:

1. Define the DFSMShsm data sets to DFSMSdfp.
2. Protect the environment with RACF.
3. Establish tape volume management strategy.
4. Install the required start procedures.
5. Define a set of parameters that control the way DFSMShsm manages the data.
6. Run a system test to certify the correctness of the installation.
7. Tune parameters after “steady” state is achieved.

It should be noted that this chapter deals with the installation of DFSMShsm in a SMS-managed environment.

2.1 SMS Considerations

The implementation of DFSMShsm in an existing SMS-managed environment requires modifications to the ACS routines and, probably, the definition of some SMS constructs.

DFSMShsm data can be divided into two groups:

- User and application data. This group includes all the data that DFSMShsm manages according to the parameters coded in the SMS management classes. Some DFSMShsm system data sets are also included in this group.
- DFSMShsm-owned data. The data in this group belongs to DFSMShsm. SMS must be updated so that it recognizes the DFSMShsm data and excludes it from SMS management.

During operation, DFSMShsm writes some information in a group of data sets that includes control data sets (CDSs), activity logs, the journal, DFSMShsm logs, and problem determination aids (PDAs). Figure 2 on page 8 shows the DFSMShsm data sets and their relationship to the different product functions.

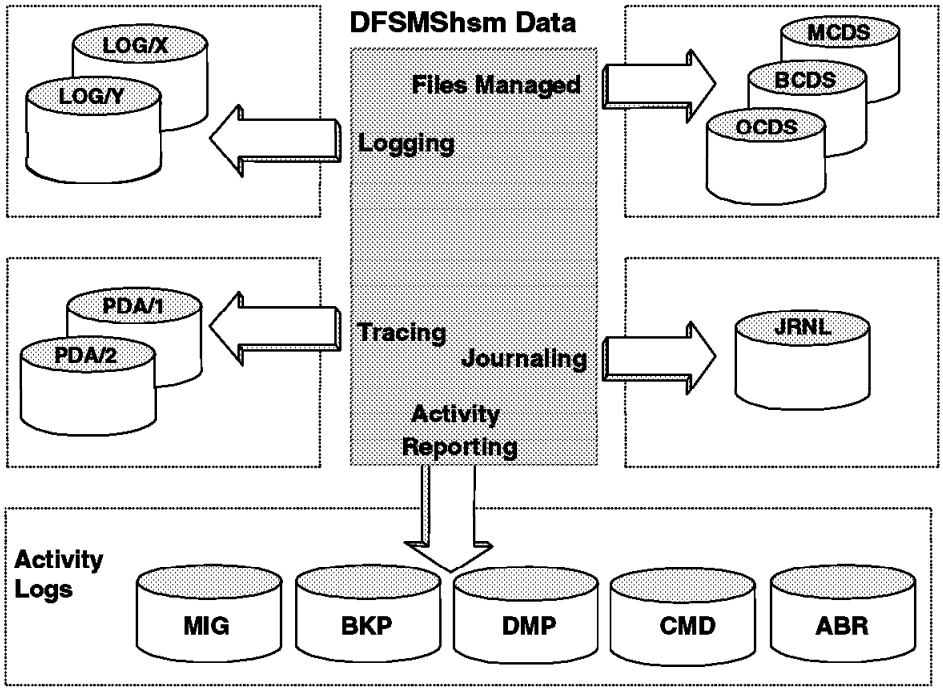


Figure 2. DFSMShsm Data Sets

2.1.1 ACS Routines

You must update your set of ACS routines in order to introduce SMS support for the DFSMShsm data sets.

The definition of DFSMShsm data sets to SMS through the ACS routines is done by filtering on the data set names. DFSMShsm provides a structured approach to data set naming in the following ways:

- CDSs and journal data set names are chosen by the system programmer when DFSMShsm is implemented (as defined in the DFSMShsm starter set job). The name should comply with installation standards.
- DFSMShsm-owned data as migrated or backup copies are identified by a common high-level qualifier (defined to DFSMShsm at implementation), and by a combination of DFSMShsm generated qualifiers.
- DFSMShsm logs and trace data set names are defined by the system programmer at implementation time.
- Activity log data set names are defined by DFSMShsm using a naming structure with a unique high-level qualifier, HSMACT, followed by four additional qualifiers that have a fixed format.
- CDS backup data set names are either defined according to installation standards or defaulted by DFSMShsm in an easily identifiable structure.

Figure 97 on page 240, Figure 98 on page 248, and Figure 99 on page 255 provide some examples of the ACS code used to manage DFSMShsm data. The system programmer or the storage administrator must merge this information into the existing code.

We have defined a high-level qualifier of HSM for our DFSMShsm data sets, which is reflected in the ACS routines. Figure 97 on page 240 shows the following:

- A storage class for high performance and with guaranteed space (SC54GRT) is assigned to the DFSMShsm control data sets and journal. The guaranteed space storage attribute allows you to reserve space on specific system-managed volumes for data sets that have specific performance or availability requirements. Guaranteed space is specified in the storage class for the CDSs — migration control data set (MCDS), offline control data set (OCDS), backup control data set (BCDS) — and the journal in order to direct them to different volumes behind different control units and subsystems to maximize performance and minimize data loss in the event of a hardware failure. This storage class is explicitly specified in the allocation of these data sets. Only three storage administrator userids, MHLRES3, MHLRES4, and MHLRES5, can use the SC54GRT storage class.
- Data allocated on DFSMShsm-owned volumes (migration and backup) is excluded from SMS management by assigning a null storage class to indicate that the data set should not be SMS-managed.

  DFSMShsm must be able to direct allocation of data sets it manages to DFSMShsm-owned storage devices so that, for example, backup versions of data sets go to backup volumes and migration copies go to migration volumes. DFSMShsm-owned volumes are not SMS-managed, so a storage class must not be assigned to DFSMShsm-owned data sets.
- Other DFSMShsm data sets, such as logs and CDS backups, are recognized and allocated in the SMS volume pools.
- When a data set is recalled or recovered by DFSMShsm, the storage class is retained if one already exists.

Examples of the parameters used in the SMS classes referenced in these routines are presented in 2.1.2, “Storage Class” on page 9.

Figure 98 on page 248 shows sample code required in the management class ACS routine, which assigns different management criteria to the DFSMShsm-owned data sets according to their use and importance.

Figure 99 on page 255 shows sample code that we used in our storage group ACS routine. You may need to create another storage group and update your storage group ACS routine if want to separate the DFSMShsm data sets from other data.

2.1.2 Storage Class

A storage class is a collection of attributes that identify performance goals and availability requirements for data sets. It is used to select a device that can meet those goals and requirements.

Every data set that SMS manages has a storage class assigned to it. To define the storage class for the DFSMShsm data sets, first go to the ISMF Primary Option Menu (see Figure 3 on page 10) and select option 5, Storage Class, to go to the Storage Class Application Selection panel (Figure 4 on page 10).

```
                    ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.4
Enter Selection or Command ===> 5

Select one of the following options and press Enter:

0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Criteria
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availability
6  Storage Group             - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
C  Data Collection           - Process Data Collection Function
L  List                      - Perform Functions Against Saved ISMF Lists
R  Removable Media Manager   - Perform Functions Against Removable Media
```

Figure 3. ISMF Primary Option Menu

On the Storage Class Application Selection panel select option 3, to define the storage classes you need. We define a storage class named SC54STD.

```
                      STORAGE CLASS APPLICATION SELECTION
Command ===>

To perform Storage Class Operations, Specify:
  CDS Name  . . . . . . . 'SYS1.SC54.SCDS1'
                            (1 to 44 character data set name or 'Active')
  Storage Class Name  . . SC54STD  (For Storage Class List, fully or
                                    partially specified or * for all)
Select one of the following options  :
  3  1. List          - Generate a list of Storage Classes
     2. Display       - Display a Storage Class
     3. Define        - Define a Storage Class
     4. Alter         - Alter a Storage Class
     5. Cache Display - Display Storage Classes/Cache Sets
If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

If Cache Display is Chosen, Specify Cache Structure Name . .
```

Figure 4. Sample Storage Class Application Selection Panel

We are now on the first page of the Storage Class Define panel (Figure 5 on page 11), where we can specify the basic parameters that we need to accomplish the performance and availability objectives of the DFSMShsm data sets.

```
                              STORAGE CLASS DEFINE              Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Class Name  : SC54STD
To DEFINE Storage Class, Specify:
  Description ==> Storage Class for Average Response Time
              ==>
  Performance Objectives
   Direct Millisecond Response . . . . 30           (1 to 999 or blank)
   Direct Bias . . . . . .             (R, W or blank)
   Sequential Millisecond Response . . 30           (1 to 999 or blank)
   Sequential Bias . . . . . . . . . . W            (R, W or blank)
   Initial Access Response Seconds . .              (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .              (0 to 999 or blank)
  Availability . . . . . . . . . . . . S            (C, P ,S or N)
  Accessibility  . . . . . . . . . . . S            (C, P ,S or N)
   Backup  . . . . . .                              (Y, N or Blank)
   Versioning  . . . . . .                          (Y, N or Blank)
```

```
                              STORAGE CLASS DEFINE              Page 2 of 2
 Command ===>

 SCDS Name . . . . . : SYS1.SC54.SCDS1
 Storage Class Name  : SC54STD

 To DEFINE Storage Class, Specify:

   Guaranteed Space . . . . . N           (Y or N)
   Guaranteed Synchronous Write . . . N   (Y or N)
   CF Cache Set Name  . . . . . .         (up to 8 chars or blank
   CF Direct Weight . . . . . .           (1 to 11 or blank)
   CF Sequential Weight . . . . . .       (1 to 11 or blank)
```

Figure 5. Sample Storage Class Define Panel

Table 1 on page 12 shows the content of the three storage classes that were developed for the DFSMShsm data sets:

SC54GRT is the class used to allocate, on a specific volume, data sets that require high performance, such as the DFSMShsm CDSs and the journal.

SC54STD is the class used to allocate data sets that require average performance. It can be used to allocate the DFSMShsm log and PDA data sets.

SC54LOW addresses allocation in a DFSMS work pool. It is used to allocate the DFSMShsm activity logs.

Table 1. Storage Classes for DFSMShsm Data Sets

Name	Direct Response	Direct Bias	Sequential Response	Sequential Bias	Initial Access Response	Availability	Accessibility	Backup	Versioning	Guaranteed Space	Guaranteed Synchronous Write
SC54GRT	5		5			C	C			YES	
SC54STD	30		30	W		N	N			No	
SC54LOW	999		999			N	N			No	

The most important fields of the Storage Class Define panel (Figure 5 on page 11) are explained below.

DIRECT and SEQUENTIAL MILLISECOND RESPONSE fields represent the performance objective during a new data set allocation. If no volume satisfies the objective, SMS makes one attempt to find a value that closely matches the specified objective.

BIAS specifies whether the majority of I/O is expected to be R (READ), W (WRITE), or unknown. The blank represents an unknown bias.

Cache usage and DASD fast write usage attributes are based on these parameters associated with the data set. If you do not want to use cache or DASD fast write, specify the Millisecond Response field as “999.”

AVAILABILITY is used to specify whether data set processing should continue after device failure. If you want to continue to process a data set, specify C for continuous. You can specify this option if you intend to allocate data sets on duplexed or RAID volumes. If data sets do not require such a high level of availability, specify N for no preference. SMS only selects volumes that are not dual copy. The default is N.

ACCESSIBILITY is used to specify that the data set should be allocated on devices under a controller that supports the concurrent copy feature, or the SnapShot feature of the RAMAC Virtual Array (RVA). This parameter has been enhanced by ISMF APAR OW27860, part of the maintenance that introduced the DFSMSdss SnapShot support, to include two new subparameters: BACKUP and VERSIONING. BACKUP means to take a fast point-in-time backup copy of a data set, through the use of a concurrent copy session. VERSIONING means to take a backup of a point-in-time version of a data set that was created by using the SnapShot feature of the RVA. The default is N for Accessibility, and blanks for BACKUP/VERSIONING.

GUARANTEED SPACE is used to permit preallocation of space for data sets. Primary space is allocated on each volume. After primary space on a given volume is used, as many secondary extents as possible are added to the volume before space on a different volume is used.

GUARANTEED SYNCHRONOUS WRITE is used to indicate whether your system should return from a BSAM CHECK (or WAIT) issued for a WRITE against a partitioned

data set extended (PDSE) member before (nonsynchronized) or after (synchronized) the data actually has been written to DASD. Specify N for no synchronization.

CF CACHE SET NAME is used to indicate the name of a CF cache set defined in the base configuration, thereby making any data set associated with this storage class eligible for VSAM record-level sharing (RLS), in a Parallel Sysplex environment. DFSMShsm CDSs can be accessed in RLS mode, allowing DFSMShsm to take advantage of the coupling facility hardware for CDS access.

CF DIRECT WEIGHT is used to indicate the direct data's relative importance in the CF structure.

CF SEQUENTIAL WEIGHT is used to indicate the sequential data's relative importance in the CF structure.

2.1.3 Management Class

A management class is a collection of management attributes that DFSMShsm uses to control action for retention, migration, backup, and release of allocated but unused space of data sets. To define a management class, use the Management Class Define panel (see Figure 6, Figure 7 on page 14, and Figure 8 on page 14), which is displayed by selecting option 3, Management Class, on the ISMF Primary Option Menu panel, then selecting option 3, Define, on the Management Class Application Selection menu.

```
                        MANAGEMENT CLASS DEFINE              Page 1 of 5
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Management Class Name : MC54NMIG

To DEFINE Management Class, Specify:

  Description ==> Management Class for no migration
              ==>

  Expiration Attributes
    Expire after Days Non-usage  . . NOLIMIT      (1 to 9999 or NOLIMIT)
    Expire after Date/Days . . . . . NOLIMIT      (0 to 9999, yyyy/mm/dd or
                                                  NOLIMIT)

  Retention Limit  . . . . . . . . . NOLIMIT      (0 to 9999 or NOLIMIT)
```

Figure 6. Sample Management Class Define Panel (Page 1)

```
                        MANAGEMENT CLASS DEFINE             Page 2 of 5
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Management Class Name : MC54NMIG

To ALTER Management Class, Specify:

  Partial Release . . . . . C.            (Y, C, YI, CI or N)

  Migration Attributes
    Primary Days Non-usage  . . . .            (0 to 9999 or blank)
    Level 1 Days Non-usage  . . . .            (0 to 9999, NOLIMIT or b
    Command or Auto Migrate . . . . NONE           (BOTH, COMMAND or NONE)


  GDG Management Attributes
    # GDG Elements on Primary . . .            (0 to 255 or blank)
    Rolled-off GDS Action . . . . .            (MIGRATE, EXPIRE or blan
```

Figure 7. Sample Management Class Define Panel (Page 2)

```
                        MANAGEMENT CLASS DEFINE             Page 3 of 5
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Management Class Name : MC54NMIG

To ALTER Management Class, Specify:
  Backup Attributes
    Backup Frequency  . . . . . .          (0 to 9999 or blank)
    Number of Backup Vers . . . . . .      (1 to 100 or blank)
      (Data Set Exists)
    Number of Backup Vers . . . . . .      (0 to 100 or blank)
      (Data Set Deleted)
    Retain days only Backup Ver . . .      (1 to 9999, NOLIMIT or bla
      (Data Set Deleted)
    Retain days extra Backup Vers . .      (1 to 9999, NOLIMIT or bla
    Admin or User command Backup  . . NONE      (BOTH, ADMIN or NONE)
    Auto Backup . . . . . N       (Y or N)
    Backup Copy Technique . . . . . S       (P=Conc Preferred, R=Conc
                                            Required or S=Standard)
```

Figure 8. Sample Management Class Define Panel (Page 3)

In this book, we use four management classes for the DFSMShsm data:

MC54SPEC is associated with system data sets and does not implement any type of management. It is used for the data sets that are defined in the DFSMShsm start procedure.

MC54NMIG is a primary pool management class. It is associated with the CDS backup data sets created by DFSMShsm. Because these data sets are already managed by DFSMShsm, this class does not exploit any automatic function.

MC54PRIM allows migration to ML1 and backup. It is used for data with medium-low frequency of access, such as some ABARS data sets.

MC54WORK manages data sets with a short life. It should be used for the activity log data sets when they are allocated on DASD.

Table 2 on page 15 shows the values used to code these four management classes.

Table 2. Values Used to Code DFSMShsm Management Classes

	SPACE MANAGEMENT									AVAILABILITY MANAGEMENT							
Name	Expire After Days Non-usage	Expire After Date/Days	Retention Limit	Partial Release	Primary Days Non-usage	Level 1 Days Non-usage	Command or Auto Migrate	# GDG Elements on Primary	Rolled-off GDG Action	Backup Frequency	Number of Backup Versions Data Set Exists	Number of Backup Versions Data Set Deleted	Retain Days Only Backup Version	Retain Days Extra Backup Version	Admin or User Command Backup	Auto Backup	Backup Copy Technique
MC54SPEC	NO	NO		N			NONE								NONE	N	R
MC54NMIG	NO	NO		C			NONE								NONE	N	
MC54PRIM	NO	NO		Y	3	9999	BOTH			1	2	1	30	5	BOTH	Y	
MC54WORK	7	NO		Y	3	9999	BOTH			1	1	0			BOTH	Y	

DFSMShsm deletes expired data sets during automatic space management processing. Page 1 of the Management Class Define panel contains the data set retention attributes. The retention attributes are processed before the migration attributes that you specify on the next panel. Page 2 of the Management Class Define panel contains the migration and GDG management attributes. Page 3 of the Management Class Define panel contains the management class backup attributes. The fields on the Management Class Define panel are explained below.

EXPIRE AFTER DAYS NON-USAGE specifies how much time must elapse since the last access before a data set or object becomes eligible for expiration.

EXPIRE AFTER DATE/DAYS specifies an absolute date or period after its allocation for a data set or object to become eligible for expiration. The default for both fields is NOLIMIT. The priority of EXPIRATION ATTRIBUTES is:

- If both EXPIRATION ATTRIBUTES are NOLIMIT, the data set or object never expires.
- If one of the EXPIRATION ATTRIBUTES is NOLIMIT, the non-NOLIMIT values must be used.
- If neither of the EXPIRATION ATTRIBUTES is NOLIMIT, both EXPIRATION ATTRIBUTES must be satisfied.

RETENTION LIMIT is a required value that limits the use of retention period (RETPD) and expiration date (EXPDT) values that are explicitly specified in the

job control language (JCL). If the value of a user-specified RETPD or EXPDT is within the limits specified by RETENTION LIMIT, it is saved for the data set. If a user-specified value exceeds a RETENTION LIMIT of nonzero, the RETENTION LIMIT overrides not only the RETPD/EXPDT values but also the EXPIRATION ATTRIBUTES values. The default RETENTION LIMIT field is NOLIMIT. If you specify 0, a user-specified value in the JCL or the data class derived EXPDT or RETPD is ignored (see Table 3).

Table 3. Retention Attribute Selection Example for Space Management Process

Attribute	Case 1	Case 2	Case 3	Case 4	Case 5
RETENTION LIMIT	0	**50**	100	NOLIMIT	100
EXPIRE AFTER DAYS NON-USAGE	**50**	50	50	50	50
EXPIRE AFTER DATE/DAYS	**100**	100	100	100	**100**
RETPD/EXPDT	60	60	**60**	**60**	

PARTIAL RELEASE applies to non-VSAM and VSAM extended format KSDSs. You can specify in this field whether, and under what conditions, allocated but unused space is to be released. The following four options are available for PARTIAL RELEASE:

- **Y** = Yes, release unused space at space management cycle time.
- **YI** = Yes, immediately. Release unused space when a Close is issued for a data set that was open for output and at space management cycle time.
- **C** = Conditional. If secondary space has been allocated, release unused space at space management cycle time.
- **CI** = Conditional, immediately. If secondary space has been allocated, release unused space when a Close is issued for a data set that was open for output and at space management cycle time.
- **N** = No release of unused space.

The default is N. The PARTIAL RELEASE attribute is not checked during the DFSMShsm interval migration function. If you want allocated but unused space to be released only when a secondary allocation quantity is present, specify C.

PRIMARY DAYS NON-USAGE represents the minimum number of days that must elapse since last access before a data set is eligible for migration. The default is 2.

LEVEL 1 DAYS NON-USAGE indicates the number of days that must elapse since the last reference on level 0 before a data set on ML1 can migrate from ML1 to ML2. The default is 60. If you want data sets to migrate directly from level 0 DASD to ML2 tape, specify 0 in the LEVEL 1 DAYS NON-USAGE field. The days spent on level 0 count toward eligibility for moving to ML2.

COMMAND OR AUTO MIGRATE allows you to specify whether a data set is eligible to be migrated by both command and automatic processing, by command alone, or not at all. If you do not want to migrate data sets that belong to a particular management class, specify NONE.

The **GDG MANAGEMENT ATTRIBUTES** indicate how many generations of a generation data group (GDG) to keep on primary storage and what to do with rolled-off generation data sets (GDSs). But, both the # GDG ELEMENTS ON PRIMARY and the ROLLED-OFF GDS ACTION fields are optional and have default values of blank. If left blank, no special treatment results from the data set being a GDS or a rolled-off GDS.

 # GDG ELEMENTS ON PRIMARY indicates how many of the most recent generations of a GDG to keep on primary storage. Generations that go over the limit become eligible for migration during the next run of automatic primary space management independent of the PRIMARY DAYS NON-USAGE. The most recent generations are eligible for migration only if they satisfy the PRIMARY DAYS NON-USAGE.

ROLLED-OFF GDS ACTION indicates whether to expire rolled-off GDSs or make them eligible for migration. If you specify MIGRATE, management class EXPIRATION ATTRIBUTES will be applied to the data set to determine whether or not the data set should be deleted; if not, the data is eligible for migration. If you specify EXPIRE, the rolled-off GDS will be deleted.

The ADMIN OR USER COMMAND BACKUP and AUTO BACKUP fields are required, and they have default values of BOTH and Y, respectively.

ADMIN OR USER COMMAND BACKUP is used to indicate the authority category for issuing command backups. NONE indicates that neither command nor autoprocessing is permitted. ADMIN indicates that only storage administrators can issue commands against these data sets. BOTH indicates that both storage administrators and users can issue commands against these data sets.

The remaining fields on this panel are required when you specify Y in the AUTO BACKUP field, or specify BOTH or ADMIN in the ADMIN OR USER COMMAND BACKUP field.

BACKUP FREQUENCY specifies how many days must elapse before DFSMShsm can back up data sets that have changed since the last backup. The default is 1. If you want to backup changed data sets every time a backup cycle is run, specify 0.

NUMBER OF BACKUP VERSIONS fields specify the maximum number of backup versions to retain for a data set. The default is 2 for **(DATA SET EXISTS)**, and 1 for **(DATA SET DELETED)**.

RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED) indicates how many days to keep the most recent backup version of a deleted data set, starting from the day DFSMShsm recognizes that the data set has been deleted. This attribute applies only when a data set no longer exists on a level 0 volume or a migrated volume (ML1 and ML2). The default is 60.

RETAIN DAYS EXTRA BACKUP VERSIONS indicates how many days to keep backup versions other than the most recent one. Days are calculated from the creation date of the backup version. The default is 30.

BACKUP COPY TECHNIQUE field specifies whether concurrent copy should be used during data set backup processing. R indicates that concurrent copy must be used for backup processing. P indicates that concurrent copy is preferred and it is used when the data set resides in a volume supported by concurrent copy. S is the default and indicates that concurrent copy is not used to back up the data set.

A description of object class transition attributes is beyond the scope of this book. For further information about this subject refer to the *DFSMSdfp Storage Administration Reference*.

2.1.4 Storage Group

Some storage groups can be updated to activate the DFSMShsm management functions on their data. This is done using the Pool Storage Group Alter panel (obtained through option 6 on the ISMF Primary Option Menu panel) and entering the appropriate information in the fields, as shown in Figure 9.

```
                          POOL STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Group Name  : SG54HSM
To ALTER Storage Group, Specify:
 Description ==> STORAGE GROUP FOR DFSMSHSM DATA SETS.
             ==>
 Auto Migrate . . Y  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . Y  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . Y  (Y or N)         Dump Sys/Sys Group Name  . . .

 Dump Class . . . ONSITE              (1 to 8 characters)
 Dump Class . . . OFFSITE             Dump Class . . .
 Dump Class . . .                     Dump Class . . .

 Allocation/migration Threshold: High . . 95  (1-99)      Low  . . 80  (0-99)
 Guaranteed Backup Frequency  . . . . . . NOLIMIT   (1 to 9999 or NOLIMIT)

 ALTER SMS Storage Group Status . . ..... N   (Y or N)
```

Figure 9. Pool Storage Group Alter Panel

The fields on the panel are explained below.

AUTO MIGRATE specifies whether you want the DASD volumes in this storage group to be eligible for automatic space management processing. Y specifies that DFSMShsm is to perform automatic primary space management processing and interval migration. I causes DFSMShsm to perform the same functions as if Y has been specified, and automatic interval migration independent of the DFSMShsm SETSYS INTERVALMIGRATION option. P specifies that data sets are eligible for primary space management but interval migration is not performed even if SETSYS INTERVALMIGRATION is specified. N specifies that data sets are not eligible for automatic migration.

AUTO BACKUP specifies whether you want the DASD volumes in this storage group to be eligible for automatic backup processing. Y specifies that DFSMShsm is to perform data availability management processing according to management class requirements.

The storage group and management class are interrelated. For example, if you set AUTO MIGRATE or AUTO BACKUP to N for the storage group, it will override the management class migration and backup attributes.

AUTO DUMP specifies whether you want to automatically dump all DASD volumes in this storage group. If you specify Y, DFSMShsm performs automatic dump processing.

MIGRATE, BACKUP, and **DUMP Sys/Sys Group NAME** fields should not be specified unless processing of the storage group for the function must be performed only

on that one host. Specifying these fields limits the ability of DFSMShsm to perform the request.

If you specify the SYSTEM NAME, it must match the value of the SYSNAME parameter in the IEASYSxx member of the respective PARMLIB.

DUMP CLASS can be specified with up to five unique dump class names. To use these dump classes, you must define their names and parameters, using the DEFINE command of DFSMShsm. These are optional fields.

ALLOCATION/MIGRATION THRESHOLD specifies an upper (HIGH) and lower (LOW) space limit for the DASD volumes in a pool storage group. The SMS volume selection function attempts to avoid allocation of a data set to a given DASD volume if that allocation will cause that volume's HIGH threshold to be exceeded. DFSMShsm uses the HIGH threshold value to determine whether the interval migration function should be performed, and the LOW threshold value as the threshold goal in reducing the amount of space occupied on a DASD volume in the storage group during interval migration and primary space management function. If you specify Y or I for AUTO MIGRATE, you must specify both a HIGH and a LOW threshold value.

GUARANTEED BACKUP FREQUENCY provides an additional criterion for creating a backup version and guarantees that a backup version of the data set will be created after a specified number of days has elapsed, whether or not the data set has been changed. If you specify NOLIMIT, data sets in the storage group will be backed up according to their management class specifications. If you specified Y or I for AUTO BACKUP, this field is required; otherwise it is optional.

ALTER SMS STORAGE GROUP STATUS indicates whether you want to change the status of the pool storage group. To leave the storage group status as ENABLE, accept the default, N.

2.1.5 SMS Update Checklist

The following checklist summarizes the activities necessary to update an existing SMS environment in preparation for installing and activating a DFSMShsm system:

- Identify the performance and availability requirements for all DFSMShsm data sets
- Identify the management requirements for the SMS-managed DFSMShsm data sets
- Define required storage classes
- Define required management classes
- Update storage class ACS routine
- Update management class ACS routine
- Update storage groups
- Translate, verify, and test new configuration
- Activate new configuration

2.2 PROCLIB and PARMLIB Setup

Before starting DFSMShsm, data sets key to DFSMShsm operation must be allocated, and procedures must be created in the appropriate procedure library. A base set of allocation procedures is found in SYS1.SAMPLIB. DFSMShsm ships modifiable JCL for you to build your DFSMShsm environment based on your installation requirements. To obtain the JCL, you must:

1. Edit member ARCSTRST of SYS1.SAMPLIB. This member contains instructions describing the fields to be changed.
2. Run member ARCSTRST in data set SYS1.SAMPLIB. This will allocate data set HSM.SAMPLE.CNTL containing several members that can be used to perform the installation tasks.
3. Edit member STARTER of HSM.SAMPLE.CNTL. This member contains basic instructions on how to set up the DFSMShsm environment.
4. Run member STARTER in data set HSM.SAMPLE.CNTL. The STARTER job will create DFSMSHSM in data set SYS1.PROCLIB, and a sample ARCCMD00 member in data set SYS1.PARMLIB and will allocate the DFSMShsm control data sets, journal, logs, and PDA trace data sets.

SYS1.PROCLIB member DFSMSHSM is a procedure to start DFSMShsm. It allocates the required data sets to DFSMShsm and allows you to specify several startup options.

Figure 10 on page 21 shows an example of a DFSMShsm startup procedure that was created when we ran the STARTER member in HSM.SAMPLE.CNTL. It is tailored for our primary DFSMShsm system.

```
//*****************************************************************/
//DFSMSHSM   PROC CMD=54,     USE PARMLIB MEMBER ARCCMD54
//            EMERG=NO,       ALLOW ALL DFSMSHSM FUNCTIONS
//            LOGSW=YES,      SWITCH LOGS AT STARTUP
//            STARTUP=YES,    STARTUP INFO PRINT AT STARTUP
//            UID=HSM,        DFSMSHSM-AUTHORIZED USER ID
//            SIZE=0M,        REGION SIZE FOR DFSMSHSM
//            DDD=50,         MAX DYNAMICALLY ALLOCATED DATASETS
//            HOST=1Y,        PROC.UNIT ID AND LEVEL FUNCTIONS
//            PDA=YES,        BEGIN PDA TRACING AT STARTUP
//            CDSR=YES        RESERVE CONTROL DATA SET VOLUMES
//*****************************************************************/
//  DFSMShsm STARTUP PROCEDURE FOR PRIMARY PROCESSOR                */
//*****************************************************************/
//DFSMSHSM EXEC PGM=ARCCTL,DYNAMNBR=&DDD,REGION=&SIZE,TIME=1440,
//          PARM=('EMERG=&EMERG','LOGSW=&LOGSW','CMD=&CMD',
//          'UID=&UID','STARTUP=&STARTUP','HOST=&HOST',
//          'PDA=&PDA','CDSR=&CDSR')
//*****************************************************************/
//*****************************************************************/
//*   HSMPARM DD must be deleted from the JCL or made into a      */
//*       a comment to use Concatenated Parmlib Support           */
//*****************************************************************/
//HSMPARM  DD DSN=SYS1.PARMLIB,DISP=SHR
//MSYSOUT  DD SYSOUT=A
//MSYSIN   DD DUMMY
//SYSPRINT DD SYSOUT=A,FREE=CLOSE
//SYSUDUMP DD SYSOUT=A
//*
//MIGCAT   DD DSN=&UID..MCDS,DISP=SHR
//JOURNAL  DD DSN=&UID..JRNL,DISP=SHR
//ARCLOGX  DD DSN=&UID..HSMLOGX1,DISP=OLD
//ARCLOGY  DD DSN=&UID..HSMLOGY1,DISP=OLD
//ARCPDOX  DD DSN=&UID..HSMPDOX1,DISP=OLD
//ARCPDOY  DD DSN=&UID..HSMPDOY1,DISP=OLD
//*
//BAKCAT   DD DSN=&UID..BCDS,DISP=SHR
//*
//OFFCAT   DD DSN=&UID..OCDS,DISP=SHR
//*
```

Figure 10. DFSMShsm Example of Startup Procedure for Primary Processor

Do not include keywords that are not necessary. The EXEC statement of this sample contains the following parameters:

SIZE is the DFSMShsm region size.

DDD is used to calculate the maximum number of data set allocations that DFSMShsm can hold in anticipation of reuse.

PARM is used to pass variable information to the processing program.

The length of the PARM subparameters on the EXEC statement must not exceed 100 characters (including commas, excluding parentheses). The PARM parameter of this sample contains the following subparameters:

CMD specifies the two-character suffix of the ARCCMDxx member of the library pointed to by the HSMPARM DD statement. It contains the parameters that define the DFSMShsm working options.

EMERG overrides the SETSYS EMERGENCY definition (see later) and allows, in case of errors, DFSMShsm to be started in a configuration that can be used for recovery. In normal conditions, this parameter is set to NO (default).

LOGSW specifies whether to automatically swap the log data sets at startup.

STARTUP specifies whether startup messages are to be printed at the operator console.

UID is the ID of the authorized DFSMShsm user.

HOST specifies a unique identification of the processing unit where DFSMShsm is running, followed by a character indicating whether this is a primary system (use Y) or a secondary system (use N). The backup and dump level functions are done on the primary processing unit. The automatic secondary space management functions can be performed on any host.

PDA specifies that the PDA tracing begins before the SETSYS PDA command has been processed or the DFSMShsm initialization has been completed.

CDSR specifies that DFSMShsm serializes its CDSs with volume reserves. If you have a cross-system serialization product, you can specify CDSQ=YES instead. CDSQ=YES serializes the CDSs (among multiple MVS images) with a global (SYSTEMS) exclusive enqueue but allows other tasks within a single MVS image to access the CDSs concurrently.

The following keywords are not specified in our startup procedure but are mentioned in this book and are described here for completeness:

CDSSHR specifies that DFSMShsm will run in a particular multiple processor or single processor environment instead of letting DFSMShsm determine the environment. If you specify NO for this keyword. DFSMShsm does no multiple processor serialization: no other processor should be concurrently processing the CDSs. Specifying YES for this keyword causes DFSMShsm to perform multiple processor serialization of the type requested by the CDSQ and CDSR keywords. If your system supports VSAM RLS, you can specify CDSSHR=RLS to cause DFSMShsm to perform multiprocessor serialization using RLS (serialization at a record level). The CDSs are accessed in RLS mode, and any values specified for CDSQ and CDSR are ignored. Other MVS images access the CDSs concurrently.

RESTART specifies that DFSMShsm should be restarted for all DFSMShsm abnormal ends (ABENDS). The format of the keyword is RESTART=(a,b), where *a* specifies the name of the procedure to be started, and *b* specifies any additional keywords or parameters to be passed to the procedure. If you are accessing the CDSs in RLS mode, use the RESTART keyword so that DFSMShsm automatically restarts after an SMS VSAM server error.

The DD statements in the procedure point to the following data sets:

HSMPARM is the library that contains the DFSMShsm SETSYS parameters. It is typically SYS1.PARMLIB but can be any other source format library. The member names must be ARCCMD followed by a two-character suffix, as indicated in the CMD parameter of the procedure EXEC statement.

MSYSOUT is a system data set used by DFSMShsm to interact with MVS services and to receive messages from the TSO Terminal Monitor Program (TMP) and with messages issued when dynamic memory allocation takes place.

MSYSIN is a system data set used by DFSMShsm for support of TSO processing. It must point to a DUMMY data set.

SYSPRINT is the standard DFSMShsm message data set.

SYSUDUMP is used to collect dumps generated when an error occurs in the DFSMShsm primary or secondary address spaces. It is used only if the SETSYS NOSYS1DUMP option has been requested.

MIGCAT points to the MCDS.

JOURNAL points to the journal data set.

ARCLOGX is one of two data sets to which DFSMShsm sends log information. It is used as an alternative to the data set pointed to by ARCLOGY DD.

ARCLOGY is one of two data sets to which DFSMShsm sends log information. It is used as an alternative to the data set pointed to by ARCLOGX DD.

ARCPDOX is one of two trace data sets to which DFSMShsm sends information useful for debugging. It is used only when the PDA trace is activated with a SETSYS parameter. It is not necessary if SETSYS PDA(NONE) is coded in the DFSMShsm set of parameters.

ARCPDOY is used in the same way as the trace data set pointed to by the ARCPDOX DD. It is used when the other data set is full.

BAKCAT points to the BCDS.

OFFCAT points to the OCDS.

Note: Do not add any STEPCAT or JOBCARD statements to any DFSMShsm-started procedure. The results are unpredictable.

2.2.1 Multihost Startup Procedure

The procedure shown in Figure 10 on page 21 changes when DFSMShsm is started on a secondary system in a multihost environment (see Figure 11 on page 24).

```
//*****************************************************************/
//DFSMSHSM   PROC CMD=66,     USE PARMLIB MEMBER ARCCMD66
//            EMERG=NO,       ALLOW ALL DFSMSHSM FUNCTIONS
//            LOGSW=YES,      SWITCH LOGS AT STARTUP
//            STARTUP=YES,    STARTUP INFO PRINT AT STARTUP
//            UID=HSM,        DFSMSHSM-AUTHORIZED USER ID
//            SIZE=0M,        REGION SIZE FOR DFSMSHSM
//            DDD=50,         MAX DYNAMICALLY ALLOCATED DATASETS
//            HOST=2N,        PROC.UNIT ID AND LEVEL FUNCTIONS
//            PDA=YES,        BEGIN PDA TRACING AT STARTUP
//            CDSR=YES        RESERVE CONTROL DATA SET VOLUMES
//*****************************************************************/
//* DFSMShsm STARTUP PROCEDURE FOR A SECONDARY SYSTEM               */
//*****************************************************************/
//DFSMSHSM EXEC PGM=ARCCTL,DYNAMNBR=&DDD,REGION=&SIZE,TIME=1440,
//          PARM=('EMERG=&EMERG','LOGSW=&LOGSW','CMD=&CMD',
//          'UID=&UID','STARTUP=&STARTUP','HOST=&HOST',
//          'PDA=&PDA','CDSR=&CDSR')
//*****************************************************************/
//*****************************************************************/
//*   HSMPARM DD must be deleted from the JCL or made into a      */
//*       a comment to use Concatenated Parmlib Support           */
//*****************************************************************/
//HSMPARM  DD DSN=SYS1.PARMLIB,DISP=SHR
//MSYSOUT  DD SYSOUT=A
//MSYSIN   DD DUMMY
//SYSPRINT DD SYSOUT=A,FREE=CLOSE
//SYSUDUMP DD SYSOUT=A
//*
//MIGCAT   DD DSN=&UID..MCDS,DISP=SHR
//JOURNAL  DD DSN=&UID..JRNL,DISP=SHR
//ARCLOGX  DD DSN=&UID..HSMLOGX2,DISP=OLD
//ARCLOGY  DD DSN=&UID..HSMLOGY2,DISP=OLD
//ARCPDOX  DD DSN=&UID..HSMPDOX2,DISP=OLD
//ARCPDOY  DD DSN=&UID..HSMPDOY2,DISP=OLD
//*
//BAKCAT   DD DSN=&UID..BCDS,DISP=SHR
//*
//OFFCAT   DD DSN=&UID..OCDS,DISP=SHR
//*
```

Figure 11. DFSMShsm Startup Procedure for Secondary System

The following parameters must change for the secondary system:

CMD specifies the two-character suffix of the ARCCMDxx member in the library pointed to by the HSMPARM DD statement. In a multihost environment, the ONLYIF command allows you to define parameters for multiple hosts within a single ARCCMDxx PARMLIB member. The single command immediately following the ONLYIF command is conditionally executed when the host name specified on the HSMHOST parameter matches the host being started.

You can specify a different two-character suffix from that specified for the primary system startup procedure, to have the secondary system use a different ARCCMDxx member.

HOST must specify a different identification character for each processing unit. The second character must be set to N for all secondary systems.

The following DD statements in the secondary system must point to the *same* data sets as the primary system:

MIGCAT

BAKCAT

OFFCAT

JOURNAL

The following DD statements in the secondary system must point to data sets that are *unique* for each system:

ARCLOGX

ARCLOGY

ARCPDOX

ARCPDOY

In a multihost environment, it is advisable to use a data set naming convention that allows the DFSMShsm system to be identified from the data set name. For example, a suffix can be used in the data set name that helps you associate the HOST number as coded in the procedure parameters.

2.2.2 ABARS

On ABACKUP and ARECOVER, DFSMShsm automatically starts a secondary address space that handles the request. This address space is started using the procedure name that is specified with the SETSYS ABARSPROCNAME(name) command in the ARCCMDxx PARMLIB member. We used DFHSMABR as the procedure name, as shown in Figure 12

```
//*
//*****************************************************************/
//*     ABARS SECONDARY ADDRESS SPACE STARTUP PROCEDURE           */
//*****************************************************************/
//*
//DFHSMABR PROC
//DFHSMABR EXEC PGM=ARCWCTL,REGION=0M
//SYSUDUMP DD SYSOUT=A
//MSYSIN DD DUMMY
//MSYSOUT DD SYSOUT=A
/*
```

Figure 12. ABARS Secondary Address Space Start Procedure

2.2.3 Setting Up the Base SETSYS Parameters

The way DFSMShsm works is established by a set of parameters, most of them identified by the SETSYS command. The general format is:

```
SETSYS parameter(options)
```

These parameters are contained in the PARMLIB member pointed to by the HSMPARM DD statement in the ARCCMDxx started procedure.

These parameters can be divided into four groups that define:

- Base options

- Space management characteristics
- Availability management characteristics
- ABARS support

Details and suggestions on coding this information are discussed in this topic. Refer to the *DFSMShsm Storage Administration Reference* for a detailed explanation of each keyword.

2.2.3.1 Base Parameters

The base parameters set the basic working options of DFSMShsm. These parameters are not directly related to a particular function, but they establish some defaults, such as tape management, that define the way DFSMShsm implements the available options.

1. **What is the TSO user ID of the DFSMShsm administrator or system programmer?**

   The AUTH command authorizes this user to issue DFSMShsm commands. The command can be entered in the input stream as many times as needed to define more than one authorized user. However, there should be very few users given CONTROL authority because they can grant authorization to others. The format of this command is:

```
AUTH uid DATABASEAUTHORITY(CONTROL)
```

2. **What is the job entry subsystem (JES)?**

   DFSMShsm defaults to JES2. If you want to use JES3, you must specify the JES3 parameter before you specify the first ADDVOL command.

```
SETSYS JES2

SETSYS JES3
```

3. **Do you want DFSMShsm to reblock eligible data sets during recall or recovery?**

   All system-reblockable data sets are reblocked during recall and recovery if they should be. If they are not system reblockable but of selected categories, this parameter permits DFSMShsm to reblock them. If you do not want these data sets to be reblocked, use statement format (a). The format (b) statement allows reblocking during recall or in recover to any device type supported by DFSMShsm, including target volumes of the same type as the source volume.

```
(a) SETSYS NOCONVERSION

(b) SETSYS CONVERSION(REBLOCKTOANY)
```

4. **How do I specify data set serialization for data sets being backed up or migrated?**

   To prevent a data set from being changed during backup or migration, access to the data set is controlled using serialization. DFSMShsm serialization is determined by specifying SETSYS parameters. Use statement (a) below when you are sharing volumes and a serialization facility such as global resource serialization (GRS) is not provided. In this case a reserve is placed on the volume. An example of sharing volumes without GRS is with a VM or VSE system. Use statement (b) when you are in a single-host environment or are sharing data between OS/390 systems and are using GRS to serialize at the data set level.

```
(a) SETSYS DFHSMDATASETSERIALIZATION

(b) SETSYS USERDATASETSERIALIZATION
```

5. **Do you want to activate a DFSMShsm exit?**

   You can use DFSMShsm installation exits to customize DFSMShsm processing according to your installation requirements. These exits are described in the *DFSMS/MVS Installation Exits* manual, where you can find additional information about them.

   If you are not going to use a DFSMShsm exit, specify statement (a) below.

   Statement (b) shows an example of how to activate an exit. In this example, we are activating the DFSMShsm tape volume exit ARCTVEXT. This exit is called when a DFSMShsm-owned tape no longer contains valid data and therefore becomes empty. The ARTVEXT exit is used to tell to the tape management system that DFSMShsm has released the ownership of a DFSMShsm tape. If you are running DFSMShsm and DFSMSrmm Version 1 Release 4, you do not have to specify a tape volume exit.

   Note: If you have installed an original equipment manufacturer (OEM) tape management system, specify statement (b) and ask the tape management system vendor for its version of ARCTVEXT.

```
(a) SETSYS EXITON(  )

(b) SETSYS EXITON(TV)
```

6. **Do you want to write updated CDS records in the journal data set?**

   Journaling is necessary because it is the only way to recover an affected CDS without losing all updates entered after last backup. The JOURNAL parameter specifies that DFSMShsm write the BCDS, MCDS, and OCDS records in the journal data set when it updates them. Statement (a) below guarantees that each CDS change is recorded immediately to the journal. With this choice, if the system is heavily loaded, it is advisable to activate the DASD fast write function where the journal is allocated. Statement (b) specifies that DFSMShsm is not to write the updated BCDS, MCDS, and OCDS records in the journal data set.

```
(a) SETSYS JOURNAL(RECOVERY)

(b) SETSYS NOJOURNAL
```

7. **Do you want SMF to contain DFSMShsm statistics?**

   If yes, specify statement (a) below with an SMF user code smfid to assign it to DFSMShsm-generated records. DFSMShsm uses two consecutive user codes (smfid and smfid+1). The fields are required by the REPORT command. If you do not want DFSMShsm to write SMF records, use statement (b).

```
(a) SETSYS SMF(smfid)

(b) SETSYS NOSMF
```

8. **Which SYSOUT class do you want to assign to the DFSMShsm output?**

   With this command, the class and number of copies is assigned to the DFSMShsm procedure output. DFSMShsm defaults are class A and one copy (A 1).

```
   SETSYS SYSOUT(class copies)
```

9. **Do you want the DFSMShsm dump to be written in a system dump data set?**

   With statement (a) below, which we recommend, dump is written in a system dump data set every time an error occurs in the primary or secondary address space. This format is required if you are using the Interactive Problem Control System (IPCS). With statement (b), DFSMShsm dumps are written where indicated with a SYSABEND, SYSUDUMP, or SYSMDUMP DD statement in the DFSMShsm start procedure.

```
(a) SETSYS SYS1DUMP

(b) SETSYS NOSYS1DUMP
```

10. **Do you want DFSMShsm to write all activity log messages?**

    ACTLOGMSGLVL determines which ARC0734I data set movement messages will be written to the activity log. Statement (a) below specifies that messages will be generated and logged for all activities. Statement (b) specifies that messages will be generated only for activities with a nonzero return code. Statement (c) specifies that only the original space management message is generated.

```
(a) SETSYS ACTLOGMSGLVL(FULL)

(b) SETSYS ACTLOGMSGLVL(EXCEPTIONLY)

(c) SETSYS ACTLOGMSGLVL(REDUCED)
```

11. **Do you want DFSMShsm to write the activity log messages on DASD or in a SYSOUT queue?**

    ACTLOGTYPE allows you to choose whether to make the activity logs a DASD data set (a) or a SYSOUT data set (b). For *class*, substitute an alphanumeric character for the class DFSMShsm is to use for output. DASD specifies that DFSMShsm dynamically allocate DASD data sets with the high-level qualifier HSMACT.

    The DASD activity logs can be managed by DFSMShsm according to the SMS management class specifications (see 2.1.3, “Management Class” on page 13).

```
(a) SETSYS ACTLOGTYPE(DASD)

(b) SETSYS ACTLOGTYPE(SYSOUT(class))
```

12. **Do you want DFSMShsm to print information on the system console and in the activity logs?**

    Do not code this parameter if you do not want DFSMShsm monitor messages on the system console. Otherwise, you can select the information you want by coding STARTUP for messages at DFSMShsm initialization. MONITOR also the SPACE and VOLUME subparameters. SPACE is used to print the volume space-use messages at system console, during space management. VOLUME specifies that data set processing messages (ARC0734I) are going to be printed on the system console.

```
   SETSYS MONITOR(STARTUP SPACE VOLUME)
```

13. **Do you want warning messages on space utilization inside the CDSs and the journal?**

    Code this parameter only if you want DFSMShsm to change the default threshold value for the CDSs and journal occupancy, that is 80%.

```
   SETSYS MONITOR(JOURNAL(80)                                    -
                  BCDS(80)                                       -
                  MCDS(80)                                       -
                  OCDS(80))
```

14. **Do you want to limit the amount of space that DFSMShsm allocates in the common service area (CSA)?**

Reasonable starting values which are also the default values, are shown in statement (a). If you do not want to limit the amount of space allocated in the CSA, choose statement (b).

```
(a) SETSYS CSALIMITS(MWE(4)                    -
                     MAXIMUM(100)              -
                     ACTIVE(90)                -
                     INACTIVE(30))

(b) SETSYS NOCSALIMITS
```

15. **How many tape drives can you assign to the RECYCLE process?**

    Although an MVS image can process only one RECYCLE request at a time, RECYCLE can initiate up to 15 tape processing tasks. Through the following statement, you can limit the number of tape drives that RECYCLE will request:

```
SETSYS MAXRECYCLETASKS(nn)
```

16. **Do you want to use the hardware compaction feature on 3480X tape devices?**

    The improved data recording capability (IDRC) feature on 3480s, if installed, can be used on DFSMShsm 3480X output tapes if you code statement (a). If you do not want to use this feature on 3480X devices, code statement (b). This parameter is ignored for 3490 and 3590 devices.

```
(a) SETSYS TAPEHARDWARECOMPACT

(b) SETSYS NOTAPEHARDWARECOMPACT
```

17. **Do you want to reuse partially full tapes?**

    This parameter is used for cartridge-type tape volumes. To enable each task, with the exception of data set migration, to initially request a scratch tape, use statement (a) below. Data set migration is excluded to avoid marking a tape volume full after a command migration of a single data set to ML2. This option is of value when using automatic cartridge loaders, but not if the DFSMShsm tape output is being directed to high-capacity tapes, like the Magstar media tape, as they will not be well utilized.

    If you want DFSMShsm to append to tapes that were not marked full when they were last written to, use statement (b). With statement (b), DFSMShsm marks tapes full only when the maximum block count specified in the SETSYS TAPEUTILIZATION parameter is reached.

    DFSMShsm automatically processes TAPECOPY requests for tape volumes marked as full, and partial tapes without alternates are not selected for tape duplexing. It is advantageous not to have any partially full tapes in the DFSMShsm inventory.

    If you specify the global PARTIALTAPE parameter, it applies to both migration and backup. If you want to specify this parameter differently for

migration and backup, do not code the sample shown here. Use the definitions that are described on page 84.

```
(a) SETSYS PARTIALTAPE(MARKFULL)

(b) SETSYS PARTIALTAPE(REUSE)
```

18. **Do you want to reduce the occurrences of data sets spanning tape volumes?**

    In this statement you specify the maximum number of megabytes of tape that DFSMShsm may leave unused on tape while trying to eliminate spanning data sets. The default value for this parameter was changed by APAR OW32249 and now is 500 MB. We suggest that you use the default value for all device types from 3480 through 3590-1.

```
SETSYS TAPESPANSIZE(nnn)
```

19. **Do you want DFSMShsm to wait when a tape allocation is requested?**

    If you select statement (a), DFSMShsm continues other processing when no device is immediately available and will reissue the allocation request every 10 seconds for one minute. After seven attempts, DFSMShsm asks the operator whether to retry or fail the allocation and hence the function. We recommend that you use the NOWAIT parameter.

    If you want DFSMShsm to wait until a tape request is satisfied, use statement (b). Note that in (b) all DFSMShsm functions requesting allocations, opens, or closes are stopped until this request is satisfied. JES3 installations are forced to use WAIT because JES3 schedules tape drives.

```
(a) SETSYS INPUTTAPEALLOCATION(NOWAIT)
    SETSYS OUTPUTTAPEALLOCATION(NOWAIT)
    SETSYS RECYCLETAPEALLOCATION(NOWAIT)

(b) SETSYS INPUTTAPEALLOCATION(WAIT)
    SETSYS OUTPUTTAPEALLOCATION(WAIT)
    SETSYS RECYCLETAPEALLOCATION(WAIT)
```

20. **Do you want DFSMShsm to use its own tape pool?**

    With statement (a) below, DFSMShsm will request scratch tapes every time a volume is needed for dump and each time a continuation volume is needed at end of volume (EOV) for migration and backup. The SELECTVOLUME specification is typically made in conjunction with the TAPEDELETION specification. So, according to the TAPEDELETION specification in statement (a), when DFSMShsm no longer needs a tape volume, it tells the tape management system that it has released the ownership of that tape volume. The tape management system then returns this tape volume to the global scratch pool.

    With statement (b), DFSMShsm will select the tape within a defined tape pool if any are available. It is expected that the installation will have supplied a

set of standard label tapes and identified them to DFSMShsm, using the ADDVOL command. When a tape volume no longer contains valid data, the TAPEDELETION specification in statement (b) tells DFSMShsm to keep it for reuse.

We recommended not using specific scratch pools. Global scratch pools allow you to easily take advantage of automatic cartridge loaders, reducing the tape mount wait time. Global scratch pools also enable use of a tape management product like DFSMSrmm.

```
(a) SETSYS SELECTVOLUME(SCRATCH)
    SETSYS TAPEDELETION(SCRATCHTAPE)

(b) SETSYS SELECTVOLUME(SPECIFIC)
    SETSYS TAPEDELETION(HSMTAPE)
```

21. **Do you want to use installation-defined esoteric unit names for tape allocations?**

    DFSMShsm always requests a mount for a specific tape for input processing, so cartridge loaders are of little value for input. To ensure that non-cartridge-loader devices are used for input, you can direct DFSMShsm to use esoteric unit names in a special way that directs a cartridge to be allocated on a different set of devices for input and for output. Esoteric tape unit names that have been defined to your OS/390 system must be defined to DFSMShsm before they can be recognized and used in DFSMShsm commands as valid unit types. All esoterics identified to DFSMShsm with the SETSYS USERUNITTABLE command must appear in a single command.

    To identify esoteric tape unit names to DFSMShsm, use statement (a) below. If an esoteric name represents a set of units having automatic cartridge loaders, and the esoteric name is used to allocate a device for output, the output esoteric is translated to the generic tape unit equivalent for mounting the tape for input. If you do not want that translation, you can specify an alternate unit name translation such as in statement (b). Substitute *eso* with the esoteric name you want to use for output allocations, and *esi* with the esoteric name you want to use for input allocations. In statement (b), *eso* would designate tape drives with automatic cartridge loaders, and *esi* would designate tape drives without automatic cartridge loaders.

    You can use statement (c) if you do not want the translation to occur for certain esoteric names. In this example, a tape written by DFSMShsm to a tape associated with the esoteric name *eso1* would be allocated for input in a tape unit also associated with *eso1*.

    If no esoteric tape unit names are identified to DFSMShsm, use statement (c). Any previously defined esoteric names are no longer known to DFSMShsm. Statement (c) is the default.

```
(a) SETSYS USERUNITTABLE(eso1,eso2,eso3,..)

(b) SETSYS USERUNITTABLE(eso1:esi1,eso2:esi2...)

(c) SETSYS USERUNITTABLE(eso1:eso1,eso2:eso2...)

(d) SETSYS NOUSERUNITTABLE
```

22. **How long do you want DFSMShsm to wait for a tape mount?**

    Indicate the number of minutes (maximum 120) that DFSMShsm can wait for a tape mount before asking the operator about volume availability. The DFSMShsm default is 15 minutes.

```
    SETSYS MOUNTWAITTIME(10)
```

23. **Do you want DFSMShsm to ask the operator about the availability of tapes for input before using them?**

    If you want DFSMShsm to ask the operator if the input tapes requested are available, use statement (a). If you do not want DFSMShsm to issue this operator request, use statement (b). If all of the input tapes requested are in an automated or manual tape library, these action messages are not issued, even if (...TAPES(YES)) has been specified.

```
(a) SETSYS TAPEINPUTPROMPT(MIGRATIONTAPES(YES))
    SETSYS TAPEINPUTPROMPT(BACKUPTAPES(YES))
    SETSYS TAPEINPUTPROMPT(DUMPTAPES(YES))

(b) SETSYS TAPEINPUTPROMPT(MIGRATIONTAPES(NO))
    SETSYS TAPEINPUTPROMPT(BACKUPTAPES(NO))
    SETSYS TAPEINPUTPROMPT(DUMPTAPES(NO))
```

24. **Do you want to create a DUPLEX duplicate of your migration and backup tapes?**

    DUPLEX allows DFSMShsm to create two tapes concurrently: the original tape, intended to be kept onsite, and the alternate tape, taken offsite or written to a remote tape library. Statement (a) below allows you to duplex both BACKUP and MIGRATION tapes. If you do not want to duplex your tapes, use statement (b), or do not specify the DUPLEX parameter on the SETSYS command.

```
(a) SETSYS DUPLEX(MIGRATION(Y)   -
                  BACKUP(Y))

(b) SETSYS DUPLEX(MIGRATION(N)   -
                  BACKUP(N))
```

25. **Do you want to limit the maximum tape utilization?**

    DFSMShsm provides the TAPECOPY command and the DUPLEX tape option to create alternate migration and backup copies of cartridge-type tapes. The usefulness of these copies depends on a one-to-one cartridge copy. Because not all cartridges of a given type have exactly the same capacity, DFSMShsm writes only 97% of the capacity of a cartridge by default. If you want to change the amount of data written to migration and backup tapes, use statement (a) below. You may have to issue the SETSYS TAPEUTILIZATION command for each tape unit type you have if you want to change a particular default. If you are sure that you will never want to copy these tapes, use statement (b). DFSMShsm writes the entire cartridge until the EOV is reached.

    The LIBRARYMIGRATION and LIBRARYBACKUP subparameters are the only vehicle by which you can limit the amount of media used in a library migration cartridge, because esoteric unit names are ignored in an SMS-managed tape library.

    If you are using the IBM 3591 tape subsystem (IBM 3590 Magstar emulating 3490E support), we recommend that you set the PERCENTFULL value to 1100, as in statement (c).

```
(a) SETSYS TAPEUTILIZATION(UNITTYPE(unit) PERCENTFULL(95))
    SETSYS TAPEUTILIZATION(LIBRARYMIGRATION (PERCENTFULL(95))
    SETSYS TAPEUTILIZATION(LIBRARYBACKUP (PERCENTFULL(95))

(b) SETSYS TAPEUTILIZATION(UNITTYPE(unit) PERCENTFULL(100))
    SETSYS TAPEUTILIZATION(LIBRARYMIGRATION (PERCENTFULL(100))
    SETSYS TAPEUTILIZATION(LIBRARYBACKUP (PERCENTFULL(100))

(c) SETSYS TAPEUTILIZATION(UNITTYPE(unit) PERCENTFULL(1100))
```

26. **Do you want all migrated and backup copies on DASD to be RACF indicated?**

    If the RACF environment is set up as described in 2.3, “Providing Security for DFSMShsm Resources” on page 37, with always-call support and generic profiles defined for migration and backup qualifiers, we recommend that you use statement (a) below. Migration copies and backup versions are not RACF indicated.

    Use statement (b) to have migration copies and backup versions RACF-indicated.

```
(a) SETSYS NORACFIND

(b) SETSYS RACFIND
```

27. **How do you want to protect DFSMShsm tapes?**

    DFSMShsm requires that you select one of three ways to protect tapes. If you want to protect them with RACF, use statement (a) below. In this case, RACF must be updated as explained in 2.3, “Providing Security for DFSMShsm Resources” on page 37. This is the recommended way if RACF

is installed. If RACF is not installed, you can use statement (b). This will protect tapes with the 99365 expiration date. If you choose statement (c), DFSMShsm will keep tapes as password indicated so that only security-authorized programs can access them. You can choose more than one tape security option. It is valid to have both RACF and EXPIRATION specified with the TAPESECURITY parameter.

```
(a) SETSYS TAPESECURITY(RACF|RACFINCLUDE)

(b) SETSYS TAPESECURITY(EXPIRATION|EXPIRATIONINCLUDE)

(c) SETSYS TAPESECURITY(PASSWORD)
```

28. **Do you want the system to overwrite deleted migration and backup copies of data sets according to the RACF erase-on-scratch flag?**

    Statement (a) below applies only if RACF is installed. It controls the overwriting after a migration and backup copy is deleted from a DFSMShsm-owned DASD volume, only if the erase-on-scratch option is associated with the RACF profile of the user's data set. Specify statement (b) if you do not need this overwriting to occur.

```
(a) SETSYS ERASEONSCRATCH

(b) SETSYS NOERASEONSCRATCH
```

29. **Do you want RACF discrete profiles to be copied during data set backup?**

    This parameter applies only if RACF is installed. If you use generic profiles with always-call support, select statement (a) below. However, if you do not have an always-call environment, use only discrete profiles and specify statement (b).

```
(a)  SETSYS NOPROFILEBACKUP

(b)  SETSYS PROFILEBACKUP
```

30. **Do you want to issue authorized DFSMShsm commands in a non-RACF TSO batch environment?**

    To do this, the user ID associated with the job must match the user ID entered with the AUTH command. If RACF is installed, it will pass this information to DFSMShsm. In this case, use statement (a) below. If RACF is not installed, use statement (b). An installation without RACF will have DFSMShsm retrieve the user ID for TSO batch requests from the protected step control block (PSCB). An installation must ensure that the user ID is placed in the PSCB.

```
(a) SETSYS NOACCEPTPSCBUSERID

(b) SETSYS ACCEPTPSCBUSERID
```

31. **Do you want DFSMShsm to compact data sets during migration or backup?**

    If you want DFSMShsm to compact data sets during backup and migration to DASD and during backup and migration to tape, specify statement (a) below. Optionally, you can choose which functions will use DFSMShsm compaction. For example, if you only want compaction to occur during migration to DASD, specify statement (c). We recommend that you specify statement (d). If you do not want compaction to occur, do not specify the SETSYS COMPACT command.

    If the majority of your migration and backup data goes to IDRC-capable tape devices or RVA DASD, we recommend that you not specify the SETSYS COMPACT command.

```
(a) SETSYS COMPACT(ALL)

(b) SETSYS COMPACT(DASDBACKUP)

(c) SETSYS COMPACT(DASDMIGRATE)

(d) SETSYS COMPACT(DASDBACKUP DASDMIGRATE)

(e) SETSYS COMPACT(TAPEBACKUP)

(c) SETSYS COMPACT(TAPEMIGRATE)
```

32. **Do you want to specify the percentage of minimum space saving for software compaction?**

    If you request software compaction, DFSMShsm compares any history it has of the number of bytes written to the total bytes of the original data set and computes the percentage of bytes saved by compaction. If the saved percentage is not greater than or equal to the percentage you specified, DFSMShsm will not compact the data set during subsequent migrations or backups. The DFSMShsm percentage saved default is 40.

```
SETSYS COMPACTPERCENT(pct)
```

33. **Do you want to optimize DFSMShsm compaction?**

    If you use the DFSMShsm data compaction feature, you can optimize the algorithm associating the source and object compaction tables with the data set types. This is done by passing to DFSMShsm the last-level qualifier of the data set names that belong to these two groups with the following statements:

```
SETSYS SOURCENAMES(src1,src2,.......)
SETSYS OBJECTNAMES(obj1,obj2,.......)
```

2.3 Providing Security for DFSMShsm Resources

RACF can be used to provide security for your DFSMShsm environment. You define DFSMShsm resources to RACF for use during authorization checking. RACF controls which users can issue DFSMShsm commands and access DFSMShsm data sets and DFSMShsm-owned data sets. DFSMShsm requests RACF authorization before it allows access to data sets by non-DFSMShsm-authorized users.

In this section we explain how to define the DFSMShsm started task to RACF, create RACF profiles to protect DFSMShsm resources, and how to authorize access to those resources.

2.3.1 Defining a DFSMShsm RACF User ID

For a started procedure to access any of the system′s resources, a userid must be associated with that task. The userid should have all the proper authorizations for accessing the system resources, therefore it should be assigned the RACF OPERATIONS attribute. We recommend defining a new userid for each started procedure rather than using the default started task userid (for example, STCUSER). In our system we associated userid *HSM54* with the DFSMShsm started procedure, and userid *ABR54* with the ABARS started procedure. Following are the RACF commands we used to define the DFSMShsm user IDs:

```
ADDUSER hsm54 OPERATIONS DFLTGRP(SYS1) NAME('ITSO DFSMShsm Userid')
ADDUSER abr54 OPERATIONS DFLTGRP(SYS1) NAME('ITSO ABARS Userid')
```

Note: If you do not specify the DFLTGRP parameter on the ADDUSER command, the current connect group of the user issuing the ADDUSER command is used as the default.

You can also create RACF user IDs by submitting a batch job using the JCL in Figure 13 on page 38.

```
//RACFADDU JOB ,'ADD STC RACF USERS',
//         NOTIFY=&SYSUID,
//         MSGCLASS=X,
//         CLASS=A
//*********************************************************************
//***  THIS JOB WILL CREATE THE DFSMSHSM AND ABARS STC USER IDS       ***
//*********************************************************************
//S0000010 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 ADDUSER HSM54 DFLTGRP(SYS1) NAME('DFSMSHSM STC USERID') OPERATIONS
 ADDUSER ABR54 DFLTGRP(SYS1) NAME('ABARS STC USERID') OPERATIONS
/*
```

Figure 13. Sample JCL to Create a RACF User ID

2.3.2 Identifying Started Procedures to RACF

Before RACF 2.1, the only way to associate a started procedure with a RACF user ID was by coding the RACF started procedures table, ICHRIN03. With RACF 2.1, assigning RACF identities to started procedures has been greatly simplified by the introduction of the RACF STARTED class. You can add or modify security definitions for new and existing started procedures by issuing the RDEFINE and RALTER commands.

Even though the use of the RACF STARTED class is the preferred way of identifying started procedures to RACF, it is still mandatory to have the ICHRIN03 module. RACF cannot be initialized if ICHRIN03 is not present in the system. A dummy ICHRIN03 is shipped and installed with RACF.

2.3.2.1 RACF STARTED Class

The STARTED class allows you to assign RACF identities to started procedures dynamically using the RDEFINE and RALTER commands. Resource names in the STARTED class have the format *membername.jobname*. You assign identities such as the RACF user ID and group ID, using fields in the STDATA segment. You can define the started procedure resource, using either a generic profile name or a discrete profile name. A RACF generic profile describes one or more data sets that have a similar name structure. A RACF discrete profile describes a specific data set on a specific volume. In our system we created generic profiles for started procedures. Issue the following RACF commands to assign RACF identities to the DFSMShsm and ABARS started procedures:

```
RDEFINE STARTED (HSM54.*) UACC(NONE) STDATA(USER(HSM54) GROUP(SYS1))
RDEFINE STARTED (ABR54.*) UACC(NONE) STDATA(USER(ABR54) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
```

Note: After you have added profiles to the RACF STARTED class, refresh the in-storage profiles, using the SETROPTS REFRESH command. The SETROPTS GENERIC command is needed only when you define generic profiles.

2.3.2.2 Listing the Started Class Profiles

To verify the STARTED class definitions, use the RLIST command:

```
RLIST STARTED (HSM54.*)
RLIST STARTED (ABR54.*)
```

Figure 14 shows the output of the first command.

```
 READY

rlist started (hsm54.*)
 CLASS      NAME
 -----      ----
 STARTED    HSM54.* (G)

 LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
 -----  --------   ----------------  -----------  -------
  00    MHLRES3         NONE               NONE    NO

 INSTALLATION DATA
 -----------------
 NONE

 APPLICATION DATA
 ----------------
 NONE

 AUDITING
 --------
 FAILURES(READ)

 NOTIFY
 ------
 NO USER TO BE NOTIFIED
```

Figure 14. Output of the RLIST Command

2.3.2.3 ICHRIN03 Started Procedures Table

We do not recommend using the RACF started procedures table (ICHRIN03). The preferred way of adding started procedure users to the RACF database is by using the STARTED class.

The sample JCL in Figure 15 on page 40 shows how to identify the DFSMShsm and ABARS started procedures to RACF by assembling ICHRIN03. You cannot enter a generic profile name for the communications address space here; rather, you must add a new entry for each address space you want to use.

```
//ICHRIN03 JOB 'STARTED PROC TAB',MSGLEVEL=(1,1),REGION=4096K
//*******************************************************************//
//* ICHRIN03:                                                       *//
//*                                                                 *//
//*   RACF INSTALLATION PROCEDURE STEP: CREATE/UPDATE          @03C*//
//*   STARTED PROCEDURES TABLE.                                @P2C*//
//*                                                                 *//
//* NOTE: Please read the section titled "Coding the Started       *//
//*       Procedures Module" in the System Programming              *//
//*       Library: RACF.  It is important that you are familiar     *//
//*       with this section before attempting to use or modify      *//
//*       this sample.                                              *//
//*                                                                 *//
//*******************************************************************//
//STEP1     EXEC HLASMCL PARM.L=(RENT,XREF,LIST,LET,NCAL)
//*                                                              @04C
//C.SYSIN DD *
ICHRIN03 CSECT
         TITLE 'ICHRIN03 - STARTED PROCEDURES TABLE'
         EJECT
         DC    XL2'80xx'          CHANGE 'xx' TO THE HEX VALUE FOR
                                  THE NUMBER OF ENTRIES IN THE TABLE
*
         DC    CL8'HSM54   '      PROCNAME - SPECIFY YOUR DFSMShsm
         DC    CL8'HSM54   '      STARTED TASK USER ID
         DC    CL8'SYS1    '      GROUP
         DC    XL1'00'            NOT PRIVELEDGED OR TRUSTED
         DC    XL7'00'            RESERVED
*
         DC    CL8'ABR54   '      PROCNAME - SPECIFY YOUR ABARS
         DC    CL8'ABR54   '      STARTED TASK USER ID
         DC    CL8'SYS1    '      GROUP
         DC    XL1'00'            NOT PRIVELEDGED OR TRUSTED
         DC    XL7'00'            RESERVED
*
                .                  .
                .                  .
                .                  .
                .                  .
         END
/*
//L.SYSLMOD DD DSN=SYS1.LPALIB,            ** MUST BE LIBRARY WITH**
//             DISP=SHR,UNIT=YYYY,         ** NEW RELEASE OF RACF **
//             VOL=SER=XXXXXX
//L.SYSIN   DD *
       ENTRY ICHRIN03
       NAME ICHRIN03(R)
/*
//*
******************************** Bottom of Data ********************
```

Figure 15. ICHRIN03 Sample Source

2.3.3 Protecting DFSMShsm Data Sets

You must protect DFSMShsm resources, such as the CDSs, journal, logs, and backed up data sets from unauthorized access. In member STARTER of data set HSM.SAMPLE.CNTL you specify a high-level qualifier (parameter UID) for the journal, control, and log data sets. You can define a generic RACF data set profile, to protect the DFSMShsm data sets. In our system we chose a high-level qualifier of HSM. The following RACF command defines a generic data set profile with a universal access of NONE:

```
ADDSD 'HSM.**' UACC(NONE)
```

After you have created the RACF generic profile for protecting the DFSMShsm data sets, you must permit users or groups access to the RACF profile based on their requirements. For example, your storage administrator would need RACF ALTER access to the CDSs in order to move them to another volume or to increase their space allocations.

The following RACF command can be used to give user ID HSMSTOR ALTER access to the DFSMShsm CDSs:

```
PERMIT 'HSM.**' ID(HSMSTOR) UACC(ALTER)
```

2.3.3.1 Listing the Generic Data Set Profile

To verify the DFSMShsm generic data set profile, use the LISTDSD command:

```
LISTDSD DA('hsm.**') GENERIC
```

Figure 16 shows the output of the command.

```
 READY
INFORMATION FOR DATASET HSM.** (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    SYS1             NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        SYS1           NON-VSAM

NO INSTALLATION DATA
```

Figure 16. LISTDSD Command Output

2.3.3.2 Protecting DFSMShsm Activity Logs

DFSMShsm writes its activity logs to DASD if you specify SETSYS ACTLOGTYPE(DASD) in the ARCCMDxx member of PARMLIB. DFSMShsm allocates the activity logs with a high-level qualifier of HSMACT. The following RACF command defines a generic data set profile to protect the activity logs with a universal access of NONE:

```
ADDSD 'HSMACT.**' UACC(NONE)
```

After you have created the RACF generic profile for protecting the DFSMShsm activity logs, you must permit users or groups access to the RACF profile based on their requirements. Take into consideration that someone may use a batch job to access data in the activity logs.

The following RACF command can be used to give userid HSMUSR1 READ access to the activity logs:

```
PERMIT 'HSMACT.**' ID(HSMUSR1) UACC(READ)
```

2.3.3.3 Protecting DFSMShsm Tapes

To protect DFSMShsm-managed tapes that have RACF-protected data sets on them you:

- Install and activate RACF.
- Define to RACF the tapes you want to protect by defining the TAPEVOL resource class in the RACF class descriptor table.
- Specify the SETSYS TAPESECURITY(RACF) command.

  You define the RACF environment to DFSMShsm when you specify the SETSYS TAPESECURITY(RACF) command. DFSMShsm protects each backup, migration, and dump tape with RACF.

The way that you define your RACF TAPEVOL resource class is determined by the number of tapes you want to protect.

Protecting up to 5000 Tapes: If you are protecting up to a maximum of 5000 tapes, you define two RACF resource names in the TAPEVOL resource class:

- HSMABR is the name for aggregate backup and recovery tapes
- HSMHSM is the name for all other DFSMShsm tapes

Issue the following RACF commands:

```
RDEFINE TAPEVOL HSMABR
RDEFINE TAPEVOL HSMHSM
```

You can add RACF protection to the DFSMShsm tape volumes before DFSMShsm uses them, with the exception of the HSMABR tapes. You must add the tape volume serial number to the HSMHSM tape volume set, using the following RACF COMMAND:

```
RALTER TAPEVOL HSMHSM ADDVOL(volser)
```

Protecting More Than 5000 Tapes: To RACF protect more that 5000 tapes you define multiple RACF resource names for DFSMShsm tape volume sets in the TAPEVOL resource class. The resource names are:

- HSMHSM (must be defined)

- HSMABR for aggregate backup and recovery tapes
- DFHSMx

  where x is a nonblank character (alpanumeric, national, and the hyphen) that corresponds to the last nonblank character of the tape volumer serial number. You need to define a DFHSMx resource name for each x value that exists based on your installation naming standards.

The following RACF commands would add resource names to the TAPEVOL class for HSMHSM (required), HSMABR (for aggregate backup and recovery tapes), and DFHSMA (for all tapes with a volume serial number ending with the letter A):

```
RDEFINE TAPEVOL HSMHSM
RDEFINE TAPEVOL HSMABR
RDEFINE TAPEVOL DFHSMA
```

To activate the RACF protection of tape volumes using the DFHSMx resource names that have been defined, you must issue the following RACF command on each system in the sysplex:

```
RALTER TAPEVOL HSMHSM ADDVOL(DFHSM)
```

You can add RACF protection to the DFSMShsm tape volumes before DFSMShsm uses them, with the exception of the HSMABR tapes. You must add the tape volume serial number to the appropriate DFHSMx tape volume set, based on the last nonblank character of the tape volume serial number. To protect a tape with a volume serial of POK33H you would use the following RACF command:

```
RALTER TAPEVOL DFHSMH ADDVOL(POK33H)
```

Tapes already protected in the tape volume set of HSMHSM continue to be protected.

2.3.4 Controlling the Use of DFSMShsm Commands

There are different types of DFSMShsm commands. You can issue user commands to process your own data sets or data sets to which you have the required RACF authorization, for example, HLIST and HRECALL.

Authorized commands are entered primarily by a system operator from the system console, or by a DFSMShsm authorized user (for example, a system programmer or storage administrator) from a TSO terminal with the HSEND command. There is no RACF authorization checking done for DFSMShsm commands using the system console.

2.3.5 AUTH Command

The AUTH command is used to identify users who can issue authorized DFSMShsm commands and users who can issue DFSMShsm authorized commands. The AUTH command is also used to add, delete, and change the authority of other DFSMShsm users. The DFSMShsm storage administrator should be identified as the user who can change the authority of other DFSMShsm users. A user defined as authorized through the AUTH command can issue DFSMShsm commands, bypassing RACF authorization checking.

The AUTH command is specified in the ARCCMDxx member of PARMLIB during DFSMShsm startup or entered by DFSMShsm users who have the database authority control attribute.

1. **How do I authorize a user to issue all DFSMShsm commands and change the authorization of other users?**

   The following command allows user ID ITSOHSM to add, delete, or change the DFSMShsm authorization of other users. It can be placed in the ARCCMDxx member of PARMLIB or issued by a user with the database control attribute.

   ```
   AUTH ITSOHSM DATABASEAUTHORITY(CONTROL)
   ```

2. **How do I authorize a user to issue all DFSMShsm commands?**

   The following command allows user ID ITSOHSM to issue authorized commands with the exception of the AUTH command. It can be placed in the ARCCMDxx member of PARMLIB or issued by a user with the database control attribute.

   ```
   AUTH ITSOHSM DATABASEAUTHORITY(USER)
   ```

3. **How do I cancel the authority of a user to issue authorized DFSMShsm commands?**

   The command to revoke the authority of a DFSMShsm authorized user can be issued by a user with the database control attribute or placed in the ARCCMDxx member of PARMLIB. Use the following command to revoke the DFSMShsm authority of user ITSOHSM:

   ```
   AUTH ITSOHSM REVOKE
   ```

2.3.6 ARCCATGP GROUP

Issuing the UNCATALOG, RECATALOG, DELETE, or NOSCRATCH command against a migrated data set causes the data set to be recalled before the operation is performed. It is possible to authorize certain users to issue these commands without recalling the migrated data sets by connecting the user to the RACF group ARCCATGP. When a user is logged on under RACF group ARCCATGP, DFSMShsm bypasses the automatic recall for UNCATALOG, RECATALOG, and DELETE/NOSCRATCH requests for migrated data sets. The tasks to enable DFSMShsm to bypass automatic recall during catalog operations are:

1. Define RACF group ARCCATGP, using the following RACF command:

   ```
   ADDGROUP (ARCCATGP)
   ```

2. Connect users who need to perform catalog operations without automatic recall to ARCCATGP, using the following RACF command:

   ```
   CONNECT (userid1,. . .,useridn) GROUP(ARCCATGP) AUTHORITY(USE)
   ```

3. Each user who needs to perform catalog operations without automatic recall must log on to TSO specifying the GROUP(ARCCATGP) parameter on the TSO logon screen (Figure 17 on page 45) or the GROUP=ARCCATGP parameter on the JOB statement of a batch job (Figure 18 on page 45).

```
------------------------------ TSO/E LOGON ----------------------------
IKJ56714A Enter current password for MARY

   Enter LOGON parameters below:                  RACF LOGON parameters

   Userid    ===> ITSOHSM

   Password  ===>                               New Password ===>

   Procedure ===> IKJACCNT                      Group Ident  ===> arccatgp

   Acct Nmbr ===> ACCNT#

   Size      ===> 4096

   Perform   ===>

   Command   ===>

   Enter an 'S' before each option desired below:
           -Nomail         -Nonotice        -Reconnect        -OIDcard
```

Figure 17. TSO/E LOGON Panel Specifying RACF Group ARCCATGP

```
//HSMCAT   JOB(accounting information),'ARCCATGP Example',
//         USER=ITSOHSM,GROUP=ARCCATGP,PASSWORD=password
//STEP1    EXEC PGM=....
```

Figure 18. JCL Specifying RACF Group ARCCATGP

2.3.7 Protecting Migration and Back Up Data Sets

When a data set is migrated or backed up by DFSMShsm it is given a name based on the prefix you specify in the ARCCMDxx PARMLIB member and DFSMShsm constants.

The migration copy of a data set has the name:
prefix.HMIG.Tssmmhh.user1.user2.Xydd , where prefix is the prefix you specify on the SETSYS MIGRATEPREFIX command.

The backup version of a data set has the name:
prefix.BACK.Tssmmhh.user1.user2.Xydd , where prefix is the prefix you specify on the SETSYS BACKUPPREFIX command.

Migrated and backed up data sets should not be accessed as regular MVS data sets. Use generic profiles based on the prefix you specify on the SETSYS MIGRATEPREFIX and SETSYS BACKUPPREFIX to RACF protect migrated and backed up data sets. The RACF profiles should be created with a universal access authority (UACC) of NONE. Only DFSMShsm and your storage administrator should have reason to access them. The DFSMShsm started task does not need to be on the access lists, because DFSMShsm sets itself up as a privileged user to RACF. Users who have the RACF OPERATIONS attribute will automatically have access to the profiles.

In our system, migrated data sets have the prefix HSM. Use the following RACF command to define a generic data set profile to protect all migrated data sets with a universal access of NONE:

```
ADDSD 'HSM.**' UACC(NONE)
```

After you have created the RACF generic profile for protecting all migrated data sets, you must permit users access to the RACF profile based on their requirements.

The following RACF command can be used to give userid HSMADMN ALTER access to migrated data sets:

```
PERMIT 'HSM.**' ID(HSMADMN) UACC(ALTER)
```

2.3.8 RACF Protection for ABARS

DFSMShsm-authorized users (specified in the SETSYS AUTH command in the ARCCMDxx member of PARMLIB) can issue ARECOVER and ABACKUP commands. ABARS also uses RACF Facility class profiles to permit certain operators and users to issue the ABACKUP and ARECOVER commands.

You define RACF Facility class profiles and authorize users based on the level of authorization the user requires. Comprehensive authorization allows a user to issue the ABACKUP and ARECOVER commands for all aggregates. RACF will not check the authority of the user to access each data set in a given aggregate.

Restricted authorization restricts a user to issuing ABACKUP and ARECOVER commands for only the single aggregate specified in the ABARS Facility class profile name.

The RACF Facility class profiles have names beginning with STGADMIN (storage administration). These FACILITY profiles are used to protect ABARS functions as well as many other SMS functions.

Define the profiles for comprehensive command authority with the following RACF commands:

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP
RDEFINE FACILITY STGADMIN.ARC.ARECOVER
```

The following command authorizes a user to issue the ABACKUP command for all aggregate groups:

```
PERMIT STGADMIN.ARC.ABACKUP CLASS(FACILITY) -
         ID(user_ID) ACCESS(READ)
```

More restricted aggregate backup authority can be defined with profiles (STGADMIN.ARC.ABACKUP.*agname*) for each aggregate. Issue the following command to define a facility class for the ITSOU001 aggregate:

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP.ITSOU001
```

Authority to issue an ABACKUP command for aggregate ITSOU001 is given to user HSMRES1 by the following command:

```
PERMIT STGADMIN.ARC.ABACKUP.ITSOU001 CLASS(FACILITY) -
       ID(HSMRES1) ACCESS(READ)
```

Users with this restricted authority must have a minimum of READ access to all RACF-protected data sets in the aggregate group. If they do not have this level of access to the data sets, the ABACKUP command fails.

As with the ABACKUP commands, ARECOVER commands can also be restricted with a profile for each aggregate, STGADMIN.ARC.ARECOVER.*agname*. The use of DSCONFLICT(REPLACE), REPLACE as a conflict resolution data set action or REPLACE specified by ARCCREXT can also be restricted through the use of RACF Facility class profile STGADMIN.ARC.ARECOVER.*agname*.REPLACE:

```
RDEFINE FACILITY STGADMIN.ARC.ARECOVER.ITSOU001.REPLACE
```

2.4 DFSMShsm Journal and Control Data Sets

Before starting DFSMShsm, several data sets central to DFSMShsm operation must be allocated, and procedures must be created in the appropriate procedure library. DFSMShsm requires a minimum of one, and a maximum of three CDS, and one journal data set.

In this section we explain the purpose of the CDSs and the journal data set, how to allocate them, and how DFSMShsm uses them. We also describe the allocation of an integrated catalog facility (ICF) user catalog and an alias for DFSMShsm.

We include some common data set allocation techniques and make suggestions for allocating the data sets from a recovery and performance aspect. Most of the recommendations that we make are based on tried and tested algorithms that have been built up over the years. We do not attempt to describe how the shipped sample parameters are set, but we do provide enough information to get a DFSMShsm system running.

2.4.1 Starter Set JCL Modifications

To simplify the setup process, DFSMShsm ships modifiable JCL that allows you to build your DFSMShsm environment based on your installation requirements. To obtain the JCL:

1. Edit member ARCSTRST of SYS1.SAMPLIB. This member contains instructions describing the fields to be changed.
2. Run ARCSTRST. This will allocate an HSM.SAMPLE.CNTL data set with several members that can be used to perform the installation tasks.

The JCL we use in our implementation of DFSMShsm for this book is a modification of the examples that are shipped in partitioned data set (PDS) HSM.SAMPLE.CNTL, member name STARTER.

These examples are based on the assumption that the SMS ACS routines were updated according to the rules described in 2.2, “PROCLIB and PARMLIB Setup” on page 20.

2.4.2 DFSMShsm Alias Definition

We recommend that a DFSMShsm alias be defined in an existing user catalog or a newly allocated catalog before installation. This alias should be used as a high-level qualifier for all of the DFSMShsm work and control data sets. The alias HSM is used in the examples in this book. Member STARTER in the HSM.SAMPLE.CNTL data set contains sample JCL to allocate the DFSMShsm catalog alias.

We have separated the steps of the job stream that allocate the ICF catalog, alias for DFSMShsm, journal, and CDSs and indicate our modifications. Where applicable we make comments that we believe are worthy of note.

The JCL in Figure 19 invokes IDCAMS to allocate both the catalog and its associated alias:

```
//IDCAMS EXEC PGM=IDCAMS
//*******************************************************************/
//*
//HSMMCDS  DD UNIT=3390,VOL=SER=HSM14A,DISP=SHR
//HSMCAT   DD UNIT=3390,DISP=SHR,VOL=SER=HSM14C
//*
//*******************************************************************/
//* REMOVE THE NEXT DD STATEMENT IF YOU DO NOT INTEND TO USE BACKUP */
//* AND DUMP.                                                       */
//*******************************************************************/
//*
//HSMBCDS  DD UNIT=3390,VOL=SER=HSM14A,DISP=SHR
//*
//*******************************************************************/
//*
//SYSIN    DD *
/*******************************************************************/
/* THIS JOB ALLOCATES AN INTEGRATED CATALOG FACILITY (ICF) CATALOG */
/* AND ITS ASSOCIATED ALIAS "HSM".                                 */
/*                                                                 */
/*  ****** INTEGRATED CATALOG FACILITY CATALOG REQUIRED *******    */
/*                                                                 */
/* THIS JOB ALLOCATES A USER CATALOG FOR THE DFSMSHSM CONTROL DATA */
/* SETS (CDS). SEE THE SECTION "DFSMSHSM DATA SETS" IN THE         */
/* IMPLEMENTING AND CUSTOMIZING DFSMSHSM BOOK.                     */
/*******************************************************************/
/*                                                                 */
 DEFINE UCAT(NAME(UCAT.HSM) -
        CYLINDERS(1 1) VOLUME(HSM14C) -
        STORCLAS(SC54GRT) MGMTCLAS(MC54NMIG) -
        FILE(HSMCAT) FREESPACE(10 10) -
        RECORDSIZE(4086 4086) -
        ICFCATALOG) -
        CATALOG(MCAT.OS3R5V01.VTOTCAT)
 IF MAXCC = 0 THEN DO
   DEFINE  ALIAS(NAME(HSM) RELATE(UCAT.HSM))
 END
/*                                                                 */
```

Figure 19. Sample IDCAMS JCL to Allocate the DFSMShsm ICF Catalog and Define the Alias

2.4.3 Journal and CDS Overview

The journal data set is a sequential data set used by DFSMShsm to log all updates to the DFSMShsm CDSs. It facilitates recovery of a CDS in case of errors.

The DFSMShsm CDSs are resources that combine with your installation settings to allow DFSMShsm to manage your storage environment. The CDSs are virtual

storage access method (VSAM) data sets that contain the control information describing the status of the DFSMShsm managed data sets and volumes. Following is a description of the three DFSMShsm control data sets:

MCDS The migration control data set contains information about the migrated user or application data sets.

BCDS The backup control data set contains information about the backup copies of user or application data sets.

NOTE: This data set is not required if you do not intend to use backup and dump.

OCDS The offline control data set contains information about the data sets on migration and backup tape volumes used by DFSMShsm.

NOTE: This data set is not required if you do not intend to use tapes for daily or spill backup or ML2 processing.

2.4.3.1 CDS Allocation

The DFSMShsm CDS allocations are shown in the examples that follow. You have to tailor some of the JCL and SMS information to match that of your installation.

Figure 20 on page 50 shows the modifications that we have made to the starter set JCL for allocating the MCDS. We have taken most of the defaults that DFSMShsm supplies as they give adequate performance:

Additionally we have used storage classes and management classes in our JCL that match our environment. Refer to Appendix A, “Sample ACS Routines” on page 239 for the ACS routines used in our system.

```
/*****************************************************************/
/*  ADD THE IMBED AND REPLICATE PARAMETERS IF YOU PUT            */
/*  THE INDEX FOR THE CONTROL DATA SET ON A DEVICE SERVICED      */
/*  BY A NON-CACHED CONTROLLER.                                  */
/*                                                               */
/*  THIS PROCEDURE ASSUMES A SINGLE CLUSTER MCDS.  IF MORE THAN */
/*  ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER   */
/*  CDS.                                                         */
/*****************************************************************/
/*                                                               */
IF MAXCC = 0 THEN DO
  DEFINE CLUSTER (NAME(HSM.MCDS) VOLUMES(HSM14A) -
       CYLINDERS(10 5) FILE(HSMMCDS) -
       STORCLAS(SC54GRT) -
       MGMTCLAS(MC54SPEC) -
       RECORDSIZE(435 2040) FREESPACE(0 0) -
       INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
       SPEED BUFFERSPACE(530432) -
       UNIQUE NOWRITECHECK) -
       DATA(NAME(HSM.MCDS.DATA) -
       CONTROLINTERVALSIZE(12288)) -
       INDEX(NAME(HSM.MCDS.INDEX) -
       CONTROLINTERVALSIZE(2048)) -
       CATALOG(UCAT.HSM)
END
```

Figure 20. Sample MCDS Allocation Statements

Figure 21 on page 51 shows how we modified the starter set for the BCDS:

```
/*************************************************************/
/* REMOVE THE NEXT DEFINE COMMAND IF YOU DO NOT              */
/* INTEND TO USE BACKUP AND DUMP.                            */
/*                                                           */
/* ADD THE IMBED AND REPLICATE PARAMETERS IF YOU PUT         */
/* THE INDEX FOR THE CONTROL DATA SET ON A DEVICE SERVICED   */
/* BY A NON-CACHED CONTROLLER.                               */
/*                                                           */
/* THIS PROCEDURE ASSUMES A SINGLE CLUSTER BCDS.  IF MORE THAN */
/* ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER */
/* CDS.                                                      */
/*************************************************************/
/*                                                           */
IF MAXCC = 0 THEN DO
  DEFINE CLUSTER (NAME(HSM.BCDS) VOLUMES(HSM14A) -
       CYLINDERS(10 5) FILE(HSMBCDS) -
       STORCLAS(SC54GRT) -
       MGMTCLAS(MC54SPEC) -
       RECORDSIZE(6544 6544) FREESPACE(0 0) -
       INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
       SPEED BUFFERSPACE(530432) -
       UNIQUE NOWRITECHECK) -
       DATA(NAME(HSM.BCDS.DATA) -
       CONTROLINTERVALSIZE(12288)) -
       INDEX(NAME(HSM.BCDS.INDEX) -
       CONTROLINTERVALSIZE(2048)) -
       CATALOG(UCAT.HSM)
END
/*
```

Figure 21. Sample BCDS Allocation Statements

For our OCDS we modified the starter set as shown in Figure 22 on page 52:

```
/*************************************************************/
/* REMOVE THE NEXT DEFINE COMMAND IF YOU DO NOT              */
/* INTEND TO USE TAPES FOR DAILY BACKUP, SPILL BACKUP, OR    */
/* MIGRATION LEVEL 2 PROCESSING.                             */
/*                                                           */
/* ADD THE IMBED AND REPLICATE PARAMETERS IF YOU PUT         */
/* THE INDEX FOR THE CONTROL DATA SET ON A DEVICE SERVICED   */
/* BY A NON-CACHED CONTROLLER.                               */
/*                                                           */
/* THE OCDS MAY NOT EXCEED 1 VOLUME.                         */
/*************************************************************/
/*
IF MAXCC = 0 THEN DO
  DEFINE CLUSTER (NAME(HSM.OCDS) VOLUMES(HSM14A) -
       CYLINDERS(10 5) FILE(HSMOCDS) -
       STORCLAS(SC54GRT) -
       MGMTCLAS(MC54SPEC) -
       RECORDSIZE(1800 2040) FREESPACE(0 0) -
       INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
       SPEED BUFFERSPACE(530432) -
       UNIQUE NOWRITECHECK) -
       DATA(NAME(HSM.OCDS.DATA) -
       CONTROLINTERVALSIZE(12288)) -
       INDEX(NAME(HSM.OCDS.INDEX) -
       CONTROLINTERVALSIZE(2048)) -
       CATALOG(UCAT.HSM)
END
```

Figure 22. Sample OCDS Allocation Statements

2.4.3.2 Journal Data Set Allocation

Figure 23 is our modified example of the JCL shipped for journal allocation:

```
//*************************************************************/
//*  THE JOURNAL MUST NOT EXCEED 1 VOLUME, MAY NOT HAVE       */
//*  SECONDARY ALLOCATION, AND MUST BE ALLOCATED CONTIGUOUS.  */
//*  ALLOCATE AT LEAST 100 CYLINDERS.                         */
//*************************************************************/
//LOGALC  EXEC PGM=IEFBR14
//*
//JOURNAL DD DSN=HSM.JRNL,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14B,SPACE=(CYL,(100),,CONTIG),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//*
```

Figure 23. Sample Journal Allocation Statements

2.4.4 CDS and Journal Performance Suggestions

When allocating the journal and CDSs, it is important to consider your required availability and performance levels. Particularly during automatic functions, there is intense activity on these data sets, and it is important to keep the I/O response time at a minimum level.

The following is a list of considerations about CDSs and the journal data set:

- Allocate CDSs on a cached DASD and the journal on DASD fast write.
- Attempt to keep the journal separate from the MCDS, BCDS, and OCDS.
- Allocate CDSs on RAMAC or RVA devices if possible.
- Allocate CDSs and the journal on different volumes from those containing JES3 data sets or system data sets.
- Allocate CDSs on low-use mounted or private volumes.
- Define only a primary space allocation for the CDSs and specify CONTIG for the journal.
- If CDSs and the journal are SMS-managed, they must be assigned to a storage class with the GUARANTEED SPACE attribute.
- Ensure that the CDSs, CDS backup copies, journal, and journal backup copies are not allowed to migrate.
- Use different head disk assemblies (HDAs) for CDSs and the journal; consider using volumes connected to different control units.
- Use the SETSYS CDSVERSIONBACKUP to back up the CDSs and journal; if possible use concurrent copy.
- Implement and test CDS backup and recovery procedures before you begin to manage user or application data.
- Once you have tested the CDS backup and recovery procedures, document them.

2.4.5 CDS Sharing Considerations

If you intend to run DFSMShsm in a multiple processor configuration or within a sysplex, there are considerations for the VSAM share options of the CDSs. This is important to prevent any form of integrity exposure and any interlock of the DFSMShsm CDSs.

If you are in a multiple processor configuration, the CDSs must reside on shared DASD. Specify the type of serialization you want by using the CDSQ, CDSR, or CDSSHR keywords.

To provide protection against an interlock condition without a GRS product installed you must protect your CDSs by:

- Allocating a CDS and the catalog it is cataloged in on the same volume.
- Ensuring CDSs are not placed on ML1 volumes.
- Ensuring CDSs are not placed on the same volume with system resource data sets.

There are two methods for defining the share options for the DFSMShsm CDSs.

Method 1 - VSAM SHAREOPTIONS(2 3)

The following share option strategy will provide maximum protection against accidental non-DFSMShsm concurrent updates:

- Use GRS and define the DFSMShsm CDSs with VSAM SHAREOPTIONS(2 3).
- Exclude the SYSVSAM resources related to the CDS components from being passed around the GRS ring.

Cross-region share option 2 allows only one processor at a time to open a data set for output. In this case we expect the open to be performed by DFSMShsm. No jobs will be able to update the CDSs apart from DFSMShsm.

Add the following statements to the GRSRNLxx PARMLIB member:

```
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSVSAM) RNAME(MCDS-index-name)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSVSAM) RNAME(MCDS-data-name)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSVSAM) RNAME(BCDS-index-name)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSVSAM) RNAME(BCDS-data-name)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSVSAM) RNAME(OCDS-index-name)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSVSAM) RNAME(OCDS-data-name)
```

Some additional considerations apply when you use this sharing strategy:

- Reserve contentions can occur when a GRS product is not installed and that system processes DFSMShsm and applications concurrently with VSAM data sets on the same volume.
- Specify DISP=SHR for read-only utilities.
- Specify DISP=OLD for any utility that will alter the CDSs.
- Do not reorganize your CDSs while DFSMShsm is running on any processor that uses those CDSs.

Method 2 - VSAM SHAREOPTIONS(3 3)

The DFSMShsm starter set will allocate the CDSs with VSAM SHAREOPTIONS(3 3) to allow DFSMShsm to be easily started in a multiple processor environment with GRS or equivalent function. DFSMShsm provides an appropriate serialization protocol to ensure read and update integrity of the CDSs. If you choose to implement VSAM RLS, which we discuss in 2.4.7, “VSAM Record Level Sharing” on page 63, this is the share option that you must specify.

It is important to note that if either of these share options is chosen, a data integrity exposure may exist if DFSMShsm is not active on all connected processors. Strictly control any CDS reorganization or maintenance procedure such that the utility job is allocated with a disposition of OLD (DISP=OLD).

2.4.6 CDS and Journal Backup

The DFSMShsm primary host automatically backs up its CDSs and the journal, using the SETSYS CDSVERSIONBACKUP parameter. This is done as the first phase of automatic backup. According to your SETSYS environment specifications, this backup can be scheduled to DASD or tape. However, unlike the backup of user data sets, CDSs must be backed up to only one kind of I/O device. You cannot use a mixture of tape and DASD. If backup to DASD is implemented, a set of sequential data sets must be preallocated to which the backup copies will be written.

We recommend using multiple backup versions that are written to preallocated data sets. DFSMShsm will change the suffix of the data set name dynamically and it is similar to that of GDS. The JCL necessary to allocate these data sets is in member ALLOCBK1 of PDS HSM.SAMPLE.CNTL.

When you allocate a set of data sets, the SETSYS BACKUPCOPIES(nn) parameter indicates the number of preallocated data sets for each CDS and the journal that you want to keep. The default number is 4.

Our recommendation is that you use DFSMSdss to back up the CDSs from within DFSMShsm. For the parameters that you need to define see 2.4.6.3, “Steps for Defining the CDS and Journal Backup Environment” on page 59. The invocation of DFSMSdss checks the structure of the CDSs as it dumps them and advises you of any errors that may make recovery of the CDSs impossible.

As the area of CDS and journal backup can be quite complicated, we provide a short section, 2.4.6.1, “Shortcut to Defining CDS and Journal Backups” on page 55, where we show how we set up our environment. We have included some assumptions, and, if you are certain that they apply to you, you can use them.

However, if you would like to know more about the parameters and the various options available, read the more comprehensive section, 2.4.6.2, “Creating CDS and Journal Backups” on page 59, where we explain what each parameter setting achieves.

2.4.6.1 Shortcut to Defining CDS and Journal Backups

This shortcut to defining your CDS and journal backup environment assumes that all of the following are true for the CDSs and the journal:

- You will use SETSYS CDSVERSIONBACKUP.
- Your SMS policies do not allow the CDSs and journal to be migrated or backed up.
- DFSMSdss will be used as the data mover.
- The number of backup copies you want to keep is four.
- You will be backing up to DASD.
- Your DFSMShsm authorized user ID is the same as the high level qualifier for the DFSMShsm generated data set names.

If all of the above statements are true for you, modify the JCL shown in Figure 24 on page 56 to suit your environment. We have altered the JCL to reflect our environment and changed the comments within the JCL. The jobs represent the settings that we used to back up our CDSs and journal for this book.

```
//ALLOCBK1 JOB ,'JON TATE',MSGCLASS=X,CLASS=A
//ALLOCBK EXEC PGM=IEFBR14
//*
//*****************************************************************/
//*    THIS SAMPLE JOB ALLOCATES AND CATALOGS THE CONTROL DATA SET*/
//*    BACKUP VERSION DATA SETS ON DASD VOLUMES.                  */
//*                                                               */
//*    ENSURE THAT BACKUP VERSION DATA SETS ARE PLACED ON VOLUMES */
//*    THAT ARE DIFFERENT FROM THE VOLUMES THAT THE CONTROL DATA  */
//*    SETS ARE ON.                                               */
//*                                                               */
//*    THIS SAMPLE JOB ALLOCATES FOUR BACKUP COPIES (THE DEFAULT) */
//*    FOR EACH CONTROL DATA SET.  IF YOU SPECIFY A DIFFERENT     */
//*    NUMBER OF BACKUP VERSIONS, ENSURE THAT YOU ALLOCATE A      */
//*    BACKUP COPY FOR EACH OF THE BACKUP VERSIONS YOU SPECIFY.   */
//*****************************************************************/
//*    MODIFIED STARTER SET JCL (MEMBER ALLOCBK1)                 */
//*****************************************************************/
//*  PARAMETER DEFINITION
//*
//*  3390       -  UNIT TYPE OF VOLUME TO CONTAIN THE FIRST CDS
//*                BACKUP VERSION.
//*  3390       -  UNIT TYPE OF VOLUME TO CONTAIN THE SECOND CDS
//*                BACKUP VERSION.
//*  3390       -  UNIT TYPE OF VOLUME TO CONTAIN THE THIRD CDS
//*                BACKUP VERSION.
//*  3390       -  UNIT TYPE OF VOLUME TO CONTAIN THE FOURTH CDS
//*                BACKUP VERSION.
//*  HSM14C     -  VOLUME SERIAL OF VOLUME TO CONTAIN THE FIRST CDS
//*                BACKUP VERSION.
//*  HSM14C     -  VOLUME SERIAL OF VOLUME TO CONTAIN THE SECOND CDS
//*                BACKUP VERSION.
//*  HSM14C     -  VOLUME SERIAL OF VOLUME TO CONTAIN THE THIRD CDS
//*                BACKUP VERSION.
//*  HSM14C     -  VOLUME SERIAL OF VOLUME TO CONTAIN THE FOURTH CDS
//*                BACKUP VERSION.
//*  SC54GRT    -  STORAGE CLASS WITH GUARANTEED SPACE
//*  MC54SPEC   -  MANAGEMENT CLASS WITH NO AUTOMIG, NO BACKUP AND
//*                BACKUP COPY TECHNIQUE OF 'REQUIRED'
//*  15         -  NUMBER OF CYLINDERS ALLOCATED TO CDS BACKUP
//*                VERSIONS. (CDS SIZE + 50%)
//*  100        -  NUMBER OF CYLINDERS ALLOCATED TO JOURNAL DATA
//*                SET BACKUPS (SAME AS ORIGINAL)
//*  HSM        -  AUTHORIZED USER ID FOR OUR HSM-STARTED
//*                PROCEDURE. THIS WILL BE USED AS THE
//*                HIGH-LEVEL QUALIFIER OF HSM DATA SETS.
```

Figure 24 (Part 1 of 3). Sample JCL to Allocate CDS anf Journal Backup Data Sets

```
//*****************************************************************/
//* THIS PROCEDURE ASSUMES A SINGLE CLUSTER MCDS.  IF MORE THAN    */
//* ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER     */
//* CDS.                                                           */
//*****************************************************************/
//*
//MCDSV1 DD DSN=HSM.MCDS.BACKUP.V0000001,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//MCDSV2 DD DSN=HSM.MCDS.BACKUP.V0000002,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//MCDSV3 DD DSN=HSM.MCDS.BACKUP.V0000003,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//MCDSV4 DD DSN=HSM.MCDS.BACKUP.V0000004,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//*
//*****************************************************************/
//* THIS PROCEDURE ASSUMES A SINGLE CLUSTER BCDS.  IF MORE THAN    */
//* ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER     */
//* CDS.                                                           */
//*****************************************************************/
//*
//BCDSV1 DD DSN=HSM.BCDS.BACKUP.V0000001,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//BCDSV2 DD DSN=HSM.BCDS.BACKUP.V0000002,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//BCDSV3 DD DSN=HSM.BCDS.BACKUP.V0000003,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//BCDSV4 DD DSN=HSM.BCDS.BACKUP.V0000004,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//*
```

Figure 24 (Part 2 of 3). Sample JCL to Allocate CDS anf Journal Backup Data Sets

```
//*************************************************************/
//* THE OCDS MAY NOT EXCEED 1 VOLUME.                         */
//*************************************************************/
//*
//OCDSV1 DD DSN=HSM.OCDS.BACKUP.V0000001,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//OCDSV2 DD DSN=HSM.OCDS.BACKUP.V0000002,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//OCDSV3 DD DSN=HSM.OCDS.BACKUP.V0000003,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//OCDSV4 DD DSN=HSM.OCDS.BACKUP.V0000004,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(15,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//*
//JRNLV1 DD DSN=HSM.JRNL.BACKUP.V0000001,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(100,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//JRNLV2 DD DSN=HSM.JRNL.BACKUP.V0000002,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(100,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//JRNLV3 DD DSN=HSM.JRNL.BACKUP.V0000003,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(100,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
//JRNLV4 DD DSN=HSM.JRNL.BACKUP.V0000004,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=HSM14C,SPACE=(CYL,(100,5)),STORCLAS=SC54GRT,
//   MGMTCLAS=MC54SPEC
```

Figure 24 (Part 3 of 3). Sample JCL to Allocate CDS anf Journal Backup Data Sets

Once this job has run to completion with no errors, use Interactive Storage Management Facility (ISMF) to verify that the data sets have been allocated within the storage and management class that you intended, along with the correct volume. Once you are satisfied that this is the case, add the following to your DFSMShsm ARCCMDxx PARMLIB member:

```
/*********************************************************************/
/* SAMPLE SETSYS CDSVERSIONBACKUP COMMAND AND SUBPARAMETERS THAT     */
/* DEFINE A CDS BACKUP ENVIRONMENT WHERE DSS BACKS UP FOUR COPIES OF */
/* THE CDSS IN PARALLEL TO DASD.                                     */
/*********************************************************************/

SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
   BACKUPCOPIES(4) -
   BACKUPDEVICECATEGORY(DASD) -
   MCDSBACKUPDSN(HSM.MCDS.BACKUP) -
   BCDSBACKUPDSN(HSM.BCDS.BACKUP) -
   OCDSBACKUPDSN(HSM.OCDS.BACKUP) -
   JRNLBACKUPDSN(HSM.JRNL.BACKUP))
/*
```

Figure 25. Sample SETSYS CDSCONVERSIONBACKUP Command and Subparameters

The following describes the backup environment we defined:

- DFSMSdss will back up the CDSs using logical dump.
- The backups will be done in parallel (default when DASD is specified).
- Up to four copies of the CDSs and journal will be kept.
- The DFSMShsm generated names will have a high-level qualifier of HSM.

2.4.6.2 Creating CDS and Journal Backups

When DFSMShsm is started, it gets environmental and function information from PARMLIB. There is no requirement that SYS1.PARMLIB be used. The PARMLIB pointed to by the HSMPARM DD statement in the startup procedure will use the ARCCMD00 member or an alternate member that you indicate by the CMD keyword. The SETSYS, DEFINE, and ADDVOL commands that define the way in which DFSMShsm manages your data are in the PARMLIB member (see PARMLIB member ARCCMD00 in the starter set).

The SETSYS CDSVERSIONBACKUP command determines how DFSMShsm backs up your CDSs. Subparameters of the CDSVERSIONBACKUP command allow you to specify:

- The data mover that backs up the CDSs (DFSMSdss is recommended)
- The number of backup versions to keep for the CDSs
- The device type on which to store the backup versions of the CDSs
- The names of the backup version data sets

By using SMS storage groups and management classes you can:

- Prevent backup (outside CDSVERSIONBACKUP) of the CDSs and the journal.
- Prevent migration of the CDSs and the journal
- Specify that concurrent copy be used to back up the CDSs, assuming that they are on volumes connected to a controller that provides concurrent copy.

2.4.6.3 Steps for Defining the CDS and Journal Backup Environment

If defining a CDS and journal backup environment is new to you, this step-by-step example will guide you through the process so that you will be able to define one that suits your installation needs:

1. Prevent the CDSs and the journal from being backed up as part of user data set backup. The CDSs and the journal are backed up separately as specified by the SETSYS CDSVERSIONBACKUP command.
2. If the CDSs and journal are SMS-managed:
   - Place them on volumes that are defined in a storage group with

     ```
     AUTO BACKUP ===> NO
     ```

     or
   - Associate them with a management class whose attributes are

     ```
     AUTO BACKUP ===> N
     ```

   If the CDSs and the journal are non-SMS-managed, issue the ALTERDS command to prevent them from being backed up outside CDSVERSIONBACKUP.

3. Prevent the CDSs and journal from migrating. Allowing the CDSs and the journal to migrate is inadvisable because you might not be able to recover should any of the CDSs be damaged.

   If the CDSs and journal are SMS-managed:

   - Place them on volumes that are defined in a storage group with

     `AUTO MIGRATE ===> NO`

     or

   - Associate them with a management class whose attributes are

     `COMMAND OR AUTO MIGRATE ===> NONE`

   If the CDSs and journal are non-SMS-managed, issue the SETMIG command to prevent them from migrating.

4. Determine whether your CDSs are backed up by concurrent copy. If you want your CDSs to be backed up by concurrent copy you must:

   - Ensure that they are associated with a management class Backup Copy Technique attribute of Concurrent Required or Concurrent Preferred.
   - Ensure that they are on a DASD volume with a concurrent-copy-capable 3990 controller.
   - Ensure that you specify DATAMOVER(DSS) in step 5.

5. Determine whether the data mover for the CDSs is DFSMShsm or DFSMSdss. We recommend DFSMSdss as the data mover, because DFSMSdss validates the CDSs during backup and supports concurrent copy.

   If you specify:

   `SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS))`

   DFSMShsm invokes DFSMSdss to perform a logical dump of the CDSs and uses sequential I/O to back up the journal. DFSMSdss validates the CDSs while backing them up and uses concurrent copy if it was specified in the management class.

   If you specify:

   `SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM))`

   DFSMShsm exports the CDSs and backs up the journal with sequential I/O. The CDSs are not validated during backup.

6. Choose the number of backup versions you want to keep for the CDSs. The number of backup versions that DFSMShsm keeps is determined by the number you specify on the BACKUPCOPIES subparameter of the SETSYS CDSVERSIONBACKUP command.

   Note: Whenever DFSMShsm actively accesses the CDSs in RLS mode, DFSMSdss must be specified as the datamover for the CDS backup. If data is directed to tape, the PARALLEL parameter must also be specified. If either condition is not met during auto CDS version backup, these values override existing values, and message ARC0793I is issued. If either of these conditions is not met when BACKVOL CDS is issued, the command fails.

7. Choose the device category (DASD or tape) to which you want DFSMShsm to back up your CDSs and journal. Parallel is faster than serial and is required in order to use concurrent copy.

   If you specify:

```
SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(DASD))
```

DFSMShsm always backs up the control data sets in parallel to DASD devices. If you are backing up the CDSs and the journal to DASD, you must preallocate the backup version data sets. You can preallocate the DFSMShsm CDS and journal data set by running the starter set job ALLOCBK1 (see Figure 24 on page 56) before starting DFSMShsm.

If you specify:

```
SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE))
```

DFSMShsm backs up the CDSs to tape.

Whether tape CDS backups are in parallel is determined by the data mover you specify and the optional PARALLEL|NOPARALLEL option for DFSMShsm CDS backup.

If you specify:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS))
```

CDSs are backed up to tape in parallel. Concurrent copy can be used.

If you specify:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM))
```

DFSMShsm backs up the CDSs serially. Concurrent copy is not available, and the CDSs are not validated during backup.

If you specify:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM) PARALLEL)
```

DFSMShsm backs up the control data sets to tape in parallel. Concurrent copy is not available, and the CDSs are not validated during backup.

If you are backing up the CDSs and the journal to tape, DFSMShsm dynamically allocates scratch tape volumes so you need not preallocate backup version data sets.

8. Determine the names for the backup data sets.

   You specify the names that are assigned to the backup version data sets by means of the MCDSBACKUPDSN, BCDSBACKUPDSN, OCDSBACKUPDS, and JRNLBACKUPDSN subparameters of the SETSYS CDSVERSIONBACKUP command. Your backup version data set names can be up to 35 characters (including periods) but cannot end in a period.

Figure 26 on page 62 is an example of the SETSYS CDSVERSIONBACKUP command and its subparameters, as it would appear in PARMLIB member ARCCMDxx.

```
/*******************************************************************/
/* SAMPLE SETSYS CDSVERSIONBACKUP COMMAND AND SUBPARAMETERS THAT   */
/* DEFINE A CDS BACKUP ENVIRONMENT WHERE DSS BACKS UP FOUR COPIES OF */
/* THE CDSS IN PARALLEL TO DASD.                                   */
/*******************************************************************/
/*
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
   BACKUPCOPIES(4) -
   BACKUPDEVICECATEGORY(DASD) -
   MCDSBACKUPDSN(HSM.MCDS.BACKUP) -
   BCDSBACKUPDSN(HSM.BCDS.BACKUP) -
   OCDSBACKUPDSN(HSM.OCDS.BACKUP) -
   JRNLBACKUPDSN(HSM.JRNL.BACKUP))
/*
```

Figure 26. Sample ARCCMDxx PARMLIB Member Showing CDS and Journal Backup Data Set Names

Once all of these steps have been completed successfully, on your next start, DFSMShsm will use these preallocated data sets to hold the backup copies.

2.4.6.4 Backing Up the CDS and Journal Manually

As automatic backup typically produces good copies of the CDSs and journal, it is not usually necessary to back them up manually. However, if you are ever in a position where automatic backup has failed to make successful copies of the CDSs or journal, you can use DFSMShsm commands to create successful copies.

Keep the following points in mind before you issue the command to back up the CDSs and journal manually:

- Do not issue the command during intense DFSMShsm activity as the CDSs cannot change while they are being backed up.

  Note: If you are using concurrent copy, the above is not true.

- Once you have issued the the command, the only way to prevent the backup is to stop DFSMShsm.
- The structural integrity of the CDSs is validated only if you have specified DATAMOVER(DSS).

Use the DFSMShsm BACKVOL command to back up the CDSs and journal. With this command you can:

- Identify the data mover as either DFSMShsm or DFSMSdss.
- Specify the backup device category.
- Specify that you want parallel backup to take place.

To back up the CDSs and journal in parallel to DASD, using DFSMSdss as the data mover, use the following command:

```
BACKVOL CDS(BACKUPDEVICECATEGORY(DASD) DATAMOVER(DSS))
```

With this command, a DFSMSdss logical dump is used, which allows you to use concurrent copy. This may reduce any serialization delays introduced by the

exclusive enqueue that is placed on the CDSs while the backup is taking place. Additionally validation of the CDSs will take place.

If you were to code DATAMOVER(HSM) instead, access method services (AMS) would be invoked to export the CDSs. No structural validation is done.

In all cases the journal is backed up using sequential I/O.

If you want to back up the CDSs and journal to tape, use the following command:

```
BACKVOL CDS(BACKUPDEVICECATEGORY(TAPE(PARALLEL)))
```

When PARALLEL is specified, the default data mover is DFSMSdss, so a DFSMSdss logical dump will be used to back up the CDSs. One tape drive will be allocated for each CDS and the journal.

If you want to back up the CDSs serially, you have to specify TAPE(NOPARALLEL).

Note: If your SETSYS parameters specify DATAMOVER(DSS) or you try to specify it on the BACKVOL CDS command with TAPE(NOPARALLEL), the command will fail.

Only one tape drive is allocated for all the CDS backups.

2.4.7 VSAM Record Level Sharing

DFSMShsm supports VSAM RLS for accessing the CDSs. RLS enables DFSMShsm to take advantage of the features of the coupling facility for CDS access.

Accessing CDSs in RLS mode reduces contention when running primary space management and automatic backup on two or more processors. DFSMShsm benefits from the serialization and data cache features of VSAM RLS and does not have to perform CDS verify or buffer invalidation.

2.4.7.1 Requirements for CDS RLS Serialization

CDSs accessed in RLS mode enqueue certain resources differently from CDSs accessed in non-RLS mode. Before you think about implementing RLS for your CDSs you must make sure that all the following criteria are met:

- GRS or an equivalent function is implemented.
- All operating systems running DFSMShsm must be coupling facility capable, and the processors must share access to the coupling facility.
- A minimum operating system requirement of MVS/ESA 5.2 or OS/390 2.4 and later.
- DFSMS/MVS V1R4

  Note: There is no toleration of DFSMShsm systems lower than DFSMS/MVS V1R4.
- Your CDSs must be SMS-managed.
- All processors in the installation must access the CDSs in RLS mode.
- All DFSMShsm hosts must specify CDSSHR=RLS in the DFSMShsm startup procedure.

- You must not convert the ARCGPA and ARCRJRN reserve to an enqueue for performance reasons.
- The CDS′s storage class must indicate which coupling facility to use.
- The CDSs must not be key-range KSDSs.
- You must know how to implement recovery for RLS data sets.
- You must specify DFSMSdss as the datamover for CDSVERSIONBACKUP.
- If CDS backup is directed to tape, the PARALLEL parameter must be used.

2.4.7.2 Making Your CDSs RLS Eligible

Before CDSs can be accessed in RLS mode, you must define or alter them to be RLS eligible, using the LOG(NONE) attribute. You must do this for all your CDSs (MCDSs, BCDSs, and OCDSs).

The following is an example of how to use the IDCAMS ALTER command to make the CDSs that we defined previously RLS eligible:

```
ALTER HSM.MCDS LOG(NONE)
```

Figure 27 is an example of how we could have used the DEFINE command when we initially set up our DFSMShsm environment. We show only the definition for the MCDS, but the same would have to be done for the other CDSs.

```
DEFINE CLUSTER (NAME(HSM.MCDS) VOLUMES(HSM14A) -
     CYLINDERS(10 5) FILE(HSMMCDS) -
     STORCLAS(SC54GRT) -
     MGMTCLAS(MC54SPEC) -
     RECORDSIZE(435 2040) FREESPACE(0 0) -
     INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
     SPEED BUFFERSPACE(530432) -
     UNIQUE NOWRITECHECK LOG(NONE)) -
     DATA(NAME(HSM.MCDS.DATA) -
     CONTROLINTERVALSIZE(12288)) -
     INDEX(NAME(HSM.MCDS.INDEX) -
     CONTROLINTERVALSIZE(2048)) -
     CATALOG(UCAT.HSM)
```

Figure 27. Sample DEFINE Command for DFSMShsm Environment

You must never use the ALL or UNDO parameters of the LOG keyword. If it ever becomes necessary to change the CDSs back to non-RLS-eligible, use:

```
ALTER HSM.MCDS NULLIFY(LOG)
```

2.4.7.3 Removing Key Range CDSs

The easiest way to remove key range CDSs is to remove the KEYRANGE((...)) parameter from the IDCAMS DEFINE DATA statements that you used to define your CDSs as key range. During startup, DFSMShsm dynamically calculates the key boundaries for each cluster. You can then use the QUERY CONTROLDATASETS command to display both the low and high keys that DFSMShsm calculates for each cluster.

2.4.7.4 Determining the CDS Serialization Technique

If you need to verify the CDS serialization technique that is currently used, use the QUERY CONTROLDATASETS command.

If you are using RLS the messages returned are:

```
ARC0101I QUERY CONTROLDATASETS COMMAND STARTING
ARC0947I CDS SERIALIZATION TECHNIQUE IS RLS
```

2.4.7.5 RLS Implementation

In this section we show you the steps we took to implement VSAM RLS for our CDSs. In addition to the requirements that we detail in 2.4.7.1, “Requirements for CDS RLS Serialization” on page 63, there are other considerations that must be met and assumptions that we make about your system knowledge. You should be familiar with SMS for VSAM RLS, SMS constructs, SMS classes, SMS configuration, and the coupling facility cache and lock structures. We do not recommend that you undertake these steps until you consider the impact that RLS implementation may have on your system.

2.4.7.6 Define the SCHDSs

These are linear data sets that contain information to allow processing if there is a system failure that may affect RLS. They also act as logs for sharing support.

There are size considerations for the SHCDS along with a naming convention that must be adhered to. For comprehensive information about defining these data sets refer to the *DFSMSdfp Storage Administration Reference*. We used the JCL in Figure 28 to allocate the SHCDS.

```
//DEFSHCDS JOB (999,POK),'JON TATE',CLASS=A,MSGCLASS=T,
// NOTIFY=MHLRES5,TIME=1440,REGION=4M
/*JOBPARM L=999,SYSAFF=SC47
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER (NAME(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS1) -
         LINEAR SHR(3 3) VOL(SHCDS1) -
         CYLINDERS(15 15) )
  DEFINE CLUSTER (NAME(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS2) -
         LINEAR SHR(3 3) VOL(SHCDS2) -
         CYLINDERS(15 15) )
  DEFINE CLUSTER (NAME(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS3) -
         LINEAR SHR(3 3) VOL(SHCDS3) -
         CYLINDERS(15 15) )
```

Figure 28. Sample SHCDS Allocation JCL

2.4.7.7 Coupling Facility Cache and Lock Structures

Some changes to your define CFRM policies will be necessary. The DFSMShsm policy needs to be defined. We added the structures for cache and locking to our current CFRM policy definitions, using the administrative data utility IXCMIAPU as shown in Figure 29.

```
//DEFCFRM1  JOB (999,POK),'CFRM',CLASS=A,REGION=4096K,
//            MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=MHLRES5
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
     DATA TYPE(CFRM) REPORT(YES)
     DEFINE POLICY NAME(CFRM19) REPLACE(YES)

        CF NAME(CF01)
               TYPE(009672)
               MFG(IBM)
               PLANT(02)
               SEQUENCE(000000040104)
               PARTITION(1)
               CPCID(00)
               DUMPSPACE(2048)

        CF NAME(CF02)
               TYPE(009672)
               MFG(IBM)
               PLANT(02)
               SEQUENCE(000000040104)
               PARTITION(1)
               CPCID(01)
               DUMPSPACE(2048)

        STRUCTURE NAME(IGWLOCK00)
               SIZE(28600)
               INITSIZE(14300)
               PREFLIST(CF02,CF01)
               REBUILDPERCENT(75)

        STRUCTURE NAME(HSMCACHE1)
               SIZE(64000)
               INITSIZE(32000)
               PREFLIST(CF01,CF02)
               REBUILDPERCENT(75)

        STRUCTURE NAME(HSMCACHE2)
               SIZE(64000)
               INITSIZE(32000)
               PREFLIST(CF02,CF01)
               REBUILDPERCENT(75)
```

Figure 29. CFRM Policy Definition for RLS Access of the DFSMShsm CDSs

Note: The code in Figure 29 does not represent the entire policy data for the CFRM data set. It represents the CFRM policy that specifies the requirements for the DFSMShsm RLS structures.

The coupling facility cache structure names that we chose to use are HSMCACHE1 and HSMCACHE2. The locking structure name is the required name of IGWLOCK00.

2.4.7.8 Alter the SMS Configuration

You have to update the SMS configuration with the coupling facility cache structure names you just defined. Use the ISMF panels. Select option 8 from the ISMF Primary Option Menu for Storage Administrators to display the CDS Application Selection panel (Figure 30).

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                          CDS APPLICATION SELECTION        CF CACHE SETS SAVED
Command ===>

To Perform Control Data Set Operations, Specify:
  CDS Name . . 'SYS1.SC54.SCDS1'
                            (1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:

  7  1. Display       - Display the Base Configuration
     2. Define        - Define the Base Configuration
     3. Alter         - Alter the Base Configuration
     4. Validate      - Validate the SCDS
     5. Activate      - Activate the CDS
     6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
     7. Cache Update  - Define/Alter/Delete CF Cache Sets

If CACHE Display is chosen, Enter CF Cache Set Name . . *
                             (1 to 8 character CF cache set name or * for all)
```

Figure 30. CDS Application Selection Panel for Cache

We select option 7 and define the cache sets that relate to our coupling facility cache structure names (Figure 31 on page 68).

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              CF CACHE SET UPDATE                 PAGE 1 OF 1
Command ===>

SCDS Name  : SYS1.SC54.SCDS1
Define/Alter/Delete CF Cache Sets:       ( 002  Cache Sets Currently Defined )

 Cache Set                     CF Cache Structure Names
 HSM1      HSMCACHE1

 HSM2      HSMCACHE2









More CF Cache Sets to Add?  . . N  (Y/N)
```

Figure 31. CF Cache Set Update Panel

The coupling facility cache structure names must be those that we previously defined (see 2.4.7.7, “Coupling Facility Cache and Lock Structures” on page 66).

2.4.7.9 Storage Class Changes

We took the option of altering the storage class where our CDSs were already defined. The storage class is SC54GRT, and we altered it by selecting option 4 from the Storage Class Application Selection panel (Figure 32).

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                    STORAGE CLASS APPLICATION SELECTION
Command ===>

To perform Storage Class Operations, Specify:
  CDS Name  . . . . . . . 'SYS1.SC54.SCDS1'
                          (1 to 44 character data set name or 'Active' )
  Storage Class Name  . . sc54grt   (For Storage Class List, fully or
                                     partially specified or * for all)
Select one of the following options  :
  4  1. List          - Generate a list of Storage Classes
     2. Display       - Display a Storage Class
     3. Define        - Define a Storage Class
     4. Alter         - Alter a Storage Class
     5. Cache Display - Display Storage Classes/Cache Sets
If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

If Cache Display is Chosen, Specify Cache Structure Name . .
```

Figure 32. Altering a Storage Class

We added the coupling facility cache set information (Figure 33 on page 69).

```
 Panel  Utilities  Scroll  Help

                            STORAGE CLASS ALTER              Page 2 of 2
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Class Name  : SC54GRT

To ALTER Storage Class, Specify:

  Guaranteed Space . . . . . . . . . . Y           (Y or N)
  Guaranteed Synchronous Write . . . N             (Y or N)
  CF Cache Set Name  . . . . . . . . . HSM1        (up to 8 chars or blank)
  CF Direct Weight . . . . . . . . . . 5           (1 to 11 or blank)
  CF Sequential Weight . . . . . . . . 3           (1 to 11 or blank)
```

Figure 33. Storage Class Alter Panel (Page 2 of 2)

To this we added the coupling facility cache set name that we had associated previously with the coupling facility cache structure name. The greater the weight value, the higher the importance that it is cached. We chose our values randomly.

As the CDSs were already in this storage class, we validated and then activated our SMS SCDS of SYS1.SC54.SCDS1.

If your data sets are not already allocated to a storage class that allows RLS, you have to assign them to one.

2.4.7.10 Altering IGDSMSxx Parmlib Member

To specify to SMS that the RLS address space, SMSVSAM, starts at IPL and to include other information, we added the following (Figure 34) to our PARMLIB data set, member IGDSMS54:

```
SMS ACDS(SYS1.SC54.ACDS) COMMDS(SYS1.SC54.COMMDS)
DEADLOCK_DETECTION(15,4)
SMF_TIME(YES) CF_TIME(1800) RLSINIT(YES)
RLS_MAX_POOL_SIZE(100)
```

Figure 34. Sample IDGSMSxx PARMLIB member to Implement RLS

Each MVS system will have its own SMSVSAM address space after IPL if RLSINIT(YES) is coded.

2.4.8 RACF Facility Classes

RACF 2.2 provides two new profiles that you can use to restrict access to certain RLS functions. The two new Facility class profiles that are added are:

- STGADMIN.IGWSHCDS.REPAIR - to use the AMS SHCDS command
- STGADMIN.VSAMRLS.FALLBACK - to use the V SMS,SMSVSAM,FALLBACK command

If you want to limit the access to these commands, you have to set up these profiles and authorize users to them.

2.4.9 Activating the SMSVSAM Address Space

To activate the SMSVSAM address space, an IPL of the MVS system is necessary. To verify that the SMSVSAM address space has started on your system, issue the following command:

```
D SMS,SMSVSAM,ALL
```

The information that is returned will tell you whether your system is active and what the status is of the SMS complex.

2.4.9.1 Activating the SHCDS

If your SHCDS is already active, there is no need to activate it. However if this is your first implementation of RLS, you have to issue the following commands modified to represent the names you have chosen for your SHCDSs:

```
VARY SMS,SHCDS(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS1),NEW
VARY SMS,SHCDS(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS2),NEW
VARY SMS,SHCDS(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS2),NEWSPARE
```

These commands:

- Activate the new primary SHCDS
- Activate the new secondary SHCDS
- Activate a spare SHCDS

2.4.9.2 DFSMShsm Procedure Changes

To your DFSMShsm started procedure member in SYS1.PROCLIB you have to add the following to the PROC statement:

```
CDSSHR=RLS
```

and you must add the following to your EXEC statement:

```
'CDSSHR=&CDSSHR'
```

2.4.9.3 RLS Implementation Checklist

To review, these are the steps that you need to follow to implement RLS access for the DFSMShsm CDSs:

1. Make sure that all sharing systems are at the prerequisite levels of software.
2. If not already defined, use IDCAMS to define SHCDSs and make sure that SHAREOPTION(3,3) is specified.
3. Define your coupling facility cache and lock structures.
4. Alter your SMS configuration to include the cache set information.
5. Create or alter a storage class that contains the cache set information.

6. Assign the CDSs to the new or altered storage class.
7. Alter SYS1.PARMLIB member IGDSMSxx with the RLS parameters.
8. Define the new RLS profiles in the RACF Facility class and authorize users.
9. Schedule an IPL of the MVS systems and make sure that the SMSVSAM address space becomes active.
10. Activate the SHCDSs.

2.5 Multicluster Control Data Sets

As the DFSMShsm workload increases in your installation, so will the activity that each of the CDSs is required to record. As CDSs grow, they can require more space than the physical DASD devices allow (for example, the capacity of a 3390-3 is 2.8 GB). If this occurs, the CDSs can be split across multiple volumes to accommodate their larger size. Only the MCDS and BCDS can be more than one cluster.

If, however, your site has not experienced any problems with the space requirements of your CDSs, we do not recommend that you implement a multicluster environment. If you have experienced space problems you may want to first try increasing the size of your CDSs to relieve the space problem. If you find that the space required for your CDS requires more than one volume, you need to split your CDS.

Below we show how you can define multicluster key-range and non-key-range data sets with the help of sample JCL and DFSMShsm tools.

2.5.1 Key-Range or Non-Key-Range

VSAM RLS does not support key-range KSDSs; however, DFSMShsm supports non-key-range data sets. For more information about RLS and the CDSs, refer to 2.4.7, “VSAM Record Level Sharing” on page 63. In 2.5.4, “Conversion Steps” on page 72 we include sample JCL to define RLS-elgible non-keyrange data sets. If you want to create RLS-elgible multicluster CDSs, ignore the JCL for the key range data sets and modify the non-key-range JCL to match your installation′s requirements.

2.5.2 Determining Key Ranges

First item you have to decide is where to split the CDSs so that an even division of the DFSMShsm control records occurs. DFSMShsm provides a tool that will analyze your current CDS data in all clusters and produces a report that shows the split ranges for two, three, and four clusters.

If you decide that you would like to implement multicluster CDSs, we recommend that you first analyze your existing CDSs with the HSM.SAMPLE.TOOL(SPLITCDS) job found in SYS1.SAMPLIB.

Note: HSM.SAMPLE.TOOL, which contains member SPLITCDS, is created by running ARCTOOLS, which resides in SYS1.SAMPLIB.

2.5.3 Multicluster Performance Considerations

We recommend that if possible you attempt to keep each cluster on separate DASD subsystems to improve performance and to reduce recovery time should it be necessary.

2.5.4 Conversion Steps

Once you have decided that you want to implement a multicluster CDS, there are a number of steps to take. The steps described below assume a familiarity with DFSMShsm and that there is a requirement at your site to adopt multicluster CDSs.

Follow these steps to implement a multicluster environment:

1. Stop DFSMShsm on all MVS images.
2. Back up the CDS with the AMS EXPORT command.
3. Determine whether to split the CDS into two, three, or four clusters.
4. Determine the appropriate key-ranges for each cluster.

   The HSM.SAMPLE.TOOL(SPLITCDS) job, provides the key-ranges for each cluster. If you are implementing non-key-range data sets, DFSMShsm will work out the ranges, so this is not a necessary step.
5. If you are defining a key-range multicluster, modify the JCL in Figure 35 on page 73 to suit your installation′s requirements.
6. If you are defining a non-key-range multicluster, modify the JCL in Figure 36 on page 74 to suit your installation′s requirements.

   Note: In both cases we show only the JCL to convert the MCDS.

```
/*********************************************************************/
/*  SAMPLE JOB THAT ALLOCATES A KEY-RANGE MULTICLUSTER               */
/*  MIGRATION CONTROL DATA SET.                                      */
/*********************************************************************/
//HSMCDS JOB MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
/*********************************************************************/
/*                     FIRST VOLUME DEFINITION                       */
/*********************************************************************/
      DEFINE CLUSTER (NAME(HSM.MCDS1) -
            CYLINDERS(10 5) -
            STORCLAS(SC54GRT) -
            MGMTCLAS(MC54SPEC) -
            RECORDSIZE(435 2040) FREESPACE(0 0) -
            INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
            SPEED BUFFERSPACE(530432) -
            UNIQUE NOWRITECHECK) -
            CATALOG(UCAT.HSM)
      DATA -
            (NAME(HSM.MCDS1.DATA) -
            KEYRANGE((X'00' HSM.HMIG.T4)) -
            VOLUMES(MCDS01) -
            CONTROLINTERVALSIZE(12288)) -
      INDEX -
            (NAME(HSM.MCDS1.INDEX) -
            VOLUME(MCDS01) -
            CONTROLINTERVALSIZE(2048) -
            CATALOG(UCAT.HSM)
 /*********************************************************************
 /*                     SECOND VOLUME DEFINITION                      *
 /*********************************************************************
      DEFINE CLUSTER (NAME(HSM.MCDS2) -
            CYLINDERS(10 5) -
            STORCLAS(SC54GRT) -
            MGMTCLAS(MC54SPEC) -
            RECORDSIZE(435 2040) FREESPACE(0 0) -
            INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
            SPEED BUFFERSPACE(530432) -
            UNIQUE NOWRITECHECK) -
            CATALOG(UCAT.HSM)
      DATA -
            (NAME(HSM.MCDS2.DATA)
            KEYRANGE((HSM.HMIG.T5 X'FF')) -
            VOLUME(MCDS02) -
            CONTROLINTERVALSIZE(12288)) -
      INDEX -
            (NAME(HSM.MCDS2.INDEX)
            VOLUME(MCDS02) -
            CONTROLINTERVALSIZE(2048) -
            CATALOG(UCAT.HSM)
```

Figure 35. Sample JCL for Key-Range Multicluster MCDS

```
//HSMCDS   JOB ,MSGLEVEL=(1,1)
//*****************************************************************/
//* SAMPLE JCL THAT ALLOCATES NON-KEY-RANGE MULTICLUSTER           */
//* RLS-ELIGIBLE MIGRATION CONTROL DATA SETS.                      */
//*****************************************************************/
//*
//STEP1    EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
        DEFINE CLUSTER (NAME(HSM.RLS.MCDS1) -
           STORAGECLASS(SC54GRT) -
           CYLINDERS(2) NOIMBED NOREPLICATE -
           RECORDSIZE(200 2040) FREESPACE(0 0) -
           INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
           UNIQUE LOG(NONE)) -
        DATA -
           (NAME(DFHSM.RLS.MCDS1.DATA) -
            CONTROLINTERVALSIZE(4096)) -
        INDEX -
           (NAME(DFHSM.RLS.MCDS1.INDEX) -
            CONTROLINTERVALSIZE(4096))

        DEFINE CLUSTER (NAME(DFHSM.RLS.MCDS2) -
            STORAGECLASS(SC54GRT) -
            CYLINDERS(2) NOIMBED NOREPLICATE -
            RECORDSIZE(200 2040) FREESPACE(0 0) -
            INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
            UNIQUE LOG(NONE)) -
        DATA -
           (NAME(DFHSM.RLS.MCDS2.DATA) -
           CONTROLINTERVALSIZE(4096)) -
        INDEX -
           (NAME(DFHSM.RLS.MCDS2.INDEX) -
           CONTROLINTERVALSIZE(4096))
```

Figure 36. Sample JCL for Non-Key-Range Multicluster MCDS

7. Copy the old CDS to the new multicluster CDS with the AMS REPRO command, as shown in Figure 37 on page 75 for key-range multiclusters and Figure 38 on page 75 for non-key-range multiclusters.

```
/*****************************************************************/
/* SAMPLE JOB THAT COPIES THE OLD CDS INTO THE NEWLY DEFINED KEY-  */
/* RANGE MULTICLUSTER CONTROL DATA SET WITH THE AMS REPRO COMMAND. */
//* NOTE: THE FROMKEY/TOKEY VALUES ARE ONLY SAMPLES.  THE ACTUAL   */
//* PARAMETERS USED FOR THESE KEYWORDS SHOULD BE DERIVED FROM      */
//* ACTUAL CDSS BEING USED.                                        */
/*****************************************************************/
/*
//STEP2    EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
   REPRO INDATASET(HSM.MCDS) OUTDATASET(HSM.MCDS1) -
         FROMKEY(X'00') TOKEY(HSM.HMIG.T4)
   REPRO INDATASET(HSM.MCDS) OUTDATASET(HSM.MCDS2) -
         FROMKEY(HSM.HMIG.T5) TOKEY(X'FF')
/*
```

Figure 37. Sample JCL for Key-Range Multicluster REPRO

```
//*****************************************************************/
//* COPY THE OLD CONTROL DATA SETS INTO THE NEWLY DEFINED          */
//* NON-KEY-RANGE MULTICLUSTER CONTROL DATA SETS.                  */
//* NOTE: THE FROMKEY/TOKEY VALUES ARE ONLY SAMPLES.  THE ACTUAL   */
//* PARAMETERS USED FOR THESE KEYWORDS SHOULD BE DERIVED FROM      */
//* ACTUAL CDSS BEING USED.                                        */
//*****************************************************************/
//*
//STEP2    EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
   REPRO INDATASET(HSM.MCDS) OUTDATASET(HSM.RLS.MCDS1) -
      FROMKEY(X'00') TOKEY(MIDDLE.KEY1)
   REPRO INDATASET(HSM.MCDS) OUTDATASET(HSM.RLS.MCDS2) -
      FROMKEY(MIDDLE.KEY2) TOKEY(X'FF')
/*
```

Figure 38. Sample JCL for Non-Key-Range Multicluster REPRO

8. Modify your DFSMShsm startup procedure in SYS1.PROCLIB and in other JCL (such as DCOLLECT and ARCIMPRT) that references the multicluster CDS. There must be a separate DD card for each cluster of a multicluster CDS. The ddnames should be MIGCAT, MIGCAT2, MIGCAT3, and MIGCAT4 or BAKCAT, BAKCAT2, BAKCAT3, and BAKCAT4.
9. If you back up your CDSs to DASD, preallocate new CDS backup data sets. You need backup versions for each cluster in the CDS.

   Note: Do not delete the current CDS. Instead, maintain it for a period of time until you determine that the new CDS is valid.
10. Make backup copies of the CDSs.
11. Monitor the growth of the multicluster CDS.

Once all these steps have completed successfully, DFSMShsm will use the multicluster CDSs.

2.6 CDS Backup Procedures

You can backup DFSMShsm CDSs in a variety of ways, depending on your hardware and system level. Here we discuss using SnapShot and virtual concurrent copy as your backup procedures.

2.6.1 Virtual Concurrent Copy

Concurrent copy, a function of DFSMSdss and the IBM 3990-6 control unit, enables you to take a consistent copy or dump of your data while applications update the data. The DFSMShsm CDSs can be backed up with concurrent copy. Virtual concurrent copy provides a copy operation similar to concurrent copy by using a combination of RVA, SnapShot, and DFSMSdss. If you are already using concurrent copy to back up the CDSs, and the device they reside on is an RVA, you do not have to take any additional action. Virtual concurrent copy will be automatically invoked.

You invoke virtual concurrent copy by specifying the CONCURRENT keyword on a DFSMSdss COPY or DUMP statement. When the CONCURRENT keyword is specified on a DFSMSdss COPY command and all the requirements for DFSMSdss SnapShot are met, DFSMSdss tries to perform a DFSMSdss SnapShot. In this case the logical and physical completion of the copy occur at the same time.

The benefit of virtual concurrent copy is that it can be performed in the following circumstances, when a DFSMSdss SnapShot cannot:

- The data needs to be manipulated (DUMP command used, reblocked, track packed to unlike) and the source resides on an RVA.
- The source and target reside on different RVAs, and DFSMSdss is the data mover.
- The source resides on an RVA, the target is a non-RVA device, and DFSMSdss is the data mover.
- A multivolume source data set spans multiple RVAs, and DFSMSdss is the data mover.
- A multivolume source data set resides on a combination of RVAs and devices connected to an IBM 3990-6 control unit, and DFSMSdss is the data mover.

To perform a virtual concurrent copy, space must be made available in the same RVA subsystem as the source data set. During a virtual concurrent copy, data is ″snapped″ from its source location to an intermediate location. This intermediate location is known as the working space data set (WSDS). Data is gradually copied from the WSDS to the target location.

So for virtual concurrent copy to be successful, even though the concurrent copy criteria have been met, you must have predefined at least one WSDS, but it is more likely that you will have to allocate multiple WSDSs.

The allocation requirements, catalog search, system data mover (SDM), recommendations, performance, and restrictions for WSDSs are

comprehensively covered in the redbook *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268. We recommend that you obtain a copy of the redbook before you implement virtual concurrent copy in your installation.

2.6.2 SnapShot

We document here a procedure for using SnapShot. to back up the DFSMShsm CDSs and journal residing in the same RVA subsystem and partition. If the DFSMShsm CDSs and journal do not reside in the same RVA subsystem and partition, the data mover utility of DFSMSdss will be called, and a conventional copy process will be used. Use the procedure documented here if you are not in a position to take advantage of virtual concurrent copy:

1. Issue HOLD ALL (optional)

   To get DFSMShsm into a quiesced state before you can issue SETSYS EMERGENCY, issue HOLD ALL on all hosts sharing the CDSs.

2. Issue SETSYS EMERGENCY

   SETSYS EMERGENCY prevents DFSMShsm from performing any data movement functions. Issue the command on all hosts sharing the CDSs.

3. Wait two minutes for EMERGENCY to take effect and the journal queue to drain.

4. Issue either data set SnapShot or volume SnapShot for the CDSs and journal.

   - If you are using data set level SnapShot, you must specify HOSTCOPYMODE(SHARED) parameter.
   - If you are using volume level SnapShot, you must specify TOLERATEENQFAILURE(YES).

   If you are using volume level SnapShot, we recommend that the CDSs and journal be on the same volume or that each individual CDS and journal be on its own unique volume. Figure 39 is an example of the SYSIN JCL that you could modify to perform a snap of your MCDS:

```
SNAP DATASET(SOURCE(HSM.MCDS) -
             TARGET(NEW.CDS) VOL(WP11S1) -
             HOSTCOPYMODE(SHARED) -
             DMNM(NONE) CAT(YES) REPL(YES))
```

Figure 39. Sample SYSIN JCL to Snap the DFSMShsm MCDS

5. Issue PATCH .MCVT.+3 BITS(.1.....)

   This command inhibits journaling so the BACKVOL CDS(NULLJOURNALONLY) command can be issued.

6. Issue BACKVOL CDS(NULLJOURNALONLY)

   This command nulls the journal contents.

7. Issue PATCH .MCVT.+3 BITS(.0.....)

   This command turns off the journal inhibit bit and restarts CDS update journaling.

8. Issue SETSYS NOEMERGENCY

This command allows data movement to be reactivated. Issue it on all hosts sharing the CDSs.

9. Issue RELEASE ALL (optional)

   If you issued HOLD ALL previously, you have to issue RELEASE ALL, on all hosts sharing the CDSs.

You may want to invoke a DFSMSdss dump job to dump the backup copies of the CDSs to tape. This is one of the advantages of SnapShot: once the copy has been taken, you can use it as input to a job that dumps it to tape, thereby minimizing any impact on the CDSs.

2.7 Space Management

In this section we present the parameters required to define the characteristics of DFSMShsm space management. Space management identifies a set of functions, such as:

- Migration from level 0 to ML1
- Migration from ML1 to ML2
- Migration from level 0 to ML2
- Recall from ML1 and ML2 to level 0
- Extent reduction
- Release of allocated but unused space
- Scratch obsolete or expired data sets
- Clean up obsolete records from CDSs and DFSMShsm volumes

Figure 40 represents the flow of data among these functions.

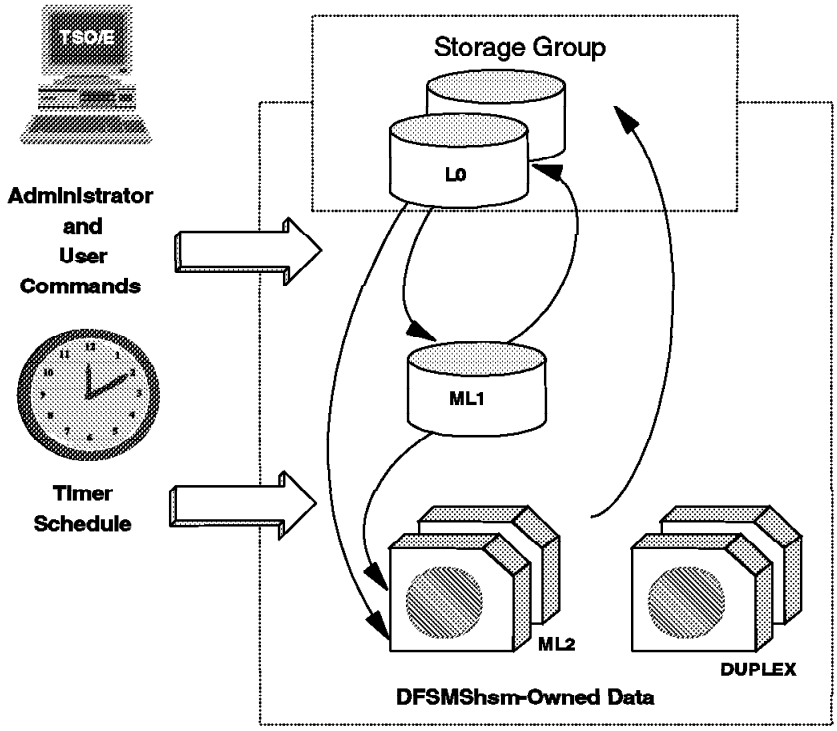


Figure 40. DFSMShsm Space Management Overview

Consider the following parameters when setting up the DFSMShsm space management options:

1. **Do you want to migrate data sets to ML2 tape?**

   If you plan to migrate the data sets on ML1 to ML2 tape, use statement (a) below. If you want your non-SMS-managed data sets on DFSMShsm-managed volumes to migrate directly to tape, use statement (b). If you do not want to use tapes for data migration, use statement (c). In the latter case, ML2 tape is selected only when the data set belongs to a management class indicating that the migration should be directly to ML2 tape or the ARCMDEXT exit redirects data sets to ML2 tape. Management class attribute `LEVEL-1-DAYS-NON-USAGE = NOLIMIT` prohibits SMS-managed data sets from migrating to ML2 tape.

   Note that the selection of one of these statements must be consistent with the contents of the SMS management classes implemented in the installation.

```
(a) SETSYS TAPEMIGRATION(ML2TAPE)

(b) SETSYS TAPEMIGRATION(DIRECT)

(c) SETSYS TAPEMIGRATION(NONE)
```

2. **Do you want to use a special unit to allocate tapes for migration?**

This parameter is related to the selection of the previous parameter. If you want DFSMShsm to always allocate the ML2 output unit with a given unit name, specify either statement (a), (b,) or (c) below. If neither (a), (b), nor (c) is specified and you want DFSMShsm to use a special unit type when a nonspecific allocation is made during ML2 migration, specify statement (d). If you specify an esoteric name, code it in the SETSYS USERUNITTABLE parameter, as explained in 2.2.3, “Setting Up the Base SETSYS Parameters” on page 25.

```
(a) SETSYS TAPEMIGRATION(DIRECT(TAPE(unittype)))

(b) SETSYS TAPEMIGRATION(ML2TAPE(TAPE(unittype)))

(c) SETSYS TAPEMIGRATION(NONE(ROUTETOTAPE(TAPE(unittype))))

(b) SETSYS MIGUNITNAME(unitname)
```

3. **What minimum level of valid data is required before recycling an ML2 tape?**

   If you want to use a different percentage value for ML2 than you use for backup tapes, you must use this parameter, which specifies the maximum percentage (pct) of valid data on a migration tape that causes it to be processed by the recycle function. Substitute a number from 0 to 99. The RECYCLE command issuer can override it when actually requesting this function.

```
SETSYS ML2RECYCLEPERCENT(pct)
```

4. **How many recalls do you want to run concurrently?**

   Substitute a decimal number from 1 to 15 for the number of concurrent recall tasks. You can specify 15 as a good starting point.

```
SETSYS MAXRECALLTASKS(xx)
```

5. **Do you want to specify the number of tasks allowed to recall data sets from tape?**

   Specify the maximum number (xx) of tape recall requests that DFSMShsm should handle. The number must be lower than or equal to the value specified with SETSYS MAXRECALLTASKS and lower than the number of tape drives you have installed, because each tape recall task requires a dedicated tape unit.

```
SETSYS TAPEMAXRECALLTASKS(xx)
```

6. **How many automatic volume migration tasks do you want to run in parallel?**

DFSMShsm can run up to 15 tasks in parallel on each host. Each manages one volume at a time and requires a dedicated tape unit when automatic volume migration is directed to tape. In this case you need to specify a number that is less than or equal to the number of tape drives you have available; if you duplex your migration tapes, this number must be half the number of available tape drives, at maximum. Choose the number of tasks (xx) you want to run and use the following statement:

```
SETSYS MAXMIGRATIONTASKS(xx)
```

7. **How many interval migration tasks do you want to run in parallel?**

   This parameter applies to the processing of primary volumes and SMS-managed volumes. It specifies the maximum number of interval migration tasks that can run concurrently. In a direct-to-tape environment, you must have a sufficient number of tape drives to support this number of tasks. The value that you specify with this parameter does not have any relationship with the SETSYS MAXMIGRATIONTASKS value. Choose the number of tasks (xx) you want to run and use the following statement:

```
SETSYS MAXINTERVALTASKS(xx)
```

8. **When do you want to start the automatic space management functions?**

   Specify the planned start (hhmm1) and stop (hhmm2) time for running primary space management functions. If you specify hhmm1 as zero (0000), DFSMShsm will not start automatic primary space management. For cycle, substitute a string of alphabetic Ys and Ns. Each of them represents a day in the cycle. Specify Y as the day primary space management is to run. Specify N as the day when automatic primary space management is not to run. You can represent up to a 31-day cycle with the Ys and Ns. The CYCLESTARTDATE subparameter specifies the start date of a cycle. However, the date (yyyy/mm/dd) cannot be a date in the future. The first cycle day begins at the start time of this function regardless of when the new calendar day begins.

   Processing of automatic secondary space management can be similarly specified. The primary and secondary space management functions can have independent cycles, dates, and times of execution on each system in the enterprise.

   The choice between activating or not activating this function on the system must be consistent with the migration and system name information in the SMS storage groups.

   We recommend you run secondary space management before primary space management to ensure space on ML1 volumes.

```
SETSYS PRIMARYSPMGMTSTART(hhmm3 hhmm4)
DEFINE PRIMARYSPMGMTCYCLE(cycle CYCLESTARTDATE(yyyy/mm/dd))

SETSYS SECONDARYSPMGMTSTART(hhmm1 hhmm2)
DEFINE SECONDARYSPMGMTCYCLE(cycle CYCLESTARTDATE(yyyy/mm/dd))
```

9. **Which high-level qualifier do you want to use for the name of DFSMShsm migrated data sets?**

   The high-level qualifier can be used as a RACF generic profile to protect these copies against unauthorized access. If you do not specify this parameter on any SETSYS command, the prefix for the migrated data sets is the UID that you specified in the DFSMShsm startup procedure. In our installation, we used UID=HSM54 in our DFSMShsm startup procedure, so our migrated data sets prefix is HSM54.

```
SETSYS MIGRATEPREFIX(prefix)
```

10. **Do you want to perform hourly migration outside the primary space management window?**

    The purpose of interval migration is to prevent the DFSMShsm-managed volumes from running out of space during the day. DFSMShsm normally does a space check every hour. Therefore, if you request interval migration, DFSMShsm uses this hourly space check to determine on which volumes interval migration is done. The interval migration will then occur on the volumes that have exceeded their maximum occupancy threshold, which you specified in the storage group. Use statement (a), if you want to use interval migration. Otherwise, use statement (b).

```
(a) SETSYS INTERVALMIGRATION

(b) SETSYS NOINTERVALMIGRATION
```

11. **How long do you want to keep obsoleted records in the MCDS?**

    Specify the number of days that you want to keep the MCDS records for recalled data sets (recalldays) and the number of days that you want to keep the daily and volume statistics records (statdays) before DFSMShsm deletes them during the secondary space management function. For recalldays, substitute a number from 0 to 999; for statdays, substitute a number from 1 to 999.

```
SETSYS MIGRATIONCLEANUPDAYS(recalldays statdays)
```

12. **Do you want to scratch expired data sets?**

    If you select statement (a) below, DFSMShsm scratches expired data sets that have explicit expiration dates. The expiration attributes in the

management class control the expiration of data sets that do not have an expiration date in their data set volume table of contents (VTOC) entry. However, data sets with an expiration date in the data set VTOC entry are not controlled by the management class and require this SETSYS parameter to control their deletion. Statement (b) specifies that DFSMShsm ignore the expiration date during space management processing. NOSCRATCH is the default value.

```
(a)  SETSYS EXPIREDDATASETS(SCRATCH)

(b)  SETSYS EXPIREDDATASETS(NOSCRATCH)
```

13. **What is the maximum number of extents desired for a non-VSAM data set?**

    Specify the maximum number of extents desired for a non-VSAM data set in statement (a) below. When this number is reached, during primary space management, DFSMShsm migrates and recalls the data set performing an extent reduction. The value 10 is a good starting point.

    If you do not want DFSMShsm to do extent reduction, use statement (b).

```
(a) SETSYS MAXEXTENTS(xx)

(b) SETSYS MAXEXTENTS(0)
```

14. **Do you want to use the SDSP facility for the migrated data sets?**

    If you want to use the SDSP facility, use statement (a). We recommend using a size of 60 KB, which represents one track of both 3390 and 3380 DASD. The 60 KB size is valid if you use RVA for ML1 volumes. Specifying SMALLDATASETPACKING requires that you define at least a number of SDSP data sets equal to the maximum number of concurrent volume migration tasks that could be running in your complex. SDSP data sets are allocated on ML1 volumes and defined in the ADDVOL command for the volume on which it resides. The SDSP data sets should be reorganized periodically to prevent out of space conditions.

    Use statement (b) if you do not want to use the SDSP facility.

```
(a) SETSYS SMALLDATASETPACKING(KB(60))

(b) SETSYS NOSMALLDATASETPACKING
```

15. **Do you want to use a particular tape unit to recycle migration tapes?**

    Select the unit type you want to use to allocate output tapes during a migration tape recycle. If you want to use an esoteric, you must also define it with the SETSYS USERUNITTABLE parameter. If you want to use an SMS-managed tape library for RECYCLE output, make sure your ACS routines were coded to accept this unittype as valid for SMS allocations.

```
SETSYS RECYCLEOUTPUT(MIGRATION(unittype))
```

16. **Do you want DFSMShsm to use software compaction for data sets migrated to DASD?**

    We recommend that you specify statement (a) in order to use software compaction for data sets when migrating to DASD (ML1), unless you plan to have RVA DASD in your ML1 pool. Specify statement (b), if you do not want software compaction to occur.

```
(a) SETSYS COMPACT(DASDMIGRATE)

(b) SETSYS COMPACT(NODASDMIGRATE)
```

17. **Do you want DFSMShsm to use software compaction for tape migrated data?**

    If you do not want software compaction to occur when migrating directly to tape, use statement (a). This is the recommended choice when IDRC is available on the tape units used by DFSMShsm. If you want DFSMShsm to do software compaction when migrating directly to tape from level 0 volumes, use statement (b). Note that, if IDRC is to be used on 3480X devices, you also have to specify TAPEHARDWARECOMPACT.

```
(a) SETSYS COMPACT(NOTAPEMIGRATE)

(b) SETSYS COMPACT(TAPEMIGRATE)
```

18. **Do you want to extend partially filled tapes?**

    If you coded this parameter with the base set of options, do not code it here.

    If you want DFSMShsm to extend migration tapes that were not completely written when used last, use statement (a) below. Otherwise, use statement (b). Note that the MARKFULL option can waste a large amount of tape capacity if Magstar cartridges are being used.

    When a migration tape volume is in use by a migration or recycle task, and a recall for a data set in this tape is issued, the tape is taken away from the output task, as the recall has priority. A migration tape taken away because of a data set recall is not related to the PARTIALTAPE option. With APAR OW30676, a taken away tape is not marked full.

    Note: The MARKFULL option can waste a large amount of capacity if Magstar cartridges are being used.

```
(a) SETSYS PARTIALTAPE(MIGRATION(REUSE))

(b) SETSYS PARTIALTAPE(MIGRATION(MARKFULL))
```

19. **What is the maximum amount of partial ML2 tapes that you expect?**

Statement (a) below specifies the desired number of partial (not empty, not full) ML2 tapes not associated with any task that you will have after a generic RECYCLE request for the ML2 tapes. Partial tapes are tapes that were *taken away* from a migration or recycle process because of data set recall. DFSMShsm does not recycle not-associated tapes to get below this number. Even with not-associated tapes able to be processed by recycle, DFSMShsm recycles only tapes that meet the selection criteria, such as percent valid of reuse capacity.

The goal value should reflect the level of recall take away and recall processing across your HSMplex. The default value of 10 is normally a reasonable balance between not filling tapes and increasing the recycle work. You can issue the LIST TTOC SELECT(NOTASSOC) command at any time to get the current number of not-associated ML2 partial tapes.

If you want DFSMShsm to process all partial tapes during a generic RECYCLE, specify statement (b); DFSMShsm will recycle all tape volumes that are not in use at that moment. Statement (c) tells DFSMShsm not to recycle any ML2 partial tapes.

The ML2PARTIALSNOTASSOCIATEDGOAL parameter of the SETSYS command was introduced by APAR OW30676.

```
(a) SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(15)

(b) SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(0)

(c) SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(NOLIMIT)
```

20. **How do I define an ML1 volume to DFSMShsm?**

    As ML1 volumes are non-system-managed volumes, the definition must be done with the ADDVOL command, implemented in start procedure member ARCCMDxx of SYS1.PARMLIB.

    Define a high threshold for the percentage of occupied space in the THRESHOLD parameter. This value is used during automatic secondary space management to trigger migration of ML1 data sets to ML2. If you use the SDSP function, specify statement (a) below. If you choose not to use SDSP, specify statement (b).

```
(a) ADDVOL volser UNIT(unit)                                    -
          MIGRATION(ML1 SMALLDATASETPACKING AUTODUMP(class))    -
          THRESHOLD(xx)

(b) ADDVOL volser UNIT(unit)                                    -
          MIGRATION(ML1 NOSMALLDATASETPACKING AUTODUMP(class)) -
          THRESHOLD(xx)
```

21. **Do you want to define an ML2 volume to DFSMShsm?**

    If you want to use scratch tapes for ML2 volumes, you need not identify the volumes to DFSMShsm. However, if you use DASD ML2 volumes or tape volumes, you must identify them to DFSMShsm. In the following example,

we are adding a 3490 tape cartridge to the DFSMShsm tape pool as an ML2 tape volume:

```
ADDVOL volser UNIT(3490)                          -
       MIGRATION(MIGRATIONLEVEL2)
```

2.8 Availability Management

DFSMShsm availability management is a function that facilitates the storage and retrieval of usable copies of data should online copies become lost or damaged in any way. We show you the basic DFSMShsm settings that allow you to decide how best to manage your data, namely, how you can:

- Make daily incremental backup copies of changed and original data sets
- Periodically dump DFSMShsm-managed and ML1 volumes
- Use command availability management outside the automatic functions
- Perform individual data set and volume restores
- Perform backup and recovery of a predefined set of data

The tasks for controlling the availability management of SMS-managed storage are accomplished by adding DFSMShsm commands to the ARCCMDxx member and by specifying attributes in the SMS storage groups, storage classes, and management classes. We assume that you are familiar enough with SMS to access the ISMF panels to change the various attributes and that you have update authority to them.

We show you the attributes that need to be changed to implement a basic availability management environment along with descriptions of the DFSMShsm commands that need to be added to the ARCCMDxx member.

2.8.1 Getting Started with Backup Availability Management

In this section we introduce you to some of the basic commands, functions, and terminology associated with backup availability management.

1. **What do you mean by backup?**

   Backup is the process of copying a data set from a level 0 or an ML1 volume to a daily backup volume. This copy is referred to as the *backup version*.

2. **What is the difference between backup and dump?**

   The backup function operates at a data set level, whereas the dump function backs up the entire allocated space on the volume.

3. **How do I specify whether automatic backup is done or not?**

   The AUTO BACKUP attribute of the storage group lets DFSMShsm know that you want it to perform automatic incremental backup on the volumes assigned to that storage group. If you specify:

```
Auto Backup  . . Y  (Y or N)
```

automatic backup is eligible to run on the volumes in the storage group.

4. **Does Auto Backup ensure that all the data sets get backed up?**

   No. If the volume itself is eligible for automatic backup, data sets are backed up according to the specifications in their management class. If the volume is not in a storage group that is eligible for automatic backup, the backup attributes for data sets on that volume are ignored. The storage group attributes and the management class attributes work together. For this reason we always point out in which class a particular attribute exists and whether it interacts with another attribute.

5. **Which backup attributes are available in the management class?**

   Take for example the following attributes for one of the management classes, MC54PRIM, that we have defined:

```
                        MANAGEMENT CLASS ALTER            Page 3 of 5
Command ===>

SCDS Name . . . . . . : SYS1.SC54.SCDS1
Management Class Name : MC54PRIM

To ALTER Management Class, Specify:
  Backup Attributes
    Backup Frequency  . . . . . . . . . 1        (0 to 9999 or blank)
    Number of Backup Vers . . . . . . . 2        (1 to 100 or blank)
      (Data Set Exists)
    Number of Backup Vers . . . . . . . 1        (0 to 100 or blank)
      (Data Set Deleted)
    Retain days only Backup Ver . . . 30       (1 to 9999, NOLIMIT or blank)
      (Data Set Deleted)
    Retain days extra Backup Vers . . 5        (1 to 9999, NOLIMIT or blank)
    Admin or User command Backup  . . BOTH     (BOTH, ADMIN or NONE)
    Auto Backup . . . . . . . . . . . Y        (Y or N)
    Backup Copy Technique . . . . . . S        (P=Conc Preferred, R=Conc
                                               Required or S=Standard)
```

Figure 41. Management Class Backup Attributes

Without going into much detail at all, what this means for the management class MC54PRIM is that:

- On the days that automatic backup runs, one day must elapse before it is backed up again and it has changed since the last backup.
- Two backup versions are retained while the data set exists.
- One backup version is retained for 30 days after the data set is deleted.
- Extra backup versions are retained for five days after they are created.
- Any users can issue commands to back up data sets.
- Automatic backup is to process eligible data sets.
- Concurrent copy is not used.

6. **What do the individual parameters in that management class do?**

   Each parameter works closely with one another to ensure that a backup copy of your data is available should you ever need it. There is no global setting that is applicable to all sites, so you must make a decision, based on your user's requirements, as to the management class definitions that are most appropriate.

7. **How do I specify the frequency of backup and what does it mean?**

   The management class backup attribute BACKUP FREQUENCY determines how frequently changed data sets are backed up automatically. It specifies the amount of days that must have elapsed since the last backup. In our example we will back up the data set every time the volume is processed for automatic backup if it has changed.

8. **Is there a way that I can guarantee the frequency of backup?**

   Use the storage group attribute, GUARANTEED BACKUP FREQUENCY. By using this parameter you can also speed up volume recovery from data set backups by reducing the total number of tapes that have to be mounted to recover the volume.

   Figure 42 shows the panel where you can alter the guaranteed backup frequency parameter.

```
                          POOL STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Group Name  : SG54HSM
To ALTER Storage Group, Specify:
 Description ==> STORAGE GROUP FOR DFSMSHSM DATA SETS.
             ==>
 Auto Migrate . . Y  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . Y  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . Y  (Y or N)         Dump Sys/Sys Group Name  . . .

 Dump Class . . . ONSITE              (1 to 8 characters)
 Dump Class . . . OFFSITE             Dump Class . . .
 Dump Class . . .                     Dump Class . . .

 Allocation/migration Threshold: High . . 95  (1-99)      Low  . . 80  (0-99)
 Guaranteed Backup Frequency  . . . . . . 7         (1 to 9999 or NOLIMIT)

 ALTER SMS Storage Group Status . . ..... N   (Y or N)
```

Figure 42. Pool Storage Group Alter Panel

   We have the value for GUARANTEED BACKUP FREQUENCY set to 7. A value less than 7 may produce a large backup load on your system and is not recommended.

9. **What is the difference between GUARANTEED BACKUP FREQUENCY and BACKUP FREQUENCY?**

   The difference is that by setting the GUARANTEED BACKUP FREQUENCY attribute you ensure that a copy of the data set is taken after the specified number of days regardless of whether the data set has changed.

10. **How can I use the data set changed bit to back up data sets?**

    If you are installing DFSMShsm for the first time, you may have data sets that do not have the data set changed indicator in the format 1 DSCB turned on. DFSMShsm will use this field to decide whether a data set needs to be backed up. DFSMShsm has a parameter where you can ask that a copy of the data set is taken regardless of the setting of this bit. The parameter options are:

```
SETSYS INCREMENTALBACKUP(ORIGINAL)

SETSYS INCREMENTALBACKUP(CHANGEDONLY)
```

Specifying ORIGINAL means that a copy is taken for each data set that does not have the indicator turned on. Specifying CHANGEDONLY means that only data sets with the data set changed indicator set to on are backed up.

For the first few cycles of DFSMShsm automatic backup we recommend that you specify ORIGINAL. Over time you may want to alter it to CHANGEDONLY once you are more confident of the data that is being backed up.

11. **How do I specify the number of backup versions to keep for an existing data set?**

    The management class attribute NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) allows you to set the maximum number of backup versions to retain for a data set. In our management class we have set this to 2. This means that the two most recent copies of the data set will be kept.

12. **How do I specify the number of backup versions to keep for a data set that has been deleted?**

    The management class attribute NUMBER OF BACKUP VERSIONS (DATA SET DELETED) allows you to set the maximum number of backup versions to retain once a data set has been deleted. For example, if you set this to 1, only the most recent copy of the data set will be kept.

13. **How can I make sure that the only backup version of a data set that has been deleted from primary or migration volumes exists for a certain period of time?**

    By using the RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED), you can keep the most recent version. For example, if you set this to 30, the backup version would be kept for 30 days after the data set had been deleted before it is erased.

14. **How do I keep the extra backup versions that I have been collecting?**

    The RETAIN DAYS EXTRA BACKUP VERSIONS allows you to keep all but the most recent copy for a specified number of days. For example, if you specify 5, any extra versions will be kept for five days. The most recent copy is not affected by the setting of this parameter.

15. **How does DFSMShsm decide whether a data set has been deleted?**

    DFSMShsm makes this decision as part of EXPIREBV processing. If you have never run EXPIREBV processing before, you may want to double check your management class settings before running EXPIREBV for the first time.

16. **How can I control who has the ability to back up data sets by command?**

    By using the ADMIN OR USER COMMAND BACKUP, you can decide whether the storage administrator, or end user, or both or even no one can back up the data sets by command. The settings are fairly self-explanatory:

    - ADMIN - Only the storage administrator can perform command backups.
    - BOTH - End users as well as the storage administrator can perform command backups.

- NONE - Neither end users nor the storage administrator can perform command backups.

In our example we have both end users and storage administrators using command backup. However, this setting applies only to this particular management class.

17. **Within the management class I have specified AUTO BACKUP as Y. Does this mean that automatic backup will be performed?**

    No. This is a parameter that is affected by another within the management class. If you have specified ADMIN OR USER COMMAND BACKUP as NONE, then regardless of the AUTO BACKUP attribute setting, data sets associated with this management class are not backed up. Also, if you specify Y, then you are required to fill in the other backup attribute fields.

    The example that we have used allows automatic backup to be performed as we have specified BOTH and Y.

18. **If I want to back up data sets using concurrent copy, what do I have to do?**

    Concurrent copy applies only to data sets residing on level 0 volumes and will not process either ICF catalogs or PDSs. Use of concurrent copy for data set backup is justified only if the data set is database related and there is significant value in having the data set serialized for as short a duration as possible.

    However, if you have the necessary hardware to support concurrent copy, three options are available to you through the management class BACKUP COPY TECHNIQUE attribute.

    - R - CONCURRENT REQUIRED

      Concurrent copy must be used for backup. Backup will fail for data sets that do not reside on volumes supported by concurrent copy or are otherwise unavailable for concurrent copy.

    - P - CONCURRENT PREFERRED

      Concurrent copy should be used for backup. A data set is backed up without concurrent copy if it does not reside on a volume supported by concurrent copy or is otherwise unavailable for concurrent copy.

    - S - STANDARD

      Data sets are backed up without concurrent copy.

    If your data is allocated on an RVA, and you are using concurrent copy to back up your data sets, no changes are necessary to use virtual concurrent copy. Refer to 2.6.1, “Virtual Concurrent Copy” on page 76 for additional information about the function of concurrent copy and virtual concurrent copy. We also recommend that you read *Implementing Concurrent Copy* and *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*. Both of these redbooks provide an excellent background and implementation methodology for concurrent copy and virtual concurrent copy.

19. **Is that all I have to do to ensure that concurrent copy is used for my data sets?**

    No. The storage class parameter, ACCESSIBILITY, affects where your data sets are placed on concurrent copy capable devices. In this field you can specify whether allocation to a concurrent copy capable device is required (C), preferred (P), discouraged (S), or ignored (N) during volume selection.

Specifying a management class Backup Copy Technique as R (required) does not affect the placement of data sets on a concurrent copy volume.

Figure 43 shows how to set your storage class such that concurrent copy can be used.

```
                              STORAGE CLASS ALTER              Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Class Name  : SC54GRT
To ALTER Storage Class, Specify:
  Description ==> STORCLASS FOR GUARANTEED SPACE ALLOCATION
              ==>
  Performance Objectives
   Direct Millisecond Response . . . . 5          (1 to 999 or blank)
   Direct Bias . . . . . .            (R, W or blank)
   Sequential Millisecond Response . .            (1 to 999 or blank)
   Sequential Bias . . . . . .            (R, W or blank)
   Initial Access Response Seconds . . 5          (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .            (0 to 999 or blank)
  Availability . . . . . . . . . . . . S          (C, P ,S or N)
  Accessibility  . . . . . . . . . . . P          (C, P ,S or N)
   Backup  . . . . . .            (Y, N or Blank)
   Versioning  . . . . . .            (Y, N or Blank)
```

Figure 43. Storage Class Settings for Concurrent Copy

20. **Are there any settings that I need to make to my ARCCMDxx PARMLIB member for automatic backup?**

    Yes. If you want automatic backup to start and finish automatically, you have to set up a SETSYS parameter. For example, if you were to specify:

```
SETSYS AUTOBACKUPSTART(0100 0200 0300)
```

    you are asking DFSMShsm to:

    - Plan to start automatic backup at 0100.
    - Start automatic backup no later than 0200.
    - Start automatic backup no later than 0300 on any additional volumes.

    If DFSMShsm does not start automatic backup between 0100 and 0200 in the example we have given, automatic backup will not be done that day. After 0300, no additional DASD volumes will be started. Those already started will run to completion.

21. **Can I issue the same SETSYS command at the console?**

    Yes. There is no requirement that this SETSYS command be added to your ARCCMDxx PARMLIB member. However our recommendation is that you do add it.

22. **How can I ensure that all my eligible primary volumes are backed up before automatic backup completes?**

    Include only an automatic backup start window as follows:

```
SETSYS AUTOBACKUPSTART(0100 0200)
```

The omission of the third parameter (0300) tells DFSMShsm that all eligible primary volumes are to be backed up.

23. **How do I tell DFSMShsm the days on which I want automatic backup to run?**

    You tell DFSMShsm the days you want automatic backup to run through the DEFINE command:

```
DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(1997/01/06))
```

    If you want DFSMShsm to perform automatic backup, you must specify this in your ARCCMDxx PARMLIB member.

    This command establishes a seven-day backup cycle that starts on Monday, 6 January. Sunday is a day off for automatic backup. Specifying CYCLESTARTDATE means that the cycle will stay the same through each initialization of DFSMShsm. We recommend that you code your DEFINE similar to the example. You may find that it is easier to work with a seven-day cycle than a long string of Ys and Ns.

    If you are running automatic backup every day, then we recommend that you use a one-day backup cycle to reduce the number of partial tapes in use.

    Your string of alphabetic Ys and Ns can represent up to 31 days in the cycle.

24. **How do I specify the backup units to which I want the backup versions to go?**

    DFSMShsm can use either DASD or tape as target volumes. To direct your backup versions to tape and to a generic or esoteric that you have previously defined, you would code the following:

```
SETSYS BACKUP(TAPE(ETAPE))
```

    In this case all backups would go to tape and to the esoteric name of ETAPE, which you can set up to include the 3480 tape units that have automatic cartridge loaders.

    To direct your backups to DASD, you would code the following:

```
SETSYS BACKUP(DASD)
```

25. **How do I set up an esoteric name for my devices so that DFSMShsm can use them?**

    First you have to define the esoteric tape unit names to MVS during your system I/O generation. Then you can use the DFSMShsm SETSYS USERUNITTABLE to make them known to DFSMShsm.

    For example if you had defined ACL as all 3490 devices with automatic cartridge loaders, the command you would add to your ARCCMDxx member or issue from the console would be:

```
SETSYS USERUNITTABLE(ACL:3490)
```

Once you have done this, DFSMShsm will honor this esoteric name wherever you code it. Output will go to ACL drives, but input can use any 3490 drive.

26. **How do I limit the amount of data that is written to a cartridge-type device?**

    DFSMShsm recognizes that not all tapes have the same capacity. This is required if you use the TAPECOPY or DUPLEX capabilities for producing a second copy of an ML2 or backup tape. DFSMShsm provides you the flexibility to define the level of fullness that you want for your tapes. The following command allows you to specify the percentage of fullness that you believe is appropriate to your site:

```
SETSYS TAPEUTILIZATION(UNITTYPE(ACL) PERCENTFULL(97))
```

    This command limits those devices within the esoteric group of ACL to filling 97% of the tapes. You can issue this command for as many unit types and esoterics that you have at your site. One SETSYS TAPEUTILIZATION command is required for each unique unit name.

27. **How do I limit the data that is written to a backup tape when it is in a tape library?**

    Because esoteric unit names are ignored within a tape library, you would use the following command:

```
SETSYS TAPEUTILIZATION(LIBRARYBACKUP PERCENTFULL(97))
```

    This command specifies the desired level for a backup tape stored in a tape library.

28. **How do I make more than one copy of a backup tape at the same time?**

    By using the duplex tape capability of DFSMShsm, you can create two copies concurrently, The intention is for one to be kept onsite and the other to be stored remotely. With the following command you can make the copies concurrently:

```
SETSYS DUPLEX(BACKUP(Y))
```

    The DUPLEX tape function is an alternative to the TAPECOPY function. Over time the use of DUPLEX may make the TAPECOPY function all but forgotten in your environment.

29. **How can I RACF protect my backup versions by using a generic profile?**

    DFSMShsm generates its own name for the tape data set that contains the backup versions. It creates a name of the format:

```
bprefix.BACK.Tssmmhh.user1.user2.Xyddd
```

where bprefix is the UID that you specified in the DFSMShsm startup procedure or coded in the following command:

```
SETSYS BACKUPPREFIX(HSM)
```

You can use this prefix to create a RACF generic profile that will protect all of your backup copies. Protecting DFSMShsm resources and data sets using RACF is discussed in 2.3, “Providing Security for DFSMShsm Resources” on page 37.

30. **Can I also use the BACKUPPREFIX to prevent backup copies from being allocated to SMS-managed volumes?**

    Yes. You can update your ACS routines so that DFSMShsm backup copies are assigned to non-SMS-managed volumes and volumes that DFSMShsm has chosen.

31. **How can I increase the number of backup tasks that can run, and what should I consider before doing so?**

    DFSMShsm can have up to 15 volume backup tasks running concurrently in each processor. The following command is the one that affects the number of backup tasks:

```
SETSYS MAXBACKUPTASKS(3)
```

    In this case a maximum of three backup tasks can run concurrently.

    You have to consider the number of tape units you have available and the number of DASD volumes to back up. Each task will have its own target tape drive and its own source DASD to process. There is no point in specifying the maximum number of tasks (15) if you only have one tape drive or only 5 DASD volumes to process. If you use duplex tape, you will need two tape units for every backup task. You should also consider the system workload at the time that you run automatic backup.

32. **What is spill processing?**

    If you are backing up to DASD and the volume becomes full, DFSMShsm moves older backup versions of data sets to other volumes known as spill volumes. These volumes are usually tape. We recommend that you back up to tape, so spill processing will be available. To prevent spill process, use the following SETSYS command:

```
SETSYS NOSPILL
```

33. **What happens if automatic backup comes across a data set that is in use?**

    To guarantee the integrity of a data set DFSMShsm tries to serialize on system resources to indicate whether a data set is in use or not, recognizing the fact that a backup of a data set that may be changing is better than no backup at all. DFSMShsm provides two options, so you can make your own decision.

First there are those data sets that you do not expect to be in use but are open for update. This command allows you to retry the backup of a data set once more after a delay of 10 minutes, and if it is still in use, back up the data set up anyway:

```
SETSYS BACKUP(INUSE(RETRY(Y) DELAY(10) SERIALIZATION(PREFERRED)))
```

If you were to specify SERIALIZATION(REQUIRED) and the data set was still in use the second time around, DFSMShsm would fail the backup.

The second case applies when you know for sure that the data set is going to be open for update most of the time. Now you have a choice. You can either use the installationwide exit ARCBDEXT to back up the data set without serialization or fail the operation. The SETSYS INUSE subparameters can also apply in this situation, but the exit gives you the flexibility to override them for a particular data set.

34. **What happens during automatic backup?**

    Automatic backup is performed in four phases:

    a. Backing up the CDSs

    b. Moving backup versions

    c. Backing up migrated data sets

    d. Backing up DFSMShsm-managed volumes

    However, if you are in a multiprocessor environment, only the primary processor performs the first three phases.

35. **Does this mean that I should take any special steps to ensure that I do not impact other functions within DFSMShsm?**

    In a multiprocessor environment, always start automatic backup on the primary host processor about 10 to 30 minutes before you start it on any others, depending on the expected time to accomplish the first three phases of automatic backup. This allows the CDSs to be backed up, and the other processors do not have to wait for the CDSs.

    Another tip is to run automatic backup before space management so that backup versions created with the BACKDS command are moved off the ML1 volumes where they temporarily reside.

36. **Is there a way in which I can back up data sets in a batch environment?**

    Yes. DFSMShsm supplies a program to allow you to perform what is known as an inline backup. The program that you execute is called ARCINBAK. It allows you to back up data sets in the middle of a job by the addition of a new step to the job. In the example below note that the DD names to identify the data sets that you want backed up are identified as BACKnn. The data set names associated with these DDs will not be backed up. A DD name prefix of anything other than BACK is not allowed. The JCL in Figure 44 on page 96 is an example of how to use ARCINBAK:

```
//JOBNAME  JOB . . . ,USER=USERID,PASSWORD=USERPSWD
//STEP1    EXEC PGM=USERPGM
//SYSPRINT DD SYSOUT=A
//DSET1    DD DSN=USERID.NO3.GDG(-1),DISP=OLD
//DSET2    DD DSN=USERID.NO3.PSFB,DISP=OLD
//DSET3    DD DSN=USERID.NO4.PSFB,DISP=OLD
//DSET4    DD DSN=USERID.NO3.KSDS,DISP=OLD
/*
//STEP2    EXEC PGM=ARCINBAK
//ARCPRINT DD  SYSOUT=A
//ARCSNAP  DD  SYSOUT=A
//* ------------------------------------------------------------
//*        BACKUP OF GDG DATA SET SHOULD BE SUCCESSFUL.
//* ------------------------------------------------------------
//BACKO1   DD  DSN=*.STEP1.DSET1,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//BACKO2   DD  DSN=*.STEP1.DSET2,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ------------------------------------------------------------
//BACKO3   DD  DSN=*.STEP1.DSET4,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF GDG DATA SET SHOULD BE SUCCESSFUL.
//* ------------------------------------------------------------
//BACKO4   DD  DSN=USERID.NO1.GDG.GOOO1VOO,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ------------------------------------------------------------
//BACKO5   DD  DSN=USERID.NO1.PSFB,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF UNCATALOGED DATA SET SHOULD FAIL.
//* ------------------------------------------------------------
//BACKO6   DD  DSN=USERID.NO2.UNCAT,VOL=SER=VOLOO3,UNIT=3390,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ------------------------------------------------------------
//* ------------------------------------------------------------
//BACKO7   DD  DSN=USERID.NO1.KSDS,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF OPEN IN-USE VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ------------------------------------------------------------
//BACKO8   DD  DSN=USERID.NO2.KSDS,DISP=SHR
//* ------------------------------------------------------------
//*        BACKUP OF RACF PROTECTED NON-VSAM DATA SET
//*        BY AN UNAUTHORIZED USER SHOULD FAIL.
//* ------------------------------------------------------------
//BACKO9   DD  DSN=USERXX.NO2.PSFB,DISP=SHR
/*
```

Figure 44. Sample ARCINBAK JCL

37. **Can I use the TSO/E terminal monitor program (TMP) in a batch environment to back up data sets?**

    Yes. Program IKJEFT01 sets up a TSO/E environment from which you can issue TSO/E commands. You can create a batch job that runs the IKJEFT01 program and issues the HBACKDS command. To back up a data set using

the HBACKDS command you must have RACF UPDATE authority to the data set. You can specify a data set name filter as a parameter of the HBACKDS command. You can use the CHANGEDONLY parameter to indicate whether DFSMShsm should back up only those specified data sets that have the changed bits on in their data set VTOC entries. The JCL in Figure 45 gives an example of using the HBACKDS command in batch:

```
//BACKUP JOB (999,POK),'HBACKDS',CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//TMP      EXEC PGM=IKJEFT01
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPROC  DD  DISP=SHR,DSN=SYS1.SHARED.CLIST
//SYSTSIN    DD *
  HBACKDS ('MARYL.**') CHANGEDONLY
/*
```

Figure 45. Sample JCL to Issue HBACKDS Command in Batch

38. **Can I use aggregate backup and recovery support (ABARS) with TSO/E terminal monitor program (TMP) in a batch environment to back up data sets?**

    Yes. The ABARS component of DFSMShsm provides for data backup on a predefined set of data, known as an aggregate with the ABACKUP command. The ABACKUP command can be issued in the batch environment using the IKJEFT01 program. RACF Facility class support allows the security administrator to authorize application owners and system programs to issue the ABACKUP command for specific aggregates. For more information about ABARS refer to 2.11, “Aggregate Backup and Recovery Support” on page 114.

2.8.2 Availability Management Settings Summary

So that you do not have to wade through all the different pages associated with this topic, we show without any explanation the storage group, storage class, management class, and DFSMShsm ARCCMDxx PARMLIB member changes that we implemented on our system. Use this summary as a quick reference and memory aid.

Figure 46 on page 98 shows the Management Class Alter panel.

```
                          MANAGEMENT CLASS ALTER            Page 3 of 5
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Management Class Name : MC54PRIM

To ALTER Management Class, Specify:
  Backup Attributes
    Backup Frequency  . . . . . 1        (0 to 9999 or blank)
    Number of Backup Vers . . . . . 2    (1 to 100 or blank)
      (Data Set Exists)
    Number of Backup Vers . . . . . 1    (0 to 100 or blank)
      (Data Set Deleted)
    Retain days only Backup Ver . . . 30 (1 to 9999, NOLIMIT or blank)
      (Data Set Deleted)
    Retain days extra Backup Vers . . 5  (1 to 9999, NOLIMIT or blank)
    Admin or User command Backup  . . BOTH (BOTH, ADMIN or NONE)
    Auto Backup . . . . . Y      (Y or N)
    Backup Copy Technique . . . . . S    (P=Conc Preferred, R=Conc
                                         Required or S=Standard)
```

Figure 46. Management Class Backup Attributes

Figure 47 shows the Pool Storage Group Alter panel.

```
                          POOL STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Group Name  : SG54HSM
To ALTER Storage Group, Specify:
 Description ==> STORAGE GROUP FOR DFSMSHSM DATA SETS.
             ==>
 Auto Migrate . . Y  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . Y  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . Y  (Y or N)         Dump Sys/Sys Group Name  . . .

 Dump Class . . . ONSITE              (1 to 8 characters)
 Dump Class . . . OFFSITE             Dump Class . . .
 Dump Class . . .                     Dump Class . . .

 Allocation/migration Threshold: High . . 95  (1-99)      Low  . . 80  (0-99)
 Guaranteed Backup Frequency  . . . . . 7         (1 to 9999 or NOLIMIT)

 ALTER SMS Storage Group Status . . ..... N   (Y or N)
```

Figure 47. Storage Group Attributes

Figure 48 on page 99 shows the Storage Class Alter panel.

```
                              STORAGE CLASS ALTER              Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Storage Class Name  : SC54GRT
To ALTER Storage Class, Specify:
  Description ==> STORCLASS FOR GUARANTEED SPACE ALLOCATION
              ==>
  Performance Objectives
   Direct Millisecond Response . . . . 5          (1 to 999 or blank)
   Direct Bias . . . . . .            (R, W or blank)
   Sequential Millisecond Response . .            (1 to 999 or blank)
   Sequential Bias . . . . . .        (R, W or blank)
   Initial Access Response Seconds . . 5          (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .            (0 to 999 or blank)
  Availability . . . . . . . . . . . . S          (C, P ,S or N)
  Accessibility  . . . . . . . . . . . P          (C, P ,S or N)
   Backup  . . . . . .                (Y, N or Blank)
   Versioning  . . . . . .            (Y, N or Blank)
```

Figure 48. Storage Class Alter Panel

Figure 49 on page 100 shows the DFSMShsm ARCCMDxx PARMLIB settings that relate to automatic backup and which we will use in this book:

```
DEFINE BACKUP(YYYYYYN       /* DIRECT DFSMSHSM TO BACKUP ELIGIBLE*/ -
  CYCLESTARTDATE(1997/01/06))   /* DATA SETS DAILY TO TWO VOLUMES*/
                            /* (DUPLEX IS ON FOR BACKUP) STARTING*/
                            /* MONDAY 6TH, JANUARY 1997          */
                            /* NO BACKUPS ON SUNDAYS             */

SETSYS                      /* ACTIVATE THE BACKUP, DUMP AND     */ -
  BACKUP(TAPE(3490))        /* RECOVERY FUNCTIONS OF DFSMSHSM    */

SETSYS                                                            -
  AUTOBACKUPSTART(0100 0200 0300)
                            /* AB START AT 0100 OR NO LATER THAN */
                            /* 0200, NO AB AFTER 0300            */

SETSYS                      /* HSM IS HLQ FOR BACKED UP DATA SETS*/ -
  BACKUPPREFIX(HSM)

SETSYS                      /* MAXIMUM 3 BACKUP TASKS            */ -
  MAXBACKUPTASKS(3)         /* BACKUP ALL VOLUMES WITH AB        */ -
  NOSKIPABPRIMARY           /* ATTRIBUTE                         */

SETSYS                      /* AB TO DASD NOT USED. SPILL TURNED */ -
 NOSPILL                    /* OFF                               */

SETSYS                      /* MAKE INITIAL BACKUP COPIES OF DATA*/ -
  INCREMENTALBACKUP(ORIGINAL) /* SETS DESPITE THE SETTING OF THE */
                            /* CHANGE BIT (DS1DSCHA)             */

SETSYS                      /* ISSUE ARC0310A AFTER 10 MINUTES IF*/ -
  MOUNTWAITTIME(10)         /* NO MOUNT                          */

SETSYS                      /* USE 3490 FOR BACKUP               */ -
  UNITNAME(3490)

SETSYS                      /* ISSUE ARC0313A, ARC0314A, ARC0366A*/ -
  TAPEINPUTPROMPT(BACKUPTAPES(YES))

SETSYS                      /* MAKE TWO BACKUP COPIES            */ -
  DUPLEX(BACKUP(Y) MIGRATION(N))

SETSYS                      /* UTILIZE 97% OF TAPE CARTRIDGE     */ -
  TAPEUTILIZATION(                                                -
   UNITTYPE(3490) PERCENTFUL(97))
```

Figure 49 (Part 1 of 2). Sample ARCCMD54 PARMLIB Member

```
SETSYS                      /* DO NOT SPAN TAPES FOR DATA SETS   */ -
  TAPESPANSIZE(100)         /* 100 MBS OR LESS IN SIZE           */ -

SETSYS                      /* RECYCLED TAPES RETURN TO THE      */ -
  TAPEDELETION(             /* GLOBAL SCRATCH POOL               */ -
   BACKUP(SCRATCHTAPE)                                              -
   MIGRATION(SCRATCHTAPE)                                           -
   DUMP(SCRATCHTAPE))

SETSYS                      /* AT EOV SELECT PRIVATE SCRATCH     */ -
  SELECTVOLUME(             /* VOLUME                            */ -
  BACKUP(SCRATCH)                                                   -
  MIGRATION(SCRATCH)                                                -
  DUMP(SCRATCH))

SETSYS                      /* IDRC USED WHEN 3480X              */ -
  TAPEHARDWARECOMPACT

SETSYS                      /* DO NOT MARK TAPES FULL EARLY BUT  */ -
  PARTIALTAPE(              /* EXTEND THEM AT NEXT OPPORTUNITY   */ -
   BACKUP(REUSE)                                                    -
   MIGRATION(REUSE))
```

Figure 49 (Part 2 of 2). Sample ARCCMD54 PARMLIB Member

2.8.3 Getting Started with Dump Availability Management

The process of copying all data from a DASD volume to a tape volume is called volume dump. The dump of the entire volume is performed by DFSMShsm invoking DFSMSdss through an application interface. The entire allocated space of DFSMShsm-managed DASD volumes and ML1 volumes is dumped. Volumes not in storage groups or not ADDVOLed to DFSMShsm are dumped only by command.

In this section we show you how to:

- Set up SMS so that automatic dump will be performed
- Specify the volumes that will be dumped
- Make more than one dump copy at a time
- Specify dump class attributes, which include stacking multiple dump generations on a single Magstar cartridge.
- Setup your dump cycle.
- Use concurrent copy to dump volumes

1. **How do I specify the volumes to dump?**

   DFSMShsm needs to be told which volumes to dump and to which dump class, through the storage group specifications. Figure 50 on page 102 shows the Pool Storage Group Alter panel that allows you to specify whether you want automatic dump to be performed. Specifying Y tells DFSMShsm to perform automatic dump for all volumes in the storage group.

```
                          POOL STORAGE GROUP ALTER
 Command ===>

 SCDS Name . . . . . : SYS1.SC54.SCDS1
 Storage Group Name  : SG54HSM
 To ALTER Storage Group, Specify:
  Description ==> STORAGE GROUP FOR DFSMSHSM DATA SETS.
              ==>
  Auto Migrate . . Y  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
  Auto Backup  . . Y  (Y or N)         Backup Sys/Sys Group Name  . .
  Auto Dump  . . . Y  (Y or N)         Dump Sys/Sys Group Name  . . .

  Dump Class . . . ONSITE              (1 to 8 characters)
  Dump Class . . . OFFSITE             Dump Class . . .
  Dump Class . . .                     Dump Class . . .

  Allocation/migration Threshold: High . . 95  (1-99)      Low  . . 80  (0-99)
  Guaranteed Backup Frequency  . . . . . . 7         (1 to 9999 or NOLIMIT)

  ALTER SMS Storage Group Status . . ..... N   (Y or N)
```

Figure 50. Pool Storage Group Alter Panel

2. **What is a dump class and how is it used?**

   A dump class is a predefined DFSMShsm-named set of characteristics that describe how volume dumps are managed. With the Pool Storage Group Alter panel you can define or alter the dump classes to which the output is directed.

   The definition of the attributes is done outside ISMF, using DFSMShsm commands.

   The example shown in Figure 50 indicates that we want to direct our output to two dump classes: ONSITE and OFFSITE. You can specify up to five dump classes.

3. **How many copies of my dumped volume can I make concurrently?**

   The maximum number of dump classes that you can specify is five; it is the maximum that you can dump to concurrently with automatic dump.

4. **There must be potential for a lot of dump copies to be created. How does DFSMShsm refer to them?**

   DFSMShsm refers to each successive full-volume dump of a volume as a generation. Each dump copy in a generation is associated with a different dump class.

5. **Can automatic dump exploit the increased capacity of Magstar cartridges?**

   Yes. This is achieved by use of the STACK option when you define your dump class. With this option you can stack a specified number of DFSMShsm invoked DFSMSdss physical volume dumps to a single cartridge (or stream of cartridges) as separate files during automatic dump processing.

6. **How much can a Magstar cartridge hold in terms of dumped volumes?**

   A single Magstar cartridge can hold dumps from about 14 different 3390-3 DASD volumes.

7. **Can DFSMSdss Stand Alone services restore dumps that are stacked on a Magstar cartridge?**

Yes. With DFSMS/MVS Version 1 Release 4, the Stand-Alone services of DFSMSdss enable you to specify a FILE parameter on the RESTORE command. The FILE parameter specifies the relative position, from the beginning of the tape volume, where the dump data set begins. For more information about DFSMSdss Stand-Alone services, refer to the *DFSMSdss Storage Administration Reference*, SC26-4929 at the DFSMS/MVS Version 1 Release 4 level or later.

8. **What if DFSMShsm encounters an error while writing to one of the dump copies?**

   Should this happen, DFSMSdss discontinues writing to dump copies on which errors occur. DFSMSdss creates the multiple copies and continues its full-volume dump process as long as one output copy is good. DFSMShsm then discards the contents of only the bad copies and issues an appropriate message to indicate the type of failure.

   If all copies fail, the full-volume dump is failed.

9. **How do I specify the starting and ending times for automatic dump?**

   You have to update your ARCCMDxx PARMLIB member with a statement similar to this:

```
SETSYS AUTODUMPSTART(0400 0500 0600)
```

   In this example automatic dump will start between 4 a.m. and 5 a.m. and will not process any volumes after 6 a.m.

10. **How do I make sure that DFSMShsm dumps all volumes?**

    If you were to code:

```
SETSYS AUTODUMPSTART(0400 0500)
```

    DFSMShsm would not stop automatic dump processing until all volumes had been dumped. The key is to omit the third parameter and just specify your automatic dump start window.

11. **Can I adjust the number of tracks that are read for each EXCP in the same way that I can for DFSMSdss?**

    If you are familiar with DFSMSdss, you know that you can choose the DASD I/O buffering technique to use, using the following command:

```
SETSYS DUMPIO(n,m)
```

    where n indicates the DFSMSdss buffering technique for physical dump (BACKVOL with DUMP parameter or during automatic dump), and m indicates the value used for DFSMSdss logical dump when DFSMShsm uses DFSMSdss to move a data set.

    The values for n and m are:

    - 1 - DFSMSdss reads one track at a time.
    - 2 - DFSMSdss reads two tracks at a time.

- 3 - DFSMSdss reads three tracks at a time.
- 4 - DFSMSdss reads one cylinder at a time.

The settings that you specify for this parameter will affect your real and virtual storage requirements. Before you decide to specify anything other than the default for n, which is 1, or the default for m, which is 4, we recommend that you read the *DFSMShsm Implementation and Customization Guide*.

12. **How do I specify that I want concurrent copy to be used for volume dumps?**

    If you have concurrent copy capability and want it used for volume dumps, specify this command which causes DFSMSdss to use concurrent copy when performing full-volume dumps:

```
SETSYS VOLUMEDUMP(CC)
```

13. **What is the maximum number of dump tasks that I can run concurrently and how do I specify it?**

    DFSMShsm can run up to 15 concurrent dump tasks in one processor. This does not limit the number of dump copies as up to five dump copies can be made under each dump task. This sample command to add to your ARCCMDxx member allows up to three dump tasks to run concurrently:

```
SETSYS MAXDUMPTASKS(3)
```

14. **What should I take into consideration when specifying the number of dump tasks?**

    You have to consider the amount of tape units that you expect to have available during automatic dump processing. To determine the amount of tape drives needed, multiply the number of dump classes for each volume by the number of dump tasks. If this number exceeds the number of tape drives that you expect to have available, you have to lower your MAXDUMPTASKS parameter. If this number is below the number of tape drives that you expect to have available, you can raise your MAXDUMPTASKS parameter.

15. **How can I specify the days on which I would like automatic dump to occur?**

    Use the DFSMShsm DEFINE DUMPCYCLE command:

```
DEFINE DUMPCYCLE (NNNNNNY CYCLESTARTDATE(1997/01/06))
```

    This command establishes a seven-day dump cycle that starts on Monday, 6 January. Automatic dump will run only on Sunday. Specifying CYCLESTARTDATE means that the cycle will stay the same through each initialization of DFSMShsm. We recommend that you code your DEFINE similar to the example. You may find that it is easier to work with a seven-day cycle than a long string of Ys and Ns. This command can be added to your ARCCMDxx PARMLIB member.

16. **Is that all I need to do to let automatic dump take place?**

No. You must now define at least one dump class so that automatic dump has somewhere to direct the copies.

17. **How do I define a dump class and what are the attributes I can associate with it?**

    You create dump classes by using the DFSMShsm DEFINE DUMPCLASS command. As there are a lot of considerations that your installation may want to take into account, we briefly explain what you can achieve on dump class definition, the parameters that you can specify, and the DFSMShsm default (highlighted) on the DEFINE DUMPCLASS command. Later we will show you a sample definition that we will use throughout this book.

    For our examples we define a dump class called ONSITE. With the DEFINE DUMPCLASS command you can specify:

    - The name of the dump class:

```
DEFINE DUMPCLASS(ONSITE)
```

    - Whether dump volumes are available for reuse when invalidated:

```
DEFINE DUMPCLASS(ONSITE AUTOREUSE | NOAUTOREUSE)
```

    - Whether data set restore is allowed from a full-volume dump copy:

```
DEFINE DUMPCLASS(ONSITE DATASETRESTORE | NODATASETRESTORE)
```

    - That a dump be taken only on a particular day in the dump cycle:

```
DEFINE DUMPCLASS(ONSITE DAY(day))
```

      Note: If DAY is not specified, and the FREQUENCY parameter is met, and it is a Y day in the dump cycle, the volume will be dumped.

    - That a particular dump class be disabled for output only:

```
DEFINE DUMPCLASS(ONSITE DISABLE)
```

      Note: Issuing the DEFINE command without specifying DISABLE will enable a dump class.

    - The disposition of dump volumes:

```
DEFINE DUMPCLASS(ONSITE DISPOSITION('SUNDAY DUMP IS GOOD'))
```

      Note: This will insert the disposition statement into message ARC0637I, which is issued for each successful dump copy.

    - The minimum number of days between volume dumps to a class:

```
DEFINE DUMPCLASS(ONSITE FREQUENCY(days | 7))
```

- Whether the data set changed indicator is to be reset by DFSMSdss for each data set following a full-volume dump:

```
DEFINE DUMPCLASS(ONSITE RESET | NORESET))
```

Note: Do not use RESET if you want incremental backup to make a backup version. RESET is not used if the DASD volume is also being managed by DFSMShsm backup.

- When the data on the dump volumes becomes invalid:

```
DEFINE DUMPCLASS(ONSITE RETENTIONPERIOD(days | NOLIMIT))
```

- How many dump copies DFSMShsm should place on a dump volume assigned during one invocation of automatic dump:

```
DEFINE DUMPCLASS(ONSITE STACK(nn))
```

Note: Where nn is a value from 1 to 99.

- The expiration date to use in the tape header label:

```
DEFINE DUMPCLASS(ONSITE TAPEEXPIRATIONDATE(yyyyddd | yyddd))
```

Note: This setting is effective only if you have specified a SETSYS command of EXPIRATION or EXPIRATIONINCLUDE. The setting of this may be meaningful to certain tape management systems, and some values may mean never expire.

- The type of tape unit to use for the dump tapes:

```
DEFINE DUMPCLASS(ONSITE UNIT(unittype))
```

Note: You can use a generic name or an esoteric name defined with the SETSYS USERUNITTABLE command in place of unittype.

- The number of generations for which copies of the VTOC of dumped volumes will be kept:

```
DEFINE DUMPCLASS(ONSITE VTOCCOPIES(copies | 2))
```

18. **Can you show an example of the DEFINE DUMPCLASS command?**

The examples that we show here are those that we use in this book. We have two dump classes defined: ONSITE and OFFSITE.

The definitions for ONSITE are:

```
DEFINE DUMPCLASS(ONSITE UNIT(3490) -
  RETPD(15) AUTOREUSE RESET  -
  DATASETRESTORE VTOCCOPIES(4) -
  DAY(7) -
  DISPOSITION('ONSITE DUMP SUNDAY'))
```

Note: In our ONSITE dump class we have specified both AUTOREUSE and RESET. If SETSYS TAPEDELETION(SCRATCHTAPE) is in effect, AUTOREUSE has no meaning. Our DASD is not managed by incremental backup, so we use RESET. Do not specify RESET if you want incremental backup to make a backup version.

The definitions for OFFSITE are in Figure 51

```
DEFINE DUMPCLASS(OFFSITE UNIT(3490) -
  FREQUENCY(7) RETPD(356) -
  NOAUTOREUSE -
  NODATASETRESTORE NORESET -
  DISPOSITION('OFFSITE DUMP') -
  VTOCCOPIES(0))
```

Figure 51. OFFSITE Dump Class Definitions

2.9 Command Availability Management

We have shown how you can back up or dump your data automatically. However, there may come a time when you want to back up data sets or dump volumes outside the automatic start times.

In this section we show you how to:

- Dump a volume to a specified dump class
- Back up an individual data set
- Back up eligible changed data sets
- Back up all supported data sets on a volume
- Back up cataloged data sets using a filter

We do not show you all possible variations of the commands. Those that we show perform the most basic of functions available for each command. As your familiarity with DFSMShsm grows, you may want to investigate variations of the commands that we show here.

1. **How do I dump an entire volume with one single command?**

   An entire volume can be dumped by using the following command (in this case the volser of the volume is HSM14C):

```
BACKVOL VOLUME(HSM14C) DUMP
```

2. **Is there any way that I can specify the dump class to which I would like the dump to go?**

   By issuing the following command for the same volume you would be able to target the dump to up to five dump classes:

```
BACKVOL VOLUME(HSM14C) -
DUMP(DUMPCLASS(CHELSEA,ARSENAL,QPR,WESTHAM,INTER))
```

   These dump classes must have already been defined by means of the DFSMShsm DEFINE command.

3. **How do I back up individual data sets?**

   You can back up data sets by using either of the following commands (in this case we back up data set VIALLI.STAR):

```
BACKDS VIALLI.STAR
```

   or you could use:

```
HBACKDS VIALLI.STAR
```

   To issue the BACKDS command you need to be authorized to do so. The end-user HBACKDS command does not need authorization, and it allows filtering to be used.

4. **Is there a parameter in the management class that needs to be set to allow the backup to be taken?**

   Yes. If you remember we said that DFSMShsm will check the management class of the data set to determine the value of the ADMIN OR USER COMMAND BACKUP ATTRIBUTE. If the value is set to:

   - ADMIN - Only the storage administrator can perform command backups, using either command, but end users cannot.
   - BOTH - Storage administrators can perform command backups, using either command, and end users can use the HBACKDS command.
   - NONE - Command backup cannot be done.

5. **Can I back up all individual data sets on a volume?**

   By using the following command, all data sets on volume HSM41C would be backed up (if they are eligible and backup is permitted):

```
BACKVOL VOLUME(HSM14C) TOTAL
```

6. **What do you mean by eligible data sets for incremental backup?**

   DFSMShsm goes through all entries for data sets on a volume and those that it deems eligible have:

- The management class ADMIN OR USER COMMAND BACKUP attribute of ADMIN or BOTH
- The management class AUTO BACKUP attribute value of Y
- Eligibility, as determined by the data-set-changed indicator, as follows:
  - If the data-set-changed indicator is off and SETSYS INCREMENTALBACKUP(ORIGINAL) is specified, DFSMShsm backs up the data set if a previous backup version does not exist.
  - If the data-set-changed indicator is on and DFSMShsm can determine a last backup date, DFSMShsm backs up the data set if the number of days since the last backup date equals or exceeds the value specified with the management class BACKUP FREQUENCY attribute.
  - If the data-set-changed indicator is on but DFSMShsm cannot determine a last backup date (indicating a new data set), DFSMShsm backs up the data set.

7. **How do I back up only eligible data sets that have changed?**

   To back up only the changed data sets that exist on volume HSM14C, you can use either of the following commands:

```
BACKVOL VOLUME(HSM14C)

BACKVOL VOLUME(HSM14C) INCREMENTAL
```

8. **As a TSO user, is it possible to back up selected data sets by using filtering similar to ISPF rules?**

   Yes. DFSMShsm supports the following wildcard symbols:

   - % - one and only one character in a qualifier
   - %% - up to eight percent signs can be specified in each qualifier
   - * - one or more characters in one qualifier
   - ** - zero or more qualifiers (the double asterisk cannot precede or follow any characters; it must be preceded and followed by either a period or a blank)

   In addition to filtering, you could also specify whether DFSMShsm should back up only those nonmigrated data sets that have never been backed up or have been changed since being backed up. This is accomplished by adding the CHANGEDONLY parameter to the HBACKDS command.

   If you wanted to back up all cataloged data sets for a TSO user with a high-level qualifier of MHLRES5, you would issue this command:

```
HBACKDS 'MHLRES5.**'
```

   Variations of these commands exist and are more complicated, but the above examples should give you an idea of the basics that you have to work with.

2.10 Recovery and Restore

The recovery and restore process, which could be considered the opposite of backup and dump, is not an automatic process. DFSMShsm will not automatically recover or restore data if it becomes damaged. The recover and restore process is driven by commands.

In this topic we show you how to:

- Recover the most recent version of a data set
- Restore a data set from a dump copy
- Restore a volume from a dump copy and update it from incremental backup versions
- Restore a volume from a full-volume dump copy
- Recover a volume from DFSMShsm backup versions

1. **Is there anything I need to know before I attempt to perform either data set or volume recovery or restore?**

   The recovery process has two restrictions:

   - DFSMShsm cannot recover a data set to a migration volume.
   - DFSMShsm cannot recover a data set that is currently migrated unless it is non-VSAM and NEWNAME is specified.

2. **Are there any SETSYS parameters that will facilitate data set recovery?**

   You must have the following coded if you want to use recovery or restore:

```
SETSYS BACKUP
```

   You also want to consider the number of data set recovery tasks that you want to run concurrently. The number of tasks is affected by the following parameter:

```
SETSYS MAXDSRECOVERTASKS(nn)
```

   where nn can be any number between 1 and 64.

3. **Should I set MAXDSRECOVERTASKS to 64 to get the maximum performance from DFSMShsm within recovery?**

   No. To decide how many tasks DFSMShsm should run, you must consider how many tape units are available. Additionally the number of tasks used in recovering a given volume is limited by the number of backup tape volumes required. Once all the data set recovery tasks for one volume have been scheduled, you can then request recovery for another volume if needed.

4. **How do I recover the most recent copy of a data set from either a dump or backup copy?**

   If no data set of the same name exists in the system catalog, you can use either of the following commands:

```
HRECOVER dsname

RECOVER dsname
```

where dsname is the name of the data set that you want to recover. DFSMShsm chooses the most recently created version of the data set for you. You do not have to identify where the most recent copy is yourself before you put the command together.

5. **What is the difference between HRECOVER and RECOVER?**

   HRECOVER can be issued by any user; however, the RECOVER command can be issued only by DFSMShsm-authorized users.

6. **How do I recover a data set if a catalog entry already exists for it?**

   The following two forms of the command recover a specific data set but give it a new name:

```
HRECOVER dsname NEWNAME(newdsname)

RECOVER dsname NEWNAME(newdsname)
```

   where dsname is the name of the data set that you want to recover and newdsname is the new name that you want it to be called.

7. **Is there any way that I can just replace the data set if it exists?**

   By using either of the following commands you can replace an existing, cataloged data set with the recovered version:

```
HRECOVER dsname REPLACE

RECOVER dsname REPLACE
```

8. **How do I restore a data set from a DFSMShsm dump copy?**

   If you are a DFSMShsm-authorized user, you can request that a data set be restored from a dump copy, rather than from a backup version. The following command restores the latest dump copy from available dump volumes. If the data set already exists, you would have to add the REPLACE optional parameter for the data set to be successfully restored.

```
RECOVER dsname FROMDUMP
```

9. **What happens if an out-of-space condition occurs when I am recovering an SMS-managed data set?**

   If this does happen and the messages that you get indicate that there is not enough space, retry the RECOVER command but this time add the DFDSSOPTION keyword:

```
RECOVER dsname DFDSSOPTION(VOLCOUNT(ANY))
```

This will allow the data set to extend to as many volumes as required, provided that space is available within the same storage group.

10. **How do I combine the restore of an entire volume with full-volume recovery?**

    One of the first things to consider for a volume restore with update is whether it is likely that one of your users may try to start using a data set in the period between the restore and the incremental recovery. In this situation, we suggest that you put the SMS-managed volume in DISALL status.

    Once you are satisfied that the volume is in DISALL status, there are three variations of the command that you could use.

    In all these examples we assume that:

    - The DASD volume to be recovered has a volser of OLD001.
    - The DASD unit type is a 3390.
    - The volume restore will be done to a spare DASD volume of NEW001, which will change to OLD001 during the restore operation.
    - You know either the dump class, dump volume, or dump generation from which you want to restore.
    - You want an incremental volume recovery process to follow immediately the restore as part of the same request.

    To recover from a specific dump volume with a volser of DMP001, you would use the following command:

```
RECOVER * TOVOLUME(OLD001) UNIT(3390) -
 FROMDUMP(DUMPVOLUME(DMP001) APPLYINCREMENTAL) -
 TARGETVOLUME(NEW001)
```

    To recover from a specific dump class of ONSITE, you would use the following command:

```
RECOVER * TOVOLUME(OLD001) UNIT(3390) -
 FROMDUMP(DUMPCLASS(ONSITE) APPLYINCREMENTAL) -
 TARGETVOLUME(NEW001)
```

    To recover from a specific dump generation that was the most recent, you would use the following command:

```
RECOVER * TOVOLUME(OLD001) UNIT(3390) -
 FROMDUMP(DUMPGENERATION(0) APPLYINCREMENTAL) -
 TARGETVOLUME(NEW001)
```

11. **Once I have done this, the volume is then completely restored?**

Not necessarily. You must still check all the messages related to this restore and recovery process. For instance, if the volume being recovered contains part of a multivolume data set, that partial data set is not recovered. The data set will be listed as not recovered, and you have the opportunity to recover the data set by other means.

12. **How do I just restore from a full-volume dump?**

    DFSMShsm invokes DFSMSdss to perform a full-volume restore if the following command is used:

```
RECOVER * TOVOLUME(OLD001) UNIT(3390) -
 FROMDUMP TARGETVOLUME(NEW001)
```

    DFSMShsm restores from the most recent dump copy. It searches its entire dump inventory to determine the most recent dump copy.

13. **Is there a way that I could limit the selection of dump copies searched?**

    By adding to the previous command this parameter:

```
DATE(98/07/27)
```

    You would be asking DFSMShsm to recover the latest dump copy that was made on or before July 27,1998. If you knew that the DASD volume was good on a certain date, you might use this parameter to choose a dump taken on or before that date. This may cut down the time it takes to find the most recent copy if you know approximately the date when a dump copy may have been taken.

14. **When I am restoring a volume, can I also specify either the dump class, dump volume, or dump generation?**

    Yes. The same parameters are valid whether or not you are subsequently performing volume recovery.

15. **Is it possible to recover a volume from just incremental backup versions?**

    Volume recovery or incremental volume recovery recovers a volume at the data set level to the level of its latest backup. Therefore each supported data set on the volume is at the most recent level unless it has changed since the last time it was backed up.

    The following command causes volume recovery from incremental backup versions that were backed up on or after July 27,1998:

```
RECOVER * TOVOLUME(OLD001) UNIT(3390) DATE(98/07/27)
```

16. **What drives the incremental recovery process?**

    During this phase of DFSMShsm processing, the latest level of the VTOC copy data set is used.

We have shown some examples of the commands that you can use to complement your backup and dump processes. There are many variations that you can apply at your installation and that are relevant to your environment.

2.11 Aggregate Backup and Recovery Support

ABARS is a component of DFSMShsm that performs data backup and recovery processes on a predefined set of data known as an aggregate.

In this section we show you how to:

- Define an aggregate group
- Define a selection data set
- Define an instruction data set
- Set your ABARS-related SETSYS commands
- Perform an aggregate backup (ABACKUP)
- Perform an aggregate recover (ARECOVER)

1. **What is an aggregate group and how do I define one?**

   An aggregate group identifies through ISMF your selection data set name, instruction data set name, and the management control information required to perform ABACKUP.

   Figure 52 shows the ISMF Aggregate Group Application Selection panel that is displayed after selecting option 9 from the ISMF main entry panel.

   Figure 52

```
                    AGGREGATE GROUP APPLICATION SELECTION
Command ===>

To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . . 'SYS1.SC54.SCDS1'
                          (1 to 44 Character Data Set Name or 'Active')

  Aggregate Group Name . . PAY1      (for Aggregate Group List, fully or
                                     Partially Specified or * for All)
Select one of the following Options:
  3  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria
```

Figure 52. ISMF Aggregate Group Application Selection

   We have chosen to call our aggregate group PAY1.

   Once you have selected a name for your aggregate group and pressed Enter, you are presented with the panels shown in Figure 53 on page 115 and Figure 54 on page 115.

```
                          AGGREGATE GROUP DEFINE
Command ===>

SCDS Name  . . . . . : SYS1.SC54.SCDS1
Aggregate Group Name : PAY1

To DEFINE Aggregate Group, Specify:
 Description ==>  PAYROLL DATA
             ==>

 Backup Attributes
   Number of Copies  . . . . . . . 2         (1 to 15)
   Management Class Name . . . . . MC54NMIG  (1 to 8 characters, to be
                                              defined in current SCDS)

 Output Data Set Name Prefix . . . ABARS.PAY1
                                                    (1 to 33 Characters)
 Account . . . . . .
                                                    (1 to 32 Characters)
```

Figure 53. Aggregate Group Define Panel

```
                          AGGREGATE GROUP ALTER
Command ===>

SCDS Name  . . . . . : SYS1.SC54.SCDS1
Aggregate Group Name : PAY1
To Edit a Data Set, Specify Number  . . 6   (1, 2, 3, 4, 5, or 6)
  Selection Data Sets:            (1 to 44 characters)
   1 ===> 'ABARS.PAY1.SELECT'
     Member Name ===>             (1 to 8 characters)
   2 ===>
     Member Name ===>             (1 to 8 characters)
   3 ===>
     Member Name ===>             (1 to 8 characters)
   4 ===>
     Member Name ===>             (1 to 8 characters)
   5 ===>
     Member Name ===>             (1 to 8 characters)

  Instruction Data Set:           (1 to 44 characters)
   6 ===> 'ABARS.PAY1.INSTRUCT'
```

Figure 54. Aggregate Group Alter Panel

In Figure 53 you can see that we have chosen to describe our aggregate group, PAY1, as PAYROLL DATA, make two copies of output files, take our management class definitions from management class MC54NMIG, and use ABARS.PAY1 as the data set prefix for the output data sets created by ABACKUP.

In Figure 54 we have chosen to call our selection data set ABARS.PAY1.SELECT and our instruction data set ABARS.PAY1.INSTRUCT.

2. **What is a selection data set and does ABARS use it?**

   A selection data set lists the names of the data sets to be processed during ABACKUP. You can create a selection data set by means of the Aggregate Group Alter panel, or you can predefine one yourself. Here we show you how to set up a basic aggregate with an INCLUDE statement. However, there are other parameters that you can specify in your selection data set.

The *DFSMShsm Storage Administration Guide* comprehensively details these additional optional parameters:

- EXCLUDE
- ACCOMPANY
- ACCOMPANYEXCLUDE
- ALLOCATE
- ALLOCATEEXCLUDE

Our selection data set, which we created from the ISMF panels, is called ABARS.PAY1.SELECT and has the following INCLUDE statement:

```
INCLUDE(MHLRES5.PAY1.**)
```

ABACKUP uses this statement and selects cataloged data sets that have a high-level qualifier of MHLRES5 and a second-level qualifier of PAY1.

3. **What is an instruction data set and how is it used?**

   The instruction data set is an optional data set that is free-form text meant to include information necessary in assisting with the recovery process at the recovery site.

   You may find it useful to include the following types of information:

   - SMS attributes
   - Description of the application
   - RACF environment
   - Software requirements
   - Hardware requirements
   - Unique application dependencies

   Our instruction data set, ABARS.PAY1.INSTRUCT, contains the following basic information:

```
THIS IS THE PAYROLL COLLECTION OF DATA SETS.
ABACKUP WAS FROM SMS 1.4
RESTORE TO 1.4 SYSTEM ONLY.
```

4. **Are there any changes that I have to make to the management classes to use ABARS?**

   In our sample management class, MC54NMIG, we specified our attributes for ABACKUP as shown below.

   Figure 55 on page 117

```
                          MANAGEMENT CLASS ALTER
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS1
Management Class Name : MC54NMIG

To ALTER Management Class, Specify:
  AGGREGATE Backup Attributes:
    # Versions . . . . . 4        (1 to 9999, NOLIMIT or blank)
    Retain Only Version  . . . 5  (1 to 9999, NOLIMIT or blank)
      Unit . . . . . D            (D=days, W=weeks, M=months,
                                     Y=years or blank)
    Retain Extra Versions  . . 5  (1 to 9999, NOLIMIT or blank)
      Unit . . . . . D            (D=days, W=weeks, M=months,
                                     Y=years or blank)
    Copy Serialization . . . . C  (C=continue, F=fail or blank)
    Abackup Copy Technique . . S  (P=Conc Preferred, R=Conc
                                     Required or S=Standard)
```

Figure 55. Management Class Alter Panel 5

For this management class we asked that:

- The maximum number of aggregate versions kept is 4.
- The most recent backup version of an aggregate group is kept for 5 days.
- The next most recent backup version of an aggregate group is kept for 5 days.
- ABACKUP should continue if an enqueue failure is encountered.
- Concurrent copy is not used to perform ABACKUP.

5. **Do I need to make any changes to my ARCCMDxx member for ABARS use?**

   There are parameters that you need to consider adding to your ARCCMDxx PARMLIB member. Here is a sample of the definitions that we used on our system:

```
SETSYS                        /* RECOVER DATA SET AGGREGATES TO    */ -
  ARECOVERUNITNAME(3490)      /* 3490 TAPE DEVICES.                */

SETSYS                        /* START A MAXIMUM OF TWO ADDRESS    */ -
  MAXABARSADDRESSSPACE(2)     /* SPACES FOR BACKING UP AND         */
                              /* RECOVERING AGGREGATED DATA SETS   */

SETSYS                        /* START THE SECONDARY ADDRESS       */ -
  ABARSPROCNAME(ABARS54)      /* SPACE WITH THE STARTUP PROCEDURE  */
                              /* NAMED ABARS54.                    */

SETSYS                        /* WRITE THE ABARS ACTIVITY LOG TO   */ -
  ABARSACTLOGTYPE(DASD)       /* DASD                              */

SETSYS                        /* LOG ALL ABARS MESSAGES            */ -
  ABARSACTLOGMSGLVL(FULL)

SETSYS                        /* RECOVER ML2 DATA SETS TO TAPE.    */ -
  ARECOVERML2UNIT(3490)

SETSYS                        /* USE 90% OF THE AVAILABLE TAPE FOR */ -
  ARECOVERPERCENTUTILIZED(090) /* ARECOVERY TAPES.                 */

SETSYS                        /* BACKUP AGGREGATES TO 3490 DEVICES.*/ -
  ABARSUNITNAME(3490)

SETSYS                        /* BACKUP ABARS DATA SETS WITH TWO   */ -
  ABARSBUFFERS(2)             /* DATA MOVEMENT BUFFERS.            */

SETSYS                        /* SPECIFY ABARS TO STACK THE        */ -
  ABARSTAPES(STACK)           /* ABACKUP OUTPUT ONTO A MINIMUM     */
                              /* NUMBER OF TAPE VOLUMES            */

SETSYS                        /* ABARS ACTIVITY LOG WILL NOT BE    */ -
  ABARSDELETEACTIVITY(N)      /* AUTOMATICALLY DELETED DURING      */
                              /* ABARS PROCESSING                  */

SETSYS                        /* SET PERFORMANCE OF BACKING UP     */ -
  ABARSOPTIMIZE(3)            /* LEVEL 0 DASD DATASETS             */

SETSYS                        /* TARGET DATASET IS TO BE ASSIGNED  */ -
  ARECOVERTGTGDS(SOURCE)      /* SOURCE STATUS                     */
```

We do not go into detail as to what each of these options allows. Our intent is to show the considerations that you need to take into account should you decide that you want to implement ABARS.

6. **Do I need to start an ABARS secondary address space?**

   No. However, if you want to use ABARS, you must define a procedure that DFSMShsm will use and must be added to SYS1.PROCLIB. You do not have to issue an MVS START command for (in our case) the procedure ABARS54. DFSMShsm schedules the start internally when an ABACKUP or ARECOVER task starts. For more information about the ABARS procedure refer to 2.2.2, “ABARS” on page 25.

7. **How do I perform an ABACKUP?**

As long as you have first specified the data sets that you want to back up in your selection data set, you can use the DFSMShsm ABACKUP command or ISMF panels to back up the data sets to tape.

In the following figures we show how to use the ISMF panels to perform ABACKUP. In Figure 56 we select option 5 to drive our ABACKUP.

```
                    AGGREGATE GROUP APPLICATION SELECTION
Command ===>

To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . 'ACTIVE'
                                (1 to 44 Character Data Set Name or 'Active')
  Aggregate Group Name . . PAY1      (for Aggregate Group List, fully or
                                     Partially Specified or * for All)
Select one of the following Options:
  5  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria
```

Figure 56. ABACKUP Entry Panel

You are prompted to specify ACTIVE for the CDS name for the ABACKUP.

In Figure 57 on page 120 we specify the following for the ABACKUP command for aggregate group PAY1:

- 3490 as the unit name for the ABACKUP output files
- Processing option of 2 to create backup copies of the aggregate data sets

  Note: If we had chosen option 1, Verify would determine whether any errors exist that would prevent successful completion of ABACKUP.
- We do not want to wait for ABACKUP to complete before we are returned to ISMF.
- Output files will be stacked onto the minimum number of tapes.
- The default SETSYS ABARSOPTIMIZE value will be taken.
- Data sets will not be deleted after they have been processed.
- The list of data sets that have been backed up will be written to a data set called MHLRES5.ABARS.
- We want to process the data sets wherever they reside.

```
                          AGGREGATE GROUP BACKUP
Command ===>

CDS Name : ACTIVE

  Aggregate Group Name  . . PAY1

  Unit Name . . . . . . . . 3490
  Processing Option . . . . 2                  (1=Verify, 2=Execute)
  Wait for Completion . . . N                  (Y or N)
  Stack / Nostack . . . . . S                  (S=STACK, N=NOSTACK
                                                or blank)
  Optimize  . . . . . . . .                    (1 to 4 or blank)
  Delete Data Sets After Abackup  . . N        (Y or N)

  Filter Output Data Set Name (1 to 44 Characters)
    ===> MHLRES5.ABARS

  Process only                LO         ML1        ML2        USERTAPE
```

Figure 57. Aggregate Group Backup Panel

Figure 58 shows the ABACKUP command with the options we specified in Figure 57. We can submit the ABACKUP job by selecting option 1, which will schedule an ABACKUP task.

```
                          AGGREGATE  GROUP  BACKUP
Command ===>


DFHSM Command and Processing Option:
NOWAIT ABACKUP PAY1 UNIT(3490) EXECUTE STACK
FILTEROUTPUTDATASET('MHLRES5.ABARS')



Enter 1 to Submit DFHSM ABACKUP  COMMAND
Enter 2 to Save Generated ABACKUP  PARAMETERS

  Select Option  . . 1                       (1=SUBMIT, 2=SAVE)
```

Figure 58. ABACKUP ISMF Panel 2

8. **How do I check whether ABACKUP has been successful?**

   The activity log on DASD contains all the messages relating to ABARS. On our system the data set name that is generated is:

```
HSMACT.H1.ABACKUP.PAY1.D98210.T133233
```

   You can see that the aggregate name that is being backed up is included as a qualifier for easy identification.

9. **What does the output that is generated look like?**

   The output to the data set includes the DFSMSdss JCL and joblog, the data sets that have been successfully backed up and, if any have failed, an indication as to the reason.

   The output that is written to the data set looks like this:

```
PAGE 0001  DFSMSHSM 1.4.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   98.210  13:32
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP PAY1, AT 13:32:33, STARTED TASK = ABARS54.ABAR0132
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP PAY1 WILL BE ROUTED TO HSMACT.H1.ABACKUP.PAY1.D98210.T133233
ARC6004I 0061 ABACKUP PAGE 0001     5695-DF175  DFSMSDSS V1R4.0  DATA SET SERVICES     1998.210 13:39
ARC6004I 0061 ABACKUP ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 0061 ABACKUP ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
ARC6004I 0061 ABACKUP  DUMP DATASET(FILTERDD(SYS00006)) -
ARC6004I 0061 ABACKUP  OUTDDNAME(SYS00004, -
ARC6004I 0061 ABACKUP            SYS00005  -
ARC6004I 0061 ABACKUP                 ) OPTIMIZE(3) SPHERE                        -
ARC6004I 0061 ABACKUP  ALLDATA(*) FORCECP(0) -
ARC6004I 0061 ABACKUP  SHARE TOLERATE(ENQFAILURE)
ARC6004I 0061 ABACKUP ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ARC6004I 0061 ABACKUP ADR109I (R/I)-RI01 (01), 1998.210 13:39:19 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC6004I 0061 ABACKUP ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC6004I 0061 ABACKUP ADR006I (001)-SETUP(01), 1998.210 13:39:19 EXECUTION BEGINS
ARC6075I TAPE VOLUME M195AX SUCCESSFULLY ADDED TO ABARS RACF TAPE VOLUME SET
ARC6075I TAPE VOLUME M195AV SUCCESSFULLY ADDED TO ABARS RACF TAPE VOLUME SET
ARC6004I 0061 ABACKUP ADR801I (001)-DTDSC(01),
ARC6004I 0061 ABACKUP DATA SET FILTERING IS COMPLETE. 11 OF 11 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0
ARC6004I 0061 ABACKUP FAILED FOR
ARC6004I 0061 ABACKUP                          OTHER REASONS.
ARC6004I 0061 ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARYA
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY1
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY2
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY3
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY4
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY5
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY6
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY7
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY8
ARC6004I 0061 ABACKUP                           MHLRES5.PAY1.SALARY9
ARC6004I 0061 ABACKUP ADR013I (001)-CLTSK(01), 1998.210 13:40:26 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 0061 ABACKUP ADR012I (SCH)-DSSU (01),
ARC6004I 0061 ABACKUP 1998.210 13:40:26 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC6004I 0061 ABACKUP PAGE 0001     5695-DF175  DFSMSDSS V1R4.0  DATA SET SERVICES     1998.210 13:40
ARC6004I 0061 ABACKUP ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 0061 ABACKUP ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO
ARC6004I 0061 ABACKUP  DUMP DATASET(INCLUDE( -
ARC6004I 0061 ABACKUP   MHLRES5.ABARS.PAY1.INSTRUCT                ,    -
ARC6004I 0061 ABACKUP   HSMACT.H1.ABACKUP.PAY1.D98210.T133233        )) -
ARC6004I 0061 ABACKUP  OUTDDNAME(SYS00004, -
ARC6004I 0061 ABACKUP            SYS00005  -
PAGE 0002  DFSMSHSM 1.4.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   98.210  13:32
ARC6004I 0061 ABACKUP                 ) OPTIMIZE(3) SPHERE                        -
ARC6004I 0061 ABACKUP  ALLDATA(*) FORCECP(0) -
ARC6004I 0061 ABACKUP  SHARE TOLERATE(ENQFAILURE)
ARC6004I 0061 ABACKUP ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ARC6004I 0061 ABACKUP ADR109I (R/I)-RI01 (01), 1998.210 13:40:31 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC6004I 0061 ABACKUP ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC6004I 0061 ABACKUP ADR006I (001)-SETUP(01), 1998.210 13:40:31 EXECUTION BEGINS
ARC6004I 0061 ABACKUP ADR801I (001)-DTDSC(01),
 DATA SET FILTERING IS COMPLETE. 2 OF 2 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED FOR
ARC6004I 0061 ABACKUP                          OTHER REASONS.
ARC6004I 0061 ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 0061 ABACKUP                           MHLRES5.ABARS.PAY1.INSTRUCT
ARC6004I 0061 ABACKUP                           HSMACT.H1.ABACKUP.PAY1.D98210.T133233
ARC6004I 0061 ABACKUP ADR013I (001)-CLTSK(01), 1998.210 13:40:47 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 0061 ABACKUP ADR012I (SCH)-DSSU (01),
 1998.210 13:40:47 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC6382I ACTIVITY LOG HSMACT.H1.ABACKUP.PAY1.D98210.T133233 HAS BEEN SUCCESSFULLY BACKED UP
ARC6382I INSTRUCTION DATA SET MHLRES5.ABARS.PAY1.INSTRUCT HAS BEEN SUCCESSFULLY BACKED UP
ARC6369I STORAGE REQUIREMENTS FOR AGGREGATE GROUP PAY1 ARE: LO=11K, ML1=0, ML2=0, TOTAL=11K
ARC6061I VOLUMES USED FOR CONTROL FILE ABARS.PAY1.C.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  M195AX
ARC6060I VOLUMES USED FOR DATA FILE ABARS.PAY1.D.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  M195AX
ARC6071I VOLUMES USED FOR INSTRUCTION/ACTIVITY LOG FILE ABARS.PAY1.I.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE
GROUP PAY1 ARE:
  M195AX
ARC6061I VOLUMES USED FOR CONTROL FILE ABARS.PAY1.C.C02V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  M195AV
ARC6060I VOLUMES USED FOR DATA FILE ABARS.PAY1.D.C02V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  M195AV
ARC6071I VOLUMES USED FOR INSTRUCTION/ACTIVITY LOG FILE ABARS.PAY1.I.C02V0001 DURING AGGREGATE BACKUP FOR AGGREGATE
GROUP PAY1 ARE:
  M195AV
ARC6055I AGGREGATE BACKUP HAS COMPLETED FOR AGGREGATE GROUP PAY1, AT 13:40:52, RETCODE = 000
```

10. **If anything should happen to this aggregate group of data sets, how do I perform an ARECOVER?**

    We show you how to perform a basic ARECOVER using aggregate group PAY1. We take all the defaults and specify the minimum amount of information to show you the ease with which ABARS allows a recovery operation to take place.

    In our case we are performing an ARECOVER on the assumption that we have lost all of our data sets that had a qualifier of MHLRES5.PAY1. We show only the panels that are related to our operation, not all of the possible

combinations that you can specify on an ARECOVER operation. The redbook *DFSMShsm ABARS and Mainstar Solutions*, SG24-5089, offers comprehensive coverage and suggestions about ABARS as does the manual *DFSMShsm Storage Administration Guide*, SH21-1076.

We now look at the panel flow that allows you to perform an ARECOVER of aggregate group PAY1.

In Figure 59 select option 6 to get into the ARECOVER ISMF panels.

```
                    AGGREGATE GROUP APPLICATION SELECTION
Command ===>

To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . 'ACTIVE'
                                (1 to 44 Character Data Set Name or 'Active')
  Aggregate Group Name . . PAY1      (for Aggregate Group List, fully or
                                     Partially Specified or * for All)
Select one of the following Options:
  6  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria
```

Figure 59. Arecover Option on Aggregate Group Application Selection Panel

You are prompted to specify ACTIVE for the CDS name for the ARECOVER.

The entries that we made in Figure 60 on page 123 for our ARECOVER indicate:

- The aggregate group name PAY1
- That we want data sets recovered to ML1 DASD to be backed up next time automatic backup runs
- That migrated data sets are recovered to ML1 DASD
- No rename of data sets is to take place.
- No conflict resolution of data set names is to take place.

```
                          AGGREGATE GROUP RECOVER               Page 1 of 7
Command ===>

  Abackup Control Dataset  . . ABARS.PAY1.C.CO1VOOO1
                                        (1 to 44 Characters)
    Xmit . . . . . . . . . . . N        (Y or N)
    Stack / Nostack  . . . . .          (S=STACK, N=NOSTACK or blank)

  Aggregate Group Name . . . . PAY1
    Date . . . . . . . . . . .          (yyyy/mm/dd)
    Version  . . . . . . . . .          (1 to 9999)

  Processing Option  . . . . . 3        (1=Prepare, 2=Verify, 3=Execute)
  Wait for Completion  . . . . N        (Y or N)
  Target GDG Data Set Status            (A=ACTIVE, D=DEFERRED, R=ROLLEDOFF,
                                        S=SOURCE or blank)
  Volume Count . . . . . . . .          (A=ANY, N=NONE or blank)
  Recover Instruction Data Set  . . N   (Y or N)
  Recover Activity Log  . . . . . . N   (Y or N)
```

Figure 60. Aggregate Group Recover Panel (Page 1 of 7)

The entries we made in Figure 61 for our ARECOVER indicate:

- Our control data set name
- That our ABACKUP files were not transmitted to the ARECOVER site
- A processing option of execute
- That we do not want to wait for ARECOVER to complete before we are returned to ISMF.
- We do not want the instruction data set or activity log to be recovered.

Once we have completed the Aggregate Group Recover, Page 1 of 7 panel, we are presented with Page 2 of 7 (Figure 61).

Figure 61 shows the ARECOVER JCL that is generated as a result of the options that we have specified. As we want to submit this ARECOVER JCL, we select option 2.

```
                          AGGREGATE GROUP RECOVER               Page 2 of 7
Command ===>

AGGREGATE GROUP NAME:     PAY1

  Model Entity . . . .
                                      (1 to 44 Characters)
  Target Unit Name . .

  Target Utilization Percentage . .   (1 to 100)

  Backup Migrated Data Sets . . . . Y   (Y or N)
  New Migration Level . . . . . . . 1   (1=ML1, 2=ML2, 3=SOURCELEVEL)

  Rename All to High Level Qualifier  . .          (1 to 8 characters)
  Rename Selected Data Sets . . . . . . . N        (Y or N)

  Specify Conflict Resolution Option  . . N        (Y or N)
```

Figure 61. Aggregate Group Recover Panel (Page 2 of 7)

```
                              AGGREGATE  GROUP  RECOVER            Page  7 of  7
Command ===>


DFHSM Command and Processing Option:
NOWAIT ARECOVER AGGREGATE(PAY1)
EXECUTE
MIGRATEDDATA(ML1)









Enter 1 to Submit DFHSM ARECOVER COMMAND
Enter 2 to Save Generated ARECOVER PARAMETERS

  Select Option  . . 2                          (1=SUBMIT, 2=SAVE)
```

Figure 62. Aggregate Group Recover Panel (Page 7 of 7)

11. **How do I know whether the ARECOVER has been successful?**

    In much the same way as with ABACKUP, you are provided with an ARECOVER activity log. In our example the generated name is:

```
HSMACT.H1.ARECOVER.PAY1.D98210.T152431
```

The output that is written to the data set looks like this:

```
PAGE 0001  DFSMSHSM 1.4.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   98.210  15:24
ARC6102I AGGREGATE RECOVERY STARTING USING CONTROL FILE DATA SET ABARS.PAY1.C.C01V0001, AT 15:24:31,
 STARTED TASK = ABARS54.ABAR0130
ARC6030I ACTIVITY LOG FOR CONTROL FILE DATA SET ABARS.PAY1.C.C01V0001 WILL BE ROUTED TO
HSMACT.H1.ARECOVER.PAY1.D98210.T152431
ARC6115I AGGREGATE RECOVERY USING CONTROL FILE DATA SET ABARS.PAY1.C.C01V0001 WILL USE VOLUMES
  M195AX
ARC6162I ARPOOL NOT FOUND FOR AGGREGATE GROUP PAY1 USING CONTROL FILE DATA SET ABARS.PAY1.C.C01V0001 - ARECOVER WILL
CONTINUE
ARC6004I 001D ARECOVER PAGE 0001     5695-DF175  DFSMSDSS V1R4.0  DATA SET SERVICES     1998.210 15:27
ARC6004I 001D ARECOVER ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 001D ARECOVER  RESTORE DATASET(FILTERDD(SYS00004)) -
ARC6004I 001D ARECOVER  INDDNAME(SYS00002) -
ARC6004I 001D ARECOVER  SPHERE  -
ARC6004I 001D ARECOVER  TGTGDS(SOURCE    ) -
ARC6004I 001D ARECOVER  CATALOG FORCE FORCECP(0)
ARC6004I 001D ARECOVER ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
ARC6004I 001D ARECOVER ADR109I (R/I)-RI01 (01), 1998.210 15:27:32 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC6004I 001D ARECOVER ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC6004I 001D ARECOVER ADR006I (001)-SETUP(01), 1998.210 15:27:32 EXECUTION BEGINS
ARC6004I 001D ARECOVER ADR780I (001)-TDDS (01),
ARC6004I 001D ARECOVER THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL DATA SET FORMAT AND WAS CREATED BY
ARC6004I 001D ARECOVER DFSMSDSS VERSION
ARC6004I 001D ARECOVER                          1 RELEASE 4 MODIFICATION LEVEL 0
ARC6004I 001D ARECOVER ADR396I (001)-NEWDS(01), DATA SET MHLRES5.PAY1.SALARY ALLOCATED, ON VOLUME(S): TOTTS2
ARC6004I 001D ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 001D ARECOVER DATA SET MHLRES5.PAY1.SALARY HAS BEEN CATALOGED IN CATALOG CATALOG.TOTICF2.VTOTCAT
ARC6004I 001D ARECOVER ADR489I (001)-TDLOG(01), DATA SET MHLRES5.PAY1.SALARY WAS RESTORED
ARC6004I 001D ARECOVER ADR396I (001)-NEWDS(01), DATA SET MHLRES5.PAY1.SALARYA ALLOCATED, ON VOLUME(S): TOTTS2
ARC6004I 001D ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 001D ARECOVER DATA SET MHLRES5.PAY1.SALARYA HAS BEEN CATALOGED IN CATALOG CATALOG.TOTICF2.VTOTCAT
ARC6004I 001D ARECOVER ADR489I (001)-TDLOG(01), DATA SET MHLRES5.PAY1.SALARYA WAS RESTORED
ARC6004I 001D ARECOVER ADR396I (001)-NEWDS(01), DATA SET MHLRES5.PAY1.SALARY1 ALLOCATED, ON VOLUME(S): TOTTS2
ARC6004I 001D ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 001D ARECOVER DATA SET MHLRES5.PAY1.SALARY1 HAS BEEN CATALOGED IN CATALOG CATALOG.TOTICF2.VTOTCAT
ARC6004I 001D ARECOVER ADR489I (001)-TDLOG(01), DATA SET MHLRES5.PAY1.SALARY1 WAS RESTORED
ARC6004I 001D ARECOVER ADR396I (001)-NEWDS(01), DATA SET MHLRES5.PAY1.SALARY2 ALLOCATED, ON VOLUME(S): TOTTS2
ARC6004I 001D ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 001D ARECOVER DATA SET MHLRES5.PAY1.SALARY2 HAS BEEN CATALOGED IN CATALOG CATALOG.TOTICF2.VTOTCAT
ARC6004I 001D ARECOVER ADR489I (001)-TDLOG(01), DATA SET MHLRES5.PAY1.SALARY2 WAS RESTORED
ARC6004I 001D ARECOVER ADR396I (001)-NEWDS(01), DATA SET MHLRES5.PAY1.SALARY3 ALLOCATED, ON VOLUME(S): TOTTS2
ARC6004I 001D ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 001D ARECOVER ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 001D ARECOVER                           MHLRES5.PAY1.SALARY
ARC6004I 001D ARECOVER                           MHLRES5.PAY1.SALARYA
ARC6004I 001D ARECOVER                           MHLRES5.PAY1.SALARY1
ARC6004I 001D ARECOVER                           MHLRES5.PAY1.SALARY2
ARC6004I 001D ARECOVER ADR013I (001)-CLTSK(01), 1998.210 15:27:43 TASK COMPLETED WITH RETURN CODE 0000
```

```
ARC6004I 001D ARECOVER ADR012I (SCH)-DSSU (01),
ARC6004I 001D ARECOVER 1998.210 15:27:43 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY RECOVERED USING AGGREGATE GROUP PAY1:
  MHLRES5.PAY1.SALARY
  MHLRES5.PAY1.SALARYA
  MHLRES5.PAY1.SALARY1
  MHLRES5.PAY1.SALARY2
ARC6103I AGGREGATE RECOVERY HAS COMPLETED FOR AGGREGATE GROUP PAY1, USING CONTROL FILE DATA SET ABARS.PAY1.C.C01V0001,
 AT 15:27:43, RETCODE = 000
```

The information includes the DFSMSdss generated JCL and joblog and messages that indicate the successful completion or otherwise of the ARECOVER. This ARECOVER has successfully restored all of our data sets.

The example that we have shown is a very simple one. Many installations will not be able to implement solutions as easy as this. For sites that are using SMS the implementation of ABARS is recommended.

Chapter 3. Operation Considerations

In this chapter we describe the following DFSMShsm that the system operator performs:

- Starting DFSMShsm
- Stopping DFSMShsm
- Holding DFSMShsm functions
- Releasing DFSMShsm functions
- Canceling queued DFSMShsm requests
- Restarting DFSMShsm after an abnormal end
- Changing DFSMShsm control parameters

We recommend that system operators take the time to become familiar with the DFSMShsm commands discussed in this chapter, their function, and their purpose.

3.1 Starting DFSMShsm

Before starting DFSMShsm you must create a startup procedure in your PROCLIB data set and an ARCCMDxx member, which contains a series of commands, created in your PARMLIB data set. Refer to 2.2, “PROCLIB and PARMLIB Setup” on page 20 for detailed information about creating and customizing your DFSMShsm PROCLIB and PARMLIB members.

1. **Is it possible to automatically start DFSMShsm?**

   At system initial program load (IPL) time you can specify that DFSMShsm is started automatically as part of the IPL process. This is accomplished by use of the PARMLIB member COMMNDxx that you are using in response to the request: i2 refid=msgs.IEA101A

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR OS/390 01.03.00 HBB6603
```

   This reply would direct the IPL process to use PARMLIB member COMMND88:

```
R 0,CLPA,CMD=88
```

   To start DFSMShsm automatically, you would need to have placed a start command that refers to your DFSMShsm started procedure in member COMMND88. In our system we named our start procedure HSM54 so we coded the following in our PARMLIB member:

```
COM='S HSM54'
```

   This will cause the start command for DFSMShsm to be automatically issued.

2. **Is it possible to start DFSMShsm manually?**

   If you have not added the command to start DFSMShsm in your COMMNDxx PARMLIB member, you can start DFSMShsm manually once your system is active by entering a start command at the system console. In our system we waited until the system control program initialized and then issued the following command for our HSM54 procedure:

```
S HSM54
```

3. **How can I tell whether DFSMShsm initialization has been successful?**

   There are two messages on the operator console with respect to DFSMShsm initialization. The first message that you should look for is:

```
ARC0001I DFSMSHSM  1.4.0 STARTING
```

   This self-explanatory message indicates that Version 1 Release 4 of DFSMShsm is starting.

   The second message to look for is:

```
ARC0008I DFSMSHSM INITIALIZATION SUCCESSFUL
```

   This message indicates that DFSMShsm has initialized successfully and is now ready to start receiving commands. You may want to verify this by issuing a DFSMShsm command such as QUERY ACTIVE and ensure that a response is received.

4. **What is the secondary address space and how do I start it?**

   An ABARS secondary address space is automatically started by DFSMShsm when required. You specify the maximum number of concurrent secondary address spaces, using the MAXABARSADDRESSSPACE parameter on the SETSYS command. When DFSMShsm is started, the MVS operating system reads the ABARS startup procedure to get information about the ABARS secondary address space.

5. **Can I start DFSMShsm as a batch job?**

   No. DFSMShsm can be run only as a started task.

3.2 Stopping DFSMShsm

Although it is not necessary to schedule regular starting and stopping of DFSMShsm, you will have to shut down DFSMShsm for a system IPL, to run maintenance against the DFSMShsm CDSs, or because of a problem with DFSMShsm itself. Here we show you how to stop DFSMShsm and provide some useful information pertaining to its shutdown.

1. **What is the recommended way to stop DFSMShsm?**

   We recommend that whenever it is necessary to shut down DFSMShsm that you use the MVS MODIFY command from the system console. In our sample system, HSM54, we would issue the following command:

```
F HSM54,STOP
```

This causes an orderly shutdown of DFSMShsm.

2. **What is meant by an orderly shutdown in DFSMShsm terms?**

   An orderly shutdown of DFSMShsm causes all secondary address spaces and each attached subtask to end. The CDSs are closed (unless DFSMShsm is running in a multiple host environment), and the joblog, DFSMShsm log, and activity log information is spooled to the printer.

3. **Is this the only way that I can stop DFSMShsm?**

   You can also issue the MVS STOP command to cause an orderly shutdown of DFSMShsm. If you were to issue the following command, an orderly shutdown would occur:

```
P HSM54
```

4. **How can I tell whether the shutdown has been successful?**

   There are two messages that you should pay attention to. The first message advises you that the STOP command has been issued:

```
ARC0016I DFSMSHSM SHUTDOWN HAS BEEN REQUESTED
```

   The next message that you should look for is this, which tells you that shutdown has completed successfully:

```
ARC0002I DFSMSHSM SHUTDOWN HAS COMPLETED
```

5. **Are there any other parameters that I can specify on the DFSMShsm STOP command?**

   Yes. You can specify the DUMP parameter on the STOP command:

```
F HSM54,STOP DUMP
```

   If you issue the STOP command with the DUMP parameter, an orderly shutdown does not take place, and processing is stopped immediately. DFSMShsm takes a dump of the DFSMShsm address space before it stops processing. We do not recommend that you use the dump parameter as valuable information can be lost. If you need a dump of the DFSMShsm address space for diagnosis purposes, we recommend using the MVS DUMP command. Refer to 4.6.4, “SVC Dumps” on page 156 for additional information about the use of the MVS DUMP command.

6. **I issued the DFSMShsm STOP command but nothing seemed to happen. Should I issue it again?**

   No. Issue the DFSMShsm STOP command only once.

7. **The shutdown seems to be taking a long time to complete. What is happening as part of shutdown?**

   DFSMShsm completes the current functions, usually ending them at a data set level, and consolidates SMF type 30 records as part of its shutdown. That is why it may take a while.

8. **How often does DFSMShsm check for shutdown?**

   As some commands and volume functions take a long time to complete, DFSMShsm periodically checks for a shutdown having been requested to prevent a delay in the shutdown process.

9. **What happens to DFSMShsm commands that have been entered but did not complete because DFSMShsm was shutdown?**

   As long as OS/390 has not been IPLed since DFSMShsm was shut down, user and HSENDCMD commands that did not complete are restarted as part of the DFSMShsm intialization process.

3.3 Holding DFSMShsm Functions

The HOLD command is used to selectively prevent or interrupt DFSMShsm functions from running without stopping DFSMShsm. You could use the command to prevent tape-related processing if, for example, you were drive constrained at a particular moment in time or if there were hardware errors.

For functions that process on a volume basis (backup, dump, migration, and recover) you can choose whether you want to interrupt the function at the end of the data set that is being processed or at the end of the volume that is being processed.

The HOLD command is nondestructive, but the held functions should be released as soon as practical. A function is held until it is released or DFSMShsm is shut down and restarted. Depending on the HOLD command you issue, you can prevent processing of all DFSMShsm functions or selectively choose the following functions to be held:

- Command, automatic secondary, and automatic volume space management
- Only automatic secondary and automatic volume space management
- Recall and deletion of a migrated data set
- Command backup and automatic backup
- Only automatic backup
- Aggregate backup
- Aggregate recovery
- Command dump and automatic dump
- Only automatic dump
- Audit command processing
- List
- Report
- Recovery and restore
- Recycle

- Logging
- Tape copy
- Tape replace
- Expiring backup versions

There is a lot of flexibility with the HOLD command. We discuss the HOLD commands that you will use most. For more details on the HOLD command refer to the *DFSMShsm Storage Administration Reference*.

1. **Why might I want to issue the HOLD command?**

   Using the HOLD command is a flexible alternative to stopping DFSMShsm during your prime working day. If the activity within DFSMShsm seems to be causing problems in terms of the demands that it is placing on system resources, you can selectively increase the amount of functions that are held. If possible, allow recall and recover processing to continue so that users can access their migrated and backed up data. Once the system or DFSMShsm problem is resolved, you can release the held functions on a selective basis.

   Another approach would be to hold everything and then gradually release at a function level.

2. **What happens to functions that have been held after a restart of DFSMShsm?**

   If a function has been held by the HOLD command, it is released once DFSMShsm has restarted successfully.

3. **Who can issue the HOLD command?**

   The HOLD command can be issued from the OS/390 system console or by a DFSMShsm-authorized user from a TSO terminal, using the HSENDCMD command.

4. **How do I hold all DFSMShsm functions?**

   All processing can be held by issuing the following command:

```
F HSM54,HOLD ALL
```

5. **How can I hold all automatic volume and automatic secondary space management processing?**

   To prevent or interrupt automatic volume and automatic secondary space management, issue either of the following commands:

```
F HSM54,HOLD AUTOMIGRATION

F HSM54,HOLD MIGRATION(AUTO)
```

   These commands do not prevent the running of any space management commands that were issued by command.

6. **How do I control whether the current operation finishes processing at the end of a data set or the end of a volume?**

You can influence the point at which current processing finishes for the following DFSMShsm functions:

- Space management (automatic or command)
- Backup (automatic or command)
- Dump (automatic or command)
- Recover
- Restore

Two optional parameters can be specified on the HOLD command:

```
ENDOFDATASET or EOD
ENDOFVOLUME or EOV
```

The DFSMShsm default is at the end of data set, except for the dump function, where it is end of volume. A requested volume dump is done in its entirety and then DFSMShsm stops dump processing if the DUMP function is held.

7. **How can I prevent tape recall requests from TSO users?**

   If you were to issue the following command you would allow only recall requests from tape submitted from batch jobs and all nontape volumes to be processed:

```
F HSM54,HOLD RECALL(TAPE(TSO))
```

   Note: Recalls from tape volumes that are already mounted will be satisfied.

8. **How can I prevent all recalls from tape volumes?**

   Issue the following command:

```
F HSM54,HOLD RECALL(TAPE)
```

9. **How can I tell what has been held by the HOLD command?**

   When you issue the following command you receive a series of messages that indicate which functions have been held:

```
F HSM54,QUERY ACTIVE
```

3.4 Releasing DFSMShsm Functions

As you might expect, if there is a command to prevent or interrupt DFSMShsm processing, there must be a command to perform the opposite of this. Any function that can be held can be released with the RELEASE command

The only time that functions will not be released is if journaling has been disabled. Once journalling has been disabled, it holds all DFSMShsm functions.

The RELEASE command will not be effective until the CDSs have been successfully backed up.

1. **Who can issue the RELEASE command?**

   The RELEASE command can be issued from the system console or also submitted by a DFSMShsm-authorized user from a TSO terminal, using the HSENDCMD.

2. **In the examples shown for the HOLD command, what are the corresponding RELEASE commands?**

   We show the HOLD command and its corresponding RELEASE command in the examples that follow.

```
F HSM54,HOLD ALL

F HSM54,RELEASE ALL
```

```
F HSM54,HOLD AUTOMIGRATION

F HSM54,RELEASE AUTOMIGRATION
```

```
F HSM54,HOLD RECALL(TAPE(TSO))

F HSM54,HOLD RELEASE RECALL(TAPE(TSO))
```

```
F HSM54,HOLD RECALL(TAPE)

F HSM54,RELEASE RECALL(TAPE)
```

3. **Can I RELEASE automigration if I have held migration?**

   No. If you hold the main function, you cannot issue a RELEASE for one of its subfunctions. In this case the following messages would be issued:

```
F HSM54,RELEASE AUTOMIGRATION
ARC0111I SUBFUNCTION MIGRATION(AUTO) CANNOT BE
ARC0111I (CONT.) RELEASED WHILE MAIN FUNCTION MIGRATION IS HELD
ARC0100I RELEASE COMMAND COMPLETED
```

4. **If I RELEASE a main function, will any associated subfunctions be released?**

   Yes.

There are as many variations of the RELEASE command as there are for the HOLD command. For further reference and for a comprehensive list of the commands, refer to the *DFSMShsm Storage Administration Reference*.

3.5 Canceling Queued DFSMShsm Requests

DFSMShsm allows you to cancel queued requests. A queued request is a request that has not yet been selected for processing by DFSMShsm. Once a request has been selected for processing it cannot be canceled by any DFSMShsm command.

However, if you have the DFSMS Optimizer HSM Monitor/Tuner installed, you can cancel active tasks under the following functions:

- Primary space management
- Secondary space management
- Automatic backup
- Automatic dump
- Interval migration

For more information please refer to the *DFSMS Optimizer: The New HSM Monitor/Tuner* manual.

In this section we will discuss the CANCEL command and give examples of its uses.

1. **Who can issue the CANCEL command?**

   The CANCEL command is primarily issued by the system operators from the system console. Additionally they can also be submitted by a DFSMShsm-authorized user from a TSO terminal.

2. **What can I cancel?**

   For DFSMShsm queued requests, you can cancel:

   - An individual command
   - A user ID
   - A data set name

3. **How can I find out which requests can be canceled?**

   You have to issue the DFSMShsm QUERY command to find the request numbers, user IDs, or data set names that you want to cancel. Use the following command to display all DFSMShsm requests:

```
F HSM54,QUERY REQUEST
```

   Refer to 5.2, “Query Command” on page 171 for more information about the QUERY command.

4. **If I issue the QUERY ACTIVE command, can I cancel any of the requests listed in the command output?**

   No. The requests listed as a result of the QUERY ACTIVE command are already being processed and are not available for the CANCEL command.

5. **How do I cancel requests for a single request?**

   Once you have determined the request number, issue the following command :

```
F HSM54,CANCEL REQUEST(num)
```

   where num represents the number of the request you want to cancel.

6. **How do I cancel the requests for a particular user?**

   You would issue the following command to cancel the requests for user VIALLI:

```
F HSM54,CANCEL USER(VIALLI)
```

7. **How would I cancel requests for a particular data set?**

   To cancel a request for data set name CHELSEA.RBEST you would issue the following command:

```
F HSM54,CANCEL DSNAME(CHELSEA.RBEST)
```

8. **Can I issue the CANCEL command to cancel a CANCEL command that I accidentally entered?**

   No. It is not possible to cancel a CANCEL command. Neither the CANCEL nor the QUERY command can be canceled.

For further information about the CANCEL command please refer to the *DFSMShsm Storage Administration Reference*.

3.6 Restarting DFSMShsm after an Abnormal End

DFSMShsm attempts to recover after any abnormal terminations. If DFSMShsm cannot recover from the abnormal end and the RESTART keyword is coded in the PROC statement of the startup procedure, DFSMShsm restarts itself. For more information on the PROCLIB setup, refer to 2.2, “PROCLIB and PARMLIB Setup” on page 20.

If an abnormal end occurs that interrupts MVS processing, and as long as the extended common service area (ECSA) is not destroyed, DFSMShsm can continue to process waiting requests. Additionally, if DFSMShsm is restarted, it will process any requests that are waiting in the ECSA.

1. **How can I tell whether an abnormal end has occured?**

   If a task processing within the DFSMShsm address space has abnormally ended, the following message is issued:

```
ARC0003I taskname TASK ABENDED, CODE abendcode IN MODULE modname AT OFFSET
           offset, STORAGE LOCATION location
```

This message provides you with information either to investigate the problem yourself or to raise an incident with IBM Software Support.

2. **Will the ARC0003I message be issued for tasks within the ABARS secondary address space?**

   No. ABARS issues the following variation of the message:

```
ARC6035E taskname TASK ABENDED, CODE abendcode IN MODULE modname AT OFFSET
           offset, STORAGE LOCATION location
```

   This message provides you with information either to investigate the problem yourself or to raise an incident with IBM Software Support.

3. **What else should I consider after an abnormal end has occured?**

   Try to understand the area of processing that was affected. DFSMShsm attempts recovery in most cases, but the ultimate responsibility lies with the user to determine whether any further recovery actions are required.

   You should use these as valuable sources of information to analyze any problems:

   - System log
   - DFSMShsm activity logs
   - DFSMShsm PDA trace
   - DFSMShsm generated dumps
   - DFSMShsm X/Y logs

4. **What will happen to the automatic functions of DFSMShsm?**

   If the recovery from an abnormal end has taken a long time, you may need to extend the windows for your automatic schedule. The DFSMShsm restart can occur after the automatic function window has been closed or the automatic function window may be shortened because of the abnormal end so that all volumes will not be processed in the remaining time available.

   Consider issuing the SETSYS command to adjust the automatic functions window. See 3.7, “Changing DFSMShsm Control Parameters.”

Although it is not possible to document all of the possible reasons and recovery actions for a DFSMShsm abnormal end, just being aware of the information to look for may help you to prevent or recover easily from any interruption to your environment.

3.7 Changing DFSMShsm Control Parameters

Each time you start DFSMShsm, a subset of parameters is established by default. You may want to change the defaults during normal operation, change the times that the automatic functions are scheduled to run, or increase the number of tasks related to a specific function.

Use the SETSYS command to change your control parameters. In this section we show you some examples that give an idea of the ease with which you can issue the SETSYS commands, and the effect that they will have on your environment. The SETSYS command can be coded in your ARCCMDxx

PARMLIB member, as described in 2.2, “PROCLIB and PARMLIB Setup” on page 20.

1. **Who can issue the SETSYS command?**

   Primarily we would expect that a DFSMShsm-authorized storage administrator would issue these commands, using the HSENDCMD from a TSO terminal. These commands can also be issued by the system operator from the system console.

2. **How can I modify the timings of the automatic functions?**

   In our example we issue the command from the system console, using the MVS MODIFY command. The following commands will override the current settings for automatic primary and secondary space management, backup, and dump:

```
F HSM54,SETSYS PRIMARYSPMGMSTART(0400 0500)

F HSM54,SETSYS SECONDARYSPMGMSTART(0200 0300)

F HSM54,SETSYS AUTOBACKUPSTART(0600 0700 0800)

F HSM54,SETSYS AUTODUMPSTART(0900 1000 1100)
```

   Note: These are examples only and not intended for use on your system.

3. **How can I alter the maximum number of migration and backup tasks?**

   Issue the following commands:

```
F HSM54,SETSYS MAXMIGRATIONTASKS(6)

F HSM54,SETSYS MAXBACKUPTASKS(9)
```

There are many SETSYS commands that you can change during normal operation. The SETSYS commands are documented in the *DFSMShsm Storage Administration Reference*.

Chapter 4. Administration

To keep DFSMShsm up and running, perform these administrative activities on a regular basis:

- Expiring old data set backup versions
- Recycling tapes and converting to new tape volumes
- Moving data sets to new DASD volumes
- Reorganizing the CDSs
- Journal data set maintenance
- Problem determination
- Auditing DFSMShsm

We use the HSENDCMD command to issue DFSMShsm storage administrator or system programmer commands to DFSMShsm from a TSO terminal. You must be authorized by the DFSMShsm AUTH command or by RACF to issue these commands. See 2.3.4, “Controlling the Use of DFSMShsm Commands” on page 43 for more information about authorizing users to issue system administrator commands.

4.1 Expiring Backup Versions

Availability management of DFSMShsm does not automatically delete backup versions when the related user data set is deleted. Over time a number of unneeded backup versions can be accumulated on backup volumes. You can use the EXPIREBV command to find and delete these unneeded backup versions based on the most recent status of the data set.

The EXPIREBV command causes DFSMShsm to search the BCDS for old, unwanted backup versions and to delete them (or just display information about them) based on the attributes in the management class for each data set.

You can use the EXPIREBV command with the DISPLAY parameter to see which backup copies are eligible for deletion. The following command places a list of the eligible backup versions into the specified output data set:

```
HSENDCMD EXPIREBV DISPLAY OUTDATASET('dsname')
```

If you browse the data set pointed to by the OUTDATASET parameter, you will find the output from this command (see Figure 63).

```
DISPLAY OF BACKUP VERSIONS ELIGIBLE FOR EXPIRATION AT 12:04:04 ON 1998/07/31 FOR SYSTEM=SC54

COMMAND INPUT: (DEFAULTS)

DSNAME = MHLTST4.BRODCAST                    DELETED*, WAS SMS
  (* DETERMINED ON 1998/07/30)
MANAGEMENT CLASS USED = MC54TEST

BACKUP VERSION DSNAME                SYS  GEN     RET  BACK
                              CAT  NMBR AGE  VERS PROF
HSM.BACK.T530916.MHLTST4.BRODCAST.I8210    YES  001  002  NO   NO

DSNAME = MHLTST4.COMPINFO                    DELETED*, WAS SMS
  (* DETERMINED ON 1998/07/30)
MANAGEMENT CLASS USED = MC54TEST

BACKUP VERSION DSNAME                SYS  GEN     RET  BACK
                              CAT  NMBR AGE  VERS PROF
HSM.BACK.T540916.MHLTST4.COMPINFO.I8210    YES  001  002  NO   NO


             .
             .
             .

   END OF DISPLAY - 00000015 BACKUP VERSIONS ELIGIBLE FOR EXPIRATION
```

Figure 63. Output from EXPIREBV DISPLAY Command

To expire the backup versions, issue the following command:

```
HSENDCMD EXPIREBV EXECUTE
```

Figure 64 on page 141 shows the output from the EXPIREBV EXECUTE command found in the backup activity log (for more about activity logs, see 4.6.2, “Activity Logs” on page 153).

```
 DFSMSHSM BACKUP LOG, TIME 14:10:45,  DATE 98/07/30
ARC0680I EXPIRE BACKUP VERSIONS STARTING AT 12:43:39 ON 1998/07/31, SYSTEM SC54
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T530916.MHLTST4.BRODCAST.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T540916.MHLTST4.COMPINFO.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T550916.MHLTST4.C2.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T560916.MHLTST4.C2IMBED.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T560916.MHLTST4.C2SMSUP.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T570916.MHLTST4.C2STPROC.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T580916.MHLTST4.C2SYSBS.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T580916.MHLTST4.C2SYSSP.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T590916.MHLTST4.C3EXPBV.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T001016.MHLTST4.C3RECYCL.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T011016.MHLTST4.C4AUDIT.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T021016.MHLTST4.DFSMSOPT.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T031016.MHLTST4.LOG.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.T041016.MHLTST4.RMFOS130.I8210
ARC0734I ACTION=EXBACKV FRVOL=****** TOVOL=****** TRACKS=  *** RC=  0, REASON=   0, AGE=  2,
 DSN=HSM.BACK.U041016.MHLTST4.RMFOS130.I8210
ARC0681I EXPIRE BACKUP VERSIONS ENDING AT 12:43:42 ON 1998/07/31, NUMBER OF VERSIONS EXPIRED=15,
 LAST BCDS KEY PROCESSED=OMVS54.TCPIPOE.ROUTED.PROFILE,
 PLANNED ENDING KEY WAS=Z999999999999999999999999999999999999999999999999
 DFSMSHSM BACKUP LOG, TIME 12:43:42,  DATE 98/07/31
```

Figure 64. Output from EXPIREBV EXECUTE Command

When you initially use EXPIREBV, you must issue it twice to delete unwanted versions. The first time you issue the EXPIREBV command, DFSMShsm determines that the user data set was scratched and stores the date of the first EXPIREBV execution as the scratch date in the DFSMShsm BCDS data set record (MCB). You can look at this date by using the following command:

```
HSENDCMD FIXCDS B 'dsname' OUTDATASET('out_ds')
```

In the data set pointed to by the OUTDATASET parameter, you will see the MCB record for the data set (see Figure 65 on page 142). The date EXPIREBV determined as the date on which the user data set was deleted is at offset X'48'. A subsequent EXPIREBV EXECUTE command deletes the unwanted backup versions when the requested time has passed. The requested time is specified in the management class for SMS data sets, or in the EXPIREBV command for non-SMS data sets.

```
FIXCDS B 'MHLTST4.C2SMSUP' OUTDATASET('MHLRES4.FIX.LIST')
MCH=   01102000 B0D4411D A32CDD01 B0D31AFC 177F5C06                              *     M       L                 *
+0000  C8E2D4F1 F4C4FFFF 00000000 00000000 16104368 009821OF 40007FF8 00508000   *HSM14D                     8    *
+0020  00000001 00008837 00000012 0000A000 00020002 0002FFFF 009821OC 00000000   *                                *
+0040  00000000 00000000 0098211F 00000000 C8E2D44B C2C1C3D2 4BE3F4F3 F1F0F1F6   *                HSM.BACK.T431016*
+0060  4BD4C8D3 E3E2E3F4 4BC3F2E2 D4E2E4D7 4BC9F8F2 F1FO4040 40404040 C8E2D4F1   *.MHLTST4.C2SMSUP.I8210      HSM1*
+0080  F4C3C200 009821OF 00000002 00000000 C8E2D44B C2C1C3D2 4BE3F5F6 F0F9F1F6   *4CB             HSM.BACK.T560916*
+00A0  4BD4C8D3 E3E2E3F4 4BC3F2E2 D4E2E4D7 4BC9F8F2 F1FO4040 40404040 C8E2D4F1   *.MHLTST4.C2SMSUP.I8210      HSM1*
+00C0  F4C3C200 009821OF 00000001 00000000                                       *4CB                             *
ARC0197I TYPE B, KEY MHLTST4.C2SMSUP, FIXCDS DISPLAY SUCCESSFUL
```

Figure 65. Example of a FIXCDS Display of an MCB Record

We recommend that you run the EXPIREBV EXECUTE command on a regular basis.

The EXPIREBV command allows you to delete all expired backup versions from all of your backup volumes. You can also delete the backup versions of specific data sets by issuing the BDELETE or HBDELETE commands.

The following command can delete the specified backup version n of dsname:

```
HSENDCMD BDELETE (dsname) VERSIONS(n)
```

4.2 Recycling Tape Volumes

As time goes by, migrated data sets expire, are recalled, or are marked for deletion by DELETE and HDELETE commands. Similarly, tape backup data sets are rolled off by automatic backup, or they are marked for deletion by BDELETE, HBDELETE, and EXPIREBV commands. Those data sets are still physically occupying space on the tape volumes. Over time, the tape volumes on which the invalidated data resides become fragmented, resulting in inefficiently used tape media.

In an environment where there is a lot of recall activity due to a poorly tuned environment, you may have many partial tapes. These tapes were taken away from a migration or recycle task due to a recall for a data set.

To remove the invalid data sets and consolidate the valid data on fewer tape volumes, you can use the RECYCLE command.

You specify to DFSMShsm when a tape becomes eligible for recycle processing by setting a threshold value on the SETSYS RECYCLEPERCENT and ML2RECYCLEPERCENT parameters in the ARCCMDxx member. You can also specify a value for the PERCENTVALID subparameter of the RECYCLE command if you need to override the SETSYS values.

Optionally, you may want to specify the maximum number of ML2 partial tapes you may have in your HSMplex that are not associated with an ML2 output task. Use the ML2PARTIALSNOTASSOCIATEDGOAL parameter of the SETSYS command, as explained in 2.7, “Space Management” on page 78.

Here we describe the most common commands you can use to consolidate the DFSMShsm-owned tape data. The RECYCLE command can be issued from an

operator console or by using the HSENDCMD command. The HSENDCMD command allows you to issue the RECYCLE command from any TSO user ID that you have authorized with the AUTH command.

1. **How do I display the tape volumes eligible for recycle?**

   The DISPLAY parameter of the RECYCLE command allows you to see which tapes have a low percentage of valid data. This percentage may have been specified with the SETSYS ML2RECYCLEPERCENT parameter, the SETSYS RECYCLEPERCENT, or the PERCENTVALID parameter of the RECYCLE command.

   You can use any of the following statements, depending on the set of tapes that you want to recycle:

   - (a) displays all of the eligible tapes
   - (b) displays both daily and spill backups eligible to recycle
   - (c) displays only daily backup tapes eligible to recycle
   - (d) displays the spill volumes eligible to recycle
   - (e) displays only ML2 tapes eligible to recycle

```
(a) HSENDCMD RECYCLE DISPLAY ALL

(b) HSENDCMD RECYCLE DISPLAY BACKUP

(c) HSENDCMD RECYCLE DISPLAY DAILY(day)

(d) HSENDCMD RECYCLE DISPLAY SPILL

(e) HSENDCMD RECYCLE DISPLAY ML2
```

2. **How do I generate a list of the volumes eligible for recycle?**

   The VERIFY parameter of the RECYCLE command allows you to get two lists of the eligible volumes:

   - A pull list that operators can use to manually pull the required tapes for the RECYCLE processing. The pull list is made up of small groups of alphabetized volser lists. The pull groups are listed in the sequence in which recycle processing will most likely request them.
   - A mount list that aids operators with the mount order of tapes being recycled. The mount list is in the anticipated order of the tape mounts that RECYCLE processing will request. The list of tapes is grouped according to the category (for example, ML2 and SPILL) requested.

   These lists can be helpful in an environment with a large number of tape cartridges and without automated tape libraries.

   Both lists are dynamically allocated to either a SYSOUT data set or to the data set having the prefix that you specified with the TAPELIST PREFIX(tapelist_prefix) parameter. Optionally, you can specify a fully qualified data set name for the output of these two lists with the RECYCLE command TAPELIST FULLDSNAME(tapelist_dsn) parameter. The data set is deallocated when the command completes.

The following command causes DFSMShsm to look for all eligible ML2 tapes and create the pull and mount lists in a data set that will have prefix as its high- level qualifier:

```
HSENDCMD RECYCLE ML2 VERIFY TAPELIST(PREFIX(prefix))
```

3. **How do I start the recycle process?**

   The EXECUTE parameter of the RECYCLE command specifies that DFSMShsm should recycle a specific tape volume, a category of backup volumes, tape ML2 volumes, or a combination of all volumes. Volumes that meet the eligibility criteria are recycled.

   The PERCENTVALID parameter specifies the maximum percentage of valid data that a volume can have and still be eligible for recycling.

   The following command causes DFSMShsm to recycle all eligible ML2 tapes that contain 25% or less valid data:

```
HSENDCMD RECYCLE ML2 EXECUTE PERCENTVALID(25)
```

   If you have selected the SCRATCHTAPE option on the SETSYS TAPEDELETION parameter, DFSMShsm tells the tape management system that it has released the ownership of the tape volume, so it can be returned to the installation′s scratch tape pool. If SETSYS TAPEDELETION(HSMTAPE) is specified the empty tape remains in the DFSMShsm-owned tape pool for the exclusive use of DFSMShsm. We do not recommend that you specify TAPEDELETION(HSMTAPE).

   The tape security environment you select may have an effect after recycle processing. If PASSWORD or EXPIRATION has been selected as one of the SETSYS TAPESECURITY parameters, a recycled tape must be reinitialized in order to be used outside DFSMShsm.

   We recommend that you recycle your tapes on a regular basis, to avoid running out of scratch or DFSMShsm-owned tapes. You may choose a window of less tape activity at your installation in order to run the RECYCLE.

   Finally, consider the tape copies of recycled volumes. After a tape has been recycled, the tape copy is no longer useful. When you process the RECYCLE command, DFSMShsm invalidates alternate or duplex tapes at the same time that it invalidates the tapes from which they were copied.

4. **Can I limit the number of volumes that are recycled at one time?**

   Yes. The LIMIT parameter allows you to specify that RECYCLE should quiesce after the net specified number of tapes has been returned to scratch status. The following command causes RECYCLE processing to quiesce after a net of 40 tapes is returned to scratch status. Net is determined by tapes read in for input minus tapes written to for output.

```
HSENDCMD RECYCLE ML2 EXECUTE PERCENTVALID(25) LIMIT(40)
```

Using our example, during the RECYCLE process you may empty 44 ML2 tapes, but mount 4 scratch tapes for output. This results in the net 40 tapes, and so RECYCLE processing will quiesce after processing a total of 44 tapes. The LIMIT parameter is subject to the number of tapes allowed by the PERCENTVALID parameter.

4.2.1 Conversion to Another Tape Media

You can use the RECYCLE command to substitute your old tapes for new media type tapes. You can choose the tape volumes that will be replaced by using the SELECT parameter of the RECYCLE command. The SELECT parameter and its related keywords were introduced by APAR OW27348, to improve the granularity of the RECYCLE command.

1. **Can I specify a range of old media type to convert to a new one?**

   Yes. The command below causes DFSMShsm to recycle all tape volumes that have volsers in the range of TAPE00 through TAPE99. This command could be used if you are planning to convert from 3490 media tapes to 3590 media. The TAPE00 through TAPE99 range in our example is the range of volsers of our old 3490 tapes. All of the tapes in the range will be recycled, as we are using the PERCENTVALID(100)parameter.

```
HSENDCMD RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TAPE00:TAPE99)))
      PERCENTVALID(100)
```

2. **How do I exclude a range of tapes from the conversion process**?

   You can use the following command if you know that one of your tapes, within the range of tape volsers that will be recycled, is not available. For example, the operator cannot find tape volume TAPE04, so we exclude it from the recycle processing:

```
HSENDCMD RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TAPE00:TAPE99))
                 EXCLUDE(RANGE(TAPE04:TAPE04)))
```

You can find a detailed example of migrating from 3480, 3480x, 3490 and 3490E tapes to IBM 3590 Magstar emulating 3490E tapes in 6.6, “Emulating 3490 Cartridges on a 3590 Device” on page 219.

4.2.2 Recycle Performance

Recycle performance has been improved by DFSMShsm OW29231. DFSMShsm can use three data buffers for RECYCLE processing, with the following exceptions:

- The FORCE parameter is specified on the RECYCLE command

  or

- The NOLIMIT option is specified on the SETSYS TAPEUTILIZATION command.

4.3 Moving Data Sets to New DASD Volumes

Over time you will most likely replace existing DASD volumes with new ones. This will require that you move the data sets on your level 0, level 1, and ML2 volumes to the new devices.

4.3.1 Converting Level 0 DASD Volumes

DFSMSdss can be used to perform a device conversion of level 0 DASD volumes. There are migration tools that use mirroring to convert the old DASD volumes to new ones. You can also use DFSMShsm to move a single data set or an entire level 0 volume to other level 0 volumes.

The following command moves all DASD data sets that are on level 0 volume HSM14A to other level 0 volumes.

```
HSENDCMD MIGRATE VOLUME(HSM14A) MIGRATE(0)) CONVERT
```

When processing SMS-managed data sets, you cannot direct the movement to a specific volume. Although you can specify a volume, SMS-processing determines the actual volume to which the data set is moved. The process for moving data sets first migrates the data sets to ML1 volumes and then recalls them to level 0 volumes. At the beginning of processing for MIGRATE VOLUME commands, DFSMShsm obtains a list of the management classes active in the system. As it processes each data set, DFSMShsm checks the value specified for the data set which the COMMAND OR AUTO MIGRATE attribute in the management class. If the value of the attribute is BOTH, DFSMShsm processes the data set.

In recalling each data set, DFSMShsm invokes the ACS selection services of DFSMSdfp. If SMS is active, the ACS routine may return to DFSMShsm a storage class and, optionally, a management class. If the ACS routine returns a storage class, DFSMShsm passes the data set name, along with its storage class name and management class name (if present), to DFSMSdss, which interfaces with DFSMSdfp to select a volume to receive the data set.

To prevent SMS from selecting the source volume as the target volume for the recall, change the status attribute for the volume in the storage group. Suitable status attributes are DISABLENEW and QUIESCENEW.

Any data set that should be expired is expired without being migrated, during the conversion process.

4.3.2 Converting Level 1 DASD Volumes

The FREEVOL command allows you to empty an ML1 volume in preparation for new equipment or to replace old or partially damaged DASD volumes. It moves migration copies of SMS-managed data sets from an ML1 volume based on each data set′s management class attribute value.

You use the following command to empty a specific volume from your ML1 pool:

```
HSENDCMD FREEVOL MIGRATIONVOLUME(HSM14D) AGE(0)
```

HSM14D is the volser of the ML1 volume that we are going to replace.

The AGE(0) parameter, when entered for an ML1 volume, causes DFSMShsm to move the migration copies of all SMS-managed data sets, except those that are in need of backup, from the volume and place them on other ML1 or ML2 volumes, depending on the data set's management class. If the management class has a value specified for the LEVEL-1-DAYS-NONUSAGE and the age is met, the data set migrates to an ML2 volume. If the management class has a value specified for LEVEL-1-DAYS-NONUSAGE and the data set age does not meet the criterion or if the data set has an attribute of NOLIMIT, the data set migrates to another ML1 volume. You can restrict which ML1 DASD volumes receive data sets by using the DRAIN parameter of the ADDVOL command. DRAIN removes a volume from selection candidacy.

The data sets that are in need of backup (that were migrated but are awaiting automatic backup) and the backup versions that the BACKDS command created are not moved off the volume. When you are planning to remove an ML1 volume, you must first run AUTOBACKUP on the primary host, and then execute the FREEVOL AGE(0) command.

After emptying the ML1 volume, you should delete the volume, using the following command:

```
HSENDCMD DELVOL HSM14D MIGRATION
```

4.4 Reorganizing CDSs

The DFSMShsm CDSs require occasional reorganization. Detection of a need to reorganize can be based on the information received automatically through the ARC0909I message. This message is issued when the occupancy threshold requested in the SETSYS MONITOR command has been exceeded. You can obtain the current occupancy status by using the QUERY CONTROLDATASETS command. Status is reported in the ARC0148I message.

You do not have to reorganize the CDSs for performance reasons. The occurrence of control interval (CI) or control area (CA) splits in the CDSs does not affect DFSMShsm performance. Figure 66 on page 148 shows the anatomy of a CDS. For the MCDS, mid-section records are generated for each migrated data set or VSAM component. For the BCDS, mid-section records are generated for each backup copy.

Following is a summary of considerations for reorganizing the CDSs:

- Only reorganize them when you have to increase their size, or when moving the CDSs to another DASD device.
- DFSMShsm must be shut down in all hosts that have access to the CDSs, and no other job in any host should be using them during this operation.

- Allocate the CDS that you are going to reorganize with DISP=OLD (see Figure 67 on page 149, for the BCDS, MCDS, and OCDS DD cards).
- Reorganize the CDSs with FREESPACE(0) and let DFSMShsm split the mid-section intervals. Performance will be degraded for about two or three weeks during the CI/CA split process.
- If your BCDS or MCDS is *multicluster*, then you can reorganize the clusters containing mid-section records with FREESPACE(50).

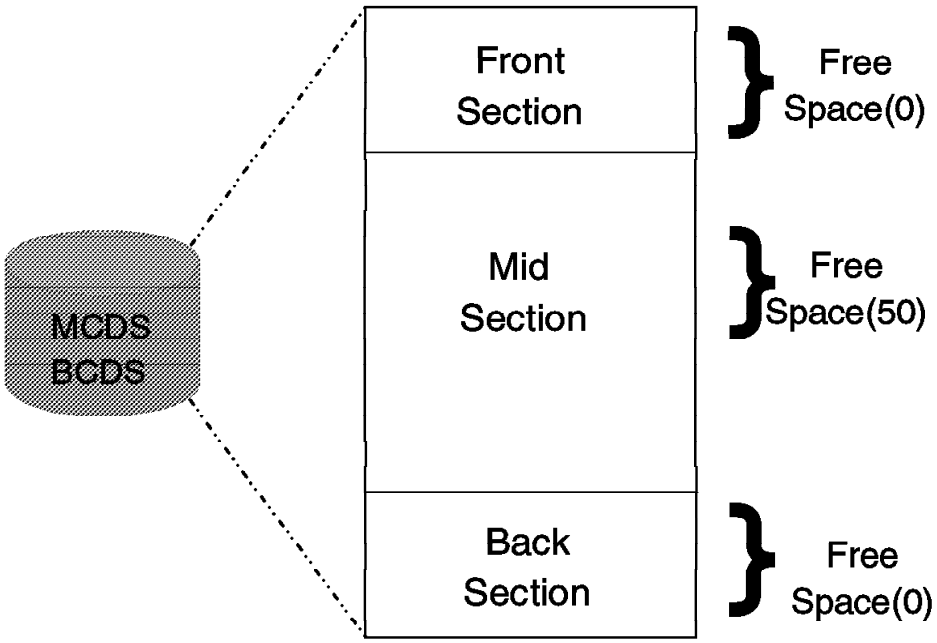


Figure 66. Anatomy of a CDS

DFSMShsm The mid-section of the CDS typically has more updates than the front section or the back section. Allocating the CDSs with free space probably causes a waste of space, as much of this free space will never be used because of the way DFSMShsm works.

The starter set job, HSMPRESS, in the HSM.SAMPLE.CNTL data set provides a sample JCL for the reorganization of the CDSs (see Figure 67).

```
//COMPRESS JOB ,'MHLRES4',MSGCLASS=X,CLASS=A
//*
//*****************************************************************/
//*  THIS SAMPLE JOB  COMPRESSES THE CONTROL DATA SETS.          */
//*                                                              */
//*  NOTE:  BEFORE RUNNING THIS JOB, YOU MUST SHUT DOWN DFSMSHSM.*/
//*****************************************************************/
//*
//ALLOCATE EXEC PGM=IEFBR14
//EXPMCDS DD DSN=HSM.EXPORT.MCDS,DISP=(,CATLG),
// UNIT=SYSDA,SPACE=(CYL,(20,2))
//EXPBCDS DD DSN=HSM.EXPORT.BCDS,DISP=(,CATLG),
// UNIT=SYSDA,SPACE=(CYL,(20,2))
//EXPOCDS DD DSN=HSM.EXPORT.OCDS,DISP=(,CATLG),
// UNIT=SYSDA,SPACE=(CYL,(20,2))
//*
//IDCAMS EXEC PGM=IDCAMS,REGION=512K
//MCDS DD DSN=HSM.MCDS,DISP=OLD
//BCDS DD DSN=HSM.BCDS,DISP=OLD
//OCDS DD DSN=HSM.OCDS,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 LISTCAT ENT(HSM.MCDS HSM.BCDS HSM.OCDS) ALL
 EXPORT HSM.MCDS ODS(HSM.EXPORT.MCDS) TEMPORARY
 IF LASTCC = 0 THEN -
        IMPORT IDS(HSM.EXPORT.MCDS) OFILE(MCDS) ERASE -
          CATALOG(UCAT.HSM)
 IF MAXCC = 0 THEN -
        DELETE HSM.EXPORT.MCDS NONVSAM
 EXPORT HSM.BCDS ODS(HSM.EXPORT.BCDS) TEMPORARY
 IF LASTCC = 0 THEN -
        IMPORT IDS(HSM.EXPORT.BCDS) OFILE(BCDS) ERASE -
          CATALOG(UCAT.HSM)
 IF MAXCC = 0 THEN -
        DELETE HSM.EXPORT.BCDS NONVSAM
 EXPORT HSM.OCDS ODS(HSM.EXPORT.OCDS) TEMPORARY
 IF LASTCC = 0 THEN -
        IMPORT IDS(HSM.EXPORT.OCDS) OFILE(OCDS) ERASE -
          CATALOG(UCAT.HSM)
 IF MAXCC = 0 THEN -
        DELETE HSM.EXPORT.OCDS NONVSAM
 LISTCAT ENT(HSM.MCDS HSM.BCDS HSM.OCDS) ALL
/*
```

Figure 67. Sample Job to Reorganize the DFSMShsm CDSs

4.5 Journal Data Set Maintenance

Occasionally, the DFSMShsm journal data set will become 100% full due to excessive DFSMShsm activity on the system. It is possible for the journal data set to reach 100% even after you have issued the BACKVOL CDS command. When this happens, DFSMShsm holds all functions that require having a record created, for example, BACKUP and MIGRATION, and then disables journal processing. As the journal approaches 100% full, you will see messages on the operator console (Figure 68 on page 150).

```
ARC0909E JOURNAL DATA SET IS ABOUT 097% FULL
ARC0909E JOURNAL DATA SET IS ABOUT 098% FULL
ARC0909E JOURNAL DATA SET IS ABOUT 099% FULL
IEC031I D37-04,IFG0554P,HSM54,HSM54,JOURNAL,2414,HSM14B,HSM.JRNL
IEA989I SLIP TRAP ID=XD37 MATCHED.  JOBNAME=HSM54   , ASID=0025.
ARC0020I ****************************
ARC0026E JOURNALING DISABLED DUE TO EOV ERROR ON 462
ARC0026E (CONT.) JOURNAL. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPEREPL,
ARC0026E (CONT.) RECYCLE, ARECOVER, AUDIT, AND EXPIREBV HELD.
ARC0020I ****************************
```

Figure 68. Console Messages for Journal Data Set at Capacity

To alleviate this situation, issue the following commands, which are based on a DFSMShsm started task name of HSM54:

1. F HSM54,SETSYS EMERGENCY
2. F HSM54,BACKVOL CONTROLDATASETS
3. F HSM54,SETSYS NOEMERGENCY
4. F HSM54,RELEASE ALL

These commands back up the CDSs to DASD or tape depending on your installation's setup. Once the backup has completed, the journal is cleared, and DFSMShsm can continue to use the journal data set. While the CDSs are being backed up, DFSMShsm queues any requests until the backup is complete.

If your journal data set fills frequently, it is undersized for the amount of DFSMShsm activity that it processes. To remedy this situation, increase the size of the journal data set:

1. F HSM54,SETSYS EMERGENCY
2. F HSM54,BACKVOL CONTROLDATASETS
3. F HSM54,STOP
4. Delete the current DFSMShsm journal data set
5. Allocate a new, larger DFSMShsm journal data set
6. S HSM54

4.6 Problem Determination

If you experience a problem with the operation of DFSMShsm, it may be necessary to gather some documentation to diagnois the problem.

In this section, we discuss the most important sources of problem determination documentation for DFSMShsm, including:

- The aid PDA trace facility
- Activity logs
- DFSMShsm log
- SVC dumps

4.6.1 PDA Facility

The PDA facility gathers DFSMShsm processing information and traces module flow and resource usage related to any DFSMShsm problem. DFSMShsm stores its trace information in the PDA log data sets.

DFSMShsm accumulates problem determination information at specific module points in the form of trace data and records this data in main storage. At predetermined intervals, the trace data is scheduled for output to DASD. The DFSMShsm trace recording function receives the trace data scheduled for output and writes this data to one of the two log data sets, ARCPDOX and ARCPDOY.

The PDA facility is automatically enabled at DFSMShsm startup. You enable or disable PDA processing with the SETSYS PDA(NONE|ON|OFF) command. We recommend that you leave PDA tracing enabled when DFSMShsm is active. The PDA trace options are:

- SETSYS PDA(NONE) specified at DFSMShsm startup causes no data to be gathered; the trace log data sets are not opened.
- SETSYS PDA(ON) causes DFSMShsm to request storage for data accumulation and the DASD trace log data sets to be opened if they have been allocated. If the trace log data sets have not been allocated, data is accumulated in internal storage only.
- SETSYS PDA(OFF) causes data accumulation to be stopped. The trace log data sets remain open.

DFSMShsm recognizes the trace log data sets by their DD names, ARCPDOX and ARCPDOY, in the DFSMShsm startup procedure. The trace log data set defined by the ARCPDOX DD statement is the data set that is written to. When the ARCPDOX trace log data set is filled, the two data set names are swapped, and recording continues on the newly renamed ARCPDOX trace log data set.

When the ARCPDOX trace log data set becomes full, and the trace log data set names are swapped, any previously recorded data in the current ARCPDOX data set is overlayed.

You need to copy the trace data in the ARCPDOY trace log data set to tape or DASD before the ARCPDOX data set becomes full in order to archive it. The PDA trace log data sets are automatically swapped when DFSMShsm is started. To switch the log data sets before the ARCPDOX data set is filled, issue the SWAPLOG PDA command. You can save the PDA trace data each time message ARC0037I is issued. This practice provides a sequential history of trace data over time so that the data is available when needed for resolving problems; it is similar to how you handle your SMF data.

4.6.1.1 Allocating the PDA Log Data Sets

Your PDA log data sets are usually allocated as part of the starter set see 2.2, “PROCLIB and PARMLIB Setup” on page 20). Figure 69 on page 152 contains sample JCL that can be used to allocate trace log data sets if the initial allocations are too small.

```
//MHLRES4J JOB  MSGLEVEL=1,CLASS=A
//STEP1    EXEC PGM=IEFBR14
//ARCPDOX  DD   DSN=HSM.HSMPDOX,DISP=(,CATLG),VOL=SER=HSM14C,
//         UNIT=3390,SPACE=(CYL,(30,2))
//ARCPDOY  DD DSN=HSM.HSMPDOY,DISP=(,CATLG),VOL=SER=HSM14C,
//         UNIT=3390,SPACE=(CYL,(30,2))
/*
```

Figure 69. Sample JCL to Allocate PDA Log Data Sets

4.6.1.2 Enabling and Switching the PDA Logs

If the PDA trace is not enabled at DFSMShsm startup, or you have disabled it, issue the following command from a system console to enable PDA tracing:

```
F DFSMSHSM,SETSYS PDA(ON)
```

To switch the PDA trace recording from one log data set to the other, issue the following command:

```
F DFSMSHSM,SWAPLOG PDA
```

4.6.1.3 Archiving PDA Data

A GDG can be used to archive the PDA trace data. You can then copy the GDS to tape or direct it to an SMS-managed DASD volume where it will be migrated to tape.

The following sample JCL can be used to create a GDG to archive the PDA trace data. You need to change the GDG name to correspond to the name you select for the starter set.

```
//MHLRES4G JOB  MSGLEVEL=1,MSGCLASS=A
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
  DEFINE GDG (NAME('HSM.HSMTRACE') LIMIT(30) SCRATCH)
/*
```

The following JCL is an example of copying the trace log data set to tape as a GDS for archive purposes. The JCL must be changed to corespond to the values you selected for your DFSMShsm starter set and your installation standards.

```
//MHLRES4C JOB  MSGLEVEL=1,MSGCLASS=A
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   DUMMY
//SYSUT1   DD   DSN=HSM.HSMPDOY,DISP=SHR
//SYSUT2   DD   DSN=HSM.HSMTRACE(+1),
//         UNIT=TAPE,
```

```
//          DISP=(NEW,CATLG,CATLG),VOL=(,,1),
//          DCB=(HSM.HSMPDOY)
```

When the PDA trace log switch occurs, DFSMShsm issues message ARC0037I. You can use an automation product to trap this message and automatically submit a copy job to archive the PDA trace data.

4.6.2 Activity Logs

The activity logs report on the backup, dump, migration, ABARS, and command processing of DFSMShsm in your system. DFSMShsm has five activity logs:

- The migration activity log provides information about space

  management activity including MIGRATE commands for volumes and levels, interval migration, and automatic primary and automatic secondary space management.
- The backup activity log provides information about automatic backup, volume command backup, and volume command recovery activities.
- The dump activity log provides information about automatic dump, command volume dump, and command volume restore activities.
- The ABARS activity log provides information about aggregate backup and recovery activities.
- The command activity log provides information about TAPECOPY and TAPEREPL activity and records error or informational messages that occur during low-level internal service processing.

With the SETSYS ACTLOGTYPE command you can direct DFSMShsm activity logs to DASD or to a SYSOUT class (see 2.2.3, “Setting Up the Base SETSYS Parameters” on page 25).

If you specify ACTLOGTYPE(DASD) on the SETSYS command, DFSMShsm dynamically allocates DASD data sets for the activity logs with the names in the following format:

HSMACT.Hmcvthost.function.agname.Dyyddd.Thhmmss

where:

Hmcvthost	DFSMShsm host ID from the DFSMShsm startup PROC statement, preceded by H
function	Either ABACKUP, ARECOVER, BAKLOG, CMDLOG, DMPLOG, or MIGLOG
agname	Aggregate group name (only present if function is ABACKUP or ARECOVER)
Dyyddd	Year and day of allocation, preceded by D
Thhmmss	Hour, minute, second of allocation, preceded by T.

Several methods can be used to check the activity logs, such as browsing or printing or processing them with a CLIST.

Figure 70 on page 154 is a sample of the information in the DFSMShsm backup activity log.

```
 DFSMSHSM BACKUP LOG, TIME 16:52:12,  DATE 98/07/29
ARC0720I AUTOMATIC BACKUP STARTING
ARC0740I CDS BACKUP STARTING AT 17:00:00 ON 1998/07/29, SYSTEM SC54, TO DASD IN PARALLEL MODE, DATAMOVER=DSS
ARC0742I BACKUP FOR MCDS STARTING AT 17:00:01 ON 1998/07/29, BACKUP COPY TECHNIQUE IS STANDARD
ARC0742I BACKUP FOR BCDS STARTING AT 17:00:01 ON 1998/07/29, BACKUP COPY TECHNIQUE IS STANDARD
ARC0742I BACKUP FOR OCDS STARTING AT 17:00:01 ON 1998/07/29, BACKUP COPY TECHNIQUE IS STANDARD
ARC0750I BACKUP FOR JRNL STARTING AT 17:00:01, ON 1998/07/29
ARC0743I JRNL SUCCESSFULLY BACKED UP TO HSM.JRNL.BACKUP.V0000003, ON VOLUME(S) HSM14C, TIME=17:00:01, DATE=1998/07/29
PAGE 0001     5695-DF175  DFSMSDSS V1R4.0  DATA SET SERVICES     1998.210 17:00
 DUMP DATASET(INCLUDE( -
 HSM.OCDS  )) -
 SPHERE -
 OUTDDNAME( -
 SYS00013  ) -
 OPTIMIZE(3) -
 COMPRESS
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ADR109I (R/I)-RI01 (01), 1998.210 17:00:01 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-SETUP(01), 1998.210 17:00:01 EXECUTION BEGINS
                .
                .

ARC0722I BACKUP STARTING ON VOLUME HSM14A(SMS) AT 17:30:18 ON 1998/07/29 SYSTEM SC54
ARC0728I VTOC FOR VOLUME HSM14C COPIED TO DATA SET HSM.VTOC.T183017.VHSM14C.D98210 ON VOLUME HSM14D
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=  6840 RC=   0, REASON=    0, AGE=  72,
 DSN=OMVS54.ADSM.STORAGE.POOL001
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=  2145 RC=   0, REASON=    0, AGE=  72,
 DSN=OMVS54.ADSM.STORAGE.POOL002
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     1 RC=   0, REASON=    0, AGE=  72, DSN=OMVS54.ADSM.DISKLOG
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     7 RC=   0, REASON=    0, AGE=  70, DSN=OMVS54.ANF.ROUTING
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     7 RC=   0, REASON=    0, AGE=  70, DSN=OMVS54.ANF.OPTIONS
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     7 RC=   0, REASON=    0, AGE=  70, DSN=OMVS54.ANF.QUEUE
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=    30 RC=   0, REASON=    0, AGE=  70, DSN=OMVS54.ANF.MSGFILE
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     0 RC=   0, REASON=    0, AGE=  70,
 DSN=OMVS54.ANFISPF.TRACE
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     1 RC=   0, REASON=    0, AGE=  62,
 DSN=OMVS54.TCPIPOE.PROFILE
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     1 RC=   0, REASON=    0, AGE=  62,
 DSN=OMVS54.TCPIPMVS.PROFILE
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     1 RC=   0, REASON=    0, AGE=  59,
 DSN=OMVS54.TCPIPMVS.ROUTED.GATEWAYS
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     1 RC=   0, REASON=    0, AGE=  58,
 DSN=OMVS54.TCPIPOE.ROUTED.PROFILE
ARC0734I ACTION=BACK-UP FRVOL=TOT54A TOVOL=M195AW TRACKS=     1 RC=   0, REASON=    0, AGE=  58,
 DSN=OMVS54.TCPIPOE.ROUTED.GATEWAYS
ARC0723I BACKUP ENDING ON VOLUME TOT54A AT 17:33:38, 35 DATA SETS BACKED UP
                .
                .

ARC0721I AUTOMATIC BACKUP ENDING
 DFSMSHSM BACKUP LOG, TIME 17:57:46,  DATE 98/07/29
```

Figure 70. Sample Backup Activity Log Data

4.6.3 DFSMShsm Log Data Sets

The DFSMShsm log data sets provide information about events on a particular processing unit and about commands entered with the LOG command. The DFSMShsm log records this information in chronological order.

Two physical sequential data sets, are identified to DFSMShsm by the ARCLOGX and ARCLOGY DD statements in the startup procedure. DFSMShsm records information in the ARCLOGX data set until the ARCLOGX data set is full. Then DFSMShsm swaps the ARCLOGX data set with the ARCLOGY data set, exchanges the names, and issues a message to inform the operator that DFSMShsm has swapped the log data sets.

To automatically swap the DFSMShsm log data sets at startup, specify LOGSW=YES on the PROC statement of your DFSMShsm startup procedure. You can also swap the DFSMShsm log data sets with the following SWAPLOG command:

```
F DFSMSHSM,SWAPLOG
```

Once the log is switched, you will see the following message on the system console:

```
ARC0020I ****************************
ARC0027I  DFSMSHSM LOG SWITCHED
ARC0020I ****************************
```

4.6.3.1 Printing the DFSMShsm Logs

After the SWAPLOG has completed, you can process the HSMLOGY data set. If the log data in HSMLOGY is not processed before the next log swapping, its contents will be lost.

You can run the HSMLOG and HSMEDIT jobs to process the HSMLOGY data set. The HSMLOG and HSMEDIT jobs are in the HSM.SAMPLE.CNTL data set.

The HSMLOG procedure runs the ARCPRLOG program to perform some initial formatting of the information in the HSMLOGY data set and prints the formatted log. The ARCPEDIT program of the EDITLOG procedure takes the formatted output from the ARCPRLOG program as input. This data is in the EDITLOG data set. The ARCPEDIT program further formats the data, and the information can be printed. The following sample JCL invokes the ARCPRLOG program:

```
//HSMLOG   JOB ,'MHLRES4',MSGCLASS=X,CLASS=A
//PRINTLOG EXEC PGM=ARCPRLOG
//ARCPRINT DD SYSOUT=*
//ARCLOG DD DSN=HSM.HSMLOGY1,DISP=OLD
//ARCEDIT DD DSN=HSM.EDITLOG,DISP=OLD
//*
//EMPTYLOG EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=HSM.HSMLOGY1,DISP=OLD
//SYSUT1 DD DUMMY,DCB=(HSM.HSMLOGY1)
/*
```

The following sample JCL invokes the ARCPEDIT program and prints the edited log information:

```
//EDITLOG  JOB ,'MHLRES4',MSGCLASS=X,CLASS=A
//EDITLOG EXEC PGM=ARCPEDIT
//ARCPRINT DD SYSOUT=*
//ARCLOG DD DSN=HSM.EDITLOG,DISP=SHR
/*
```

4.6.3.2 Disabling Logging

You can prevent DFSMShsm from making entries in the DFSMShsm logs. Even though the log entries are prevented, DFSMShsm continues to make entries in the activity logs. To prevent entries in the DFSMShsm log, specify the HOLD LOG command in the ARCMDxx member of PARMLIB.

Most of the information presented in the DFSMShsm log data sets is also present in the activity logs, in the PDA trace log data sets, and in SMF records.

The only data that is present only in the DFSMShsm logs and not in the other logs is the management work element (MWE) information, which is used by various independent software vendor (ISV) products. We suggest you hold DFSMShsm logging and use the activity and PDA trace logs instead. You may have some performance benefits.

4.6.4 SVC Dumps

In the event of a DFSMShsm abend, OS/390 Dump Services may request a dump to be written to a preallocated SYS1.DUMPxx data set or to a dynamically allocated dump data set named according to your installation standards. This dump may contain some helpful information to diagnose an error in DFSMShsm processing.

We recommend that you specify the SETSYS SYS1DUMP command in the ARCCMDxx of member PARMLIB. For error analysis, the IBM Support Center may request the SVC dump produced at the time DFSMShsm abended.

To generate an SVC dump, enter the DUMP command at the system console:

```
DUMP COMM=(text)
```

where the text you enter is the title you want the dump to have. You are then prompted with an IEE094D message, which asks you to specify the operands for the DUMP command. We suggest you reply to the IEE094D message with these parameters:

```
R id,JOBNAME=(HSM54),SDATA=(LSQA,ALLNUC,NUC,CSA,LPA,PSA,SQA,
RGN,SUM,SWA,GRSQ,TRT),END
```

where HSM54 is the name of the started task procedure for DFSMShsm in our primary processor. You can contact the IBM Support Center for DFSMShsm dump analysis. They may request that you also provide an unformatted copy of the PDA trace data sets leading up to the error and an unformatted copy of the DFSMShsm logs for about 30 minutes leading up to the hang or loop, or the activity logs.

Figure 71 contains sample JCL to copy an SVC dump data set to tape.

```
//DMPCOPY  JOB  MSGCLASS=X,CLASS=A
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
//SYSUT1   DD   DSN=SYS1.DUMP00,DISP=SHR
//SYSUT2   DD   DSN=DUMP.OUTPUT,DISP=(,KEEP),
//         UNIT=tape,VOL=SER=volser
/*
```

Figure 71. Sample SVC Dump Copy Job

4.6.5 Dump Analysis Elimination

Dump analysis elimination (DAE) is an MVS function that eliminates duplicate dumps in MVS systems or across MVS systems.

DFSMShsm generates dumps to gather first-failure diagnostic information. Problems that occur multiple times, perhaps on different hosts, typically generate a storage dump. The initial dump is helpful for diagnosing the problem; additional dumps for the same problem usually are not needed.

Controlling DFSMShsm storage dumps is a major issue with users. The ability to properly suppress duplicate dumps is an important step toward successfully managing dump data sets. DFSMShsm supplies additional information to the DAE function to build a symptom string to be used to suppress duplicate dumps for functions that run in the same or different systems.

4.6.5.1 DAE Implementation

To implement DAE for DFSMShsm, the following tasks must be completed:

- SETSYS SYS1DUMP must be specified. SYS1DUMP is the DFSMShsm default.
- PARMLIB member ADYSETxx must be coded with the SUPPRESSALL keyword, for example:

```
DAE=START,RECORDS(400),
    SYSMDUMP(MATCH,UPDATE),
    SVCDUMP(MATCH,UPDATE,SUPPRESSALL)
```

- MVS/ESA SP V4.3 introduced the ability to use DAE in a multihost environment, thereby allowing a single DAE data set to be shared across systems in a sysplex. The coupling services of the cross-system coupling facility (XCF) and GRS must be enabled for the DAE data set to be shared in a sysplex environment and for dumps to be suppressed across MVS systems.

  Only those hosts with DFSMShsm V1R4 can use the DAE suppression for dumps for DFSMShsm. DFSMShsm V1R4 uses DAE for functions that run in both the primary address space and the ABARS secondary address space.

DAE does not suppress SYSABEND, SYSUDUMP, SYSMDUMP, or SNAP dumps or dumps that originate from SLIP or DUMP operator commands. Because these dumps are taken only on demand, suppression is not desirable.

This support does not apply to dumps produced by DFSMShsm as a result of the TRAP command.

4.7 Auditing DFSMShsm

The AUDIT command detects, reports, diagnoses, and often provides repair s for discrepancies between CDSs, catalogs, and DFSMShsm-owned volumes.

To ensure data integrity, DFSMShsm uses numerous data set records to track individual data sets. These records are contained in the:

- Master catalog, which is a list of data sets for the entire system
- User catalog, which is a list of data sets accessible from that catalog
- Journal, which keeps a running record of backup and migration transactions
- SDSP data sets on migration volumes
- MCDS, which is an inventory of migrated data sets and migration volumes
- BCDS, which is an inventory of backed up data sets and volumes, dump ed volumes, and backed up aggregates
- OCDS, which contains a tape table of contents (TTOC) inventory of migration and backup tape volumes

In normal operation, these records stay in synchronization. However, because of data errors, hardware failures, or human errors, it is possible for these records to become unsynchronized. The AUDIT command allows the system to cross-check the various records concerning data sets and DFSMShsm resources. AUDIT can list errors and propose diagnostic actions or, at your option, complete most repairs itself.

Consider using the AUDIT command for the following reasons:

- After any CDS restore (highly recommended)
- After an ARC184I message (error when reading or writing DFSMShsm CDS records)
- Errors on the RECALL or DELETE of migrated data sets
- Errors on BDELETE or RECOVER of backup data sets
- DFSMShsm tape-selection problems
- RACF authorization failures
- Power or hardware failure
- Periodic checks

You can use AUDIT to cross-check the following sources of control information:

- MCDS or individual migration data set records
- BCDS or individual backup data set records or ABARS records
- OCDS or individual DFSMShsm-owned tapes
- DFSMShsm-owned DASD volumes
- Migration-volume records
- Backup-volume records
- Recoverable-volume records (from dump or incremental backup)
- Contents of SDSP data sets

It is best to use AUDIT at times of low system activity, as some audit processes can run for quite some time. However, the AUDIT process can b e used almost any time.

The following example shows how to instruct DFSMShsm to audit the BCDS an d fix any errors:

```
HSENDCMD AUDIT BCDS FIX
```

Refer to the *DFSMShsm Storage Administration Reference*, which provides examples of coding the command and guidance on interpreting its output.

Chapter 5. Monitoring

Monitoring DFSMShsm in real time or historically can be achieved in a number of ways. DFSMShsm commands provide a comprehensive view of what is happening, what has happened, and what should happen. The DFSMS Optimizer HSM Monitor/Tuner provides a graphical user interface (GUI) to DFSMShsm that can be used to monitor DFSMShsm system activity, issue commands, and cancel tasks.

In this chapter we explain how to use the DFSMShsm commands and the DFSMS Optimizer HSM Monitor/Tuner to collect information about DFSMShsm processing and monitor DFSMShsm activity.

Each of these commands gathers information from different sources within DFSMShsm, such as the MCDS, BCDS, OCDS, functional statistics records (FSRs), or the DFSMShsm address space.

5.1 List Command

The DFSMShsm LIST command obtains its information from the CDSs: MCDS, BCDS and OCDS. You can list the following categories of information:

- ABARS activity
- Backup volumes
- Data sets
- Dump classes
- Dump volumes
- Host information
- Migration information
- Primary volume information
- Tape volume information
- User authorization

Many options are available for each of the LIST commands; we do not show each and every variation. The commands that we show cover each area of DFSMShsm for which LIST gathers information. Some of the commands use only the MCDS or BCDS to gather information, whereas others use both the MCDS and BCDS. You may want to investigate the commands further to determine whether there are additional parameters that suit your installation requirements.

1. **Who can issue the LIST command?**

   The LIST command is issued primarily by a DFSMShsm-authorized storage administrator who will use the DFSMShsm HSENDCMD command from a TSO terminal. In the examples that follow we use the abbreviated command, HSEND, to issue our list commands.

2. **Some of the parameters for DFSMShsm are quite lengthy; do I have to specify the long name?**

   No. In the examples that follow we show some of the more common command parameter abbreviations.

3. **If I issue the LIST command, where will my output be directed?**

   The information that is generated from the LIST command can be returned in a number of ways:

   - Written to a terminal
   - As a printed listing
   - As a DASD data set

   The LIST command is capable of creating a large amount of output, so be sure to direct any output that may be large, for example, a list of all migration volumes, to an output data set.

4. **Which parameters do I specify on the LIST command to direct output to each of the above?**

   If you want to direct the output to a terminal, use:

```
LIST request TERMINAL
```

   If you want to obtain a printed listing, use the following command to direct the output to your SYSOUT class associated with print:

```
LIST request SYSOUT(class)
```

   If you want to send the output to a DASD data set, use the following command:

```
LIST request ODS(dsname)
```

   When directing the output to DASD, you must specify the fully qualified data set name. If the output data set does not exist, DFSMShsm dynamically allocates it. If the specified output data set exists, DFSMShsm appends the output to the end of the data set.

   In all of our examples, the output is directed to a DASD data set.

5. **How can I list information regarding my ABARS aggregate groups and any ABACKUP and ARECOVER information?**

   In the following example we list out the ABR record which is created as a result of performing an ABACKUP. Comprehensive information about the ABARS activity that has taken place is recorded within it. In this example we have chosen to list our aggregate group PAY1 and version number 1.

```
HSEND LIST AGGREGATE(PAY1) VERSION(0001) ODS(MHLRES5.ABR)
```

   The output that is written to data set looks like this:

```
 DFSMSHSM CONTROL DATASET AGGREGATE BACKUP AND RECOVERY VERSION LISTING -----
       AT 14:12:08 ON 1998/07/30 FOR SYSTEM=SC54

 ABR RECORD KEY = PAY1.1998210000101
 AGGREGATE GROUP NAME = PAY1                              DFSMSHSM VERSION = V1.4.0
 AGGREGATE ACCOUNT CODE =
 ABACKUP DATE = 1998/07/29     ABACKUP TIME = 13:32:33
 VERSION = 0001                COPY NUMBER = 01
 UNIT NAME = 3490              NUMBER OF COPIES = 02
 SOURCE SYSTEM = SC54
 MANAGEMENT CLASS = MC54NMIG
 REMOTE DESTINATIONS =         DESTINATION ID                TRANS. OK
                               NONE

 CONTROL FILE NAME = ABARS.PAY1.C.C01V0001
 DFSMSDSS DATA FILE NAME = ABARS.PAY1.D.C01V0001
 INTERNAL I/O DATA FILE NAME = ABARS.PAY1.O.C01V0001
 ACTIVITY LOG/INSTRUCTION FILE NAME = ABARS.PAY1.I.C01V0001

 ABACKUP ACTIVITY LOG DATA SET NAME = HSMACT.H1.ABACKUP.PAY1.D98210.T133233
 INSTRUCTION DATA SET NAME = MHLRES5.ABARS.PAY1.INSTRUCT


 FILTER OUTPUT DATA SET NAME = MHLRES5.ABARS

 CONTROL FILE VOLSERS =
  VOLS=  M195AX
  LIBS=  *NO_LIB*
 DFSMSDSS DATA FILE VOLSERS =
  VOLS=  M195AX
  LIBS=  *NO_LIB*
 INTERNAL I/O DATA FILE VOLSERS =
  VOLS=  M195AX
  LIBS=  *NO_LIB*
 INSTRUCTION/ACTIVITY LOG FILE VOLSERS =
  VOLS=  M195AX
  LIBS=  *NO_LIB*
 TAPE EXPIRATION DATE =

 NUMBER OF USER TAPES = 0000
 NUMBER OF ACCOMPANY TAPES = 0000

 STORAGE USED AT ABACKUP SITE :
    L0             ML1             ML2               TOTAL
    0000000011K    0000000000K     0000000000K       0000000011K

 ABACKUP ELAPSED TIME = 00:00:00.36
 TAPE STACKING = YES

AARECOVER INFORMATION:
----------------------

 DATE            TIME           RETCODE    OPTION        XMIT=Y/N
 1998/07/29      15:24:31       0000       EXECUTE       N

 STORAGE REQUIREMENTS :
    L0             ML1             ML2               TOTAL
    0000000000K    0000000000K     0000000000K       0000000000K
```

```
ARECOVER ELAPSED TIME = 00:00:00.17
ARECOVER ACTIVITY LOG DATA SET NAME = HSMACT.H1.ARECOVER.PAY1.D98210.T152431
CONFLICT RESOLUTION DATA SET NAME = NONE
RESTART DATA SET NAME = NONE
ABR RECORD KEY = PAY1.1998210000102
AGGREGATE GROUP NAME = PAY1                              DFSMSHSM VERSION = V1.4.0
AGGREGATE ACCOUNT CODE =
ABACKUP DATE = 1998/07/29     ABACKUP TIME = 13:32:33
VERSION = 0001                COPY NUMBER = 02
UNIT NAME = 3490              NUMBER OF COPIES = 02
SOURCE SYSTEM = SC54
MANAGEMENT CLASS = MC54NMIG
REMOTE DESTINATIONS =         DESTINATION ID                TRANS. OK
                              NONE

CONTROL FILE NAME = ABARS.PAY1.C.C02V0001
DFSMSDSS DATA FILE NAME = ABARS.PAY1.D.C02V0001
INTERNAL I/O DATA FILE NAME = ABARS.PAY1.O.C02V0001
ACTIVITY LOG/INSTRUCTION FILE NAME = ABARS.PAY1.I.C02V0001

ABACKUP ACTIVITY LOG DATA SET NAME = HSMACT.H1.ABACKUP.PAY1.D98210.T133233
INSTRUCTION DATA SET NAME = MHLRES5.ABARS.PAY1.INSTRUCT
FILTER OUTPUT DATA SET NAME = MHLRES5.ABARS

CONTROL FILE VOLSERS =
 VOLS=  M195AV
 LIBS=  *NO_LIB*
DFSMSDSS DATA FILE VOLSERS =
 VOLS=  M195AV
 LIBS=  *NO_LIB*
INTERNAL I/O DATA FILE VOLSERS =
 VOLS=  M195AV
 LIBS=  *NO_LIB*
INSTRUCTION/ACTIVITY LOG FILE VOLSERS =
 VOLS=  M195AV
 LIBS=  *NO_LIB*
TAPE EXPIRATION DATE =



NUMBER OF USER TAPES = 0000
NUMBER OF ACCOMPANY TAPES = 0000

STORAGE USED AT ABACKUP SITE :
   L0              ML1             ML2               TOTAL
   0000000011K     0000000000K     0000000000K       0000000011K

ABACKUP ELAPSED TIME = 00:00:00.36
TAPE STACKING = YES

ARECOVER INFORMATION: NONE

----- END OF - AGGREGATE BACKUP AND RECOVERY VERSION - LISTING -----
```

6. **How can I list all of the backup volumes that I have created through DFSMShsm?**

   If you issue the following command, you get a list of all backup volumes that exist and are known in your installation:

```
HSEND LIST BACKUPVOLUME ODS(MHLRES5.LBV)
```

   The output that is written to the data set looks like this:

```
- DFSMSHSM CONTROL DATASET - BACKUP VOLUME-- LISTING ----- AT 13:54:19  ON 98/07/30 FOR SYSTEM=SC54

 VOLSER   DEVICE  BACKUP   VOL    TOTAL    FREE   THRESH   LAST BACKUP    PSWD   EXP   RACF   EMPTY  IDRC  DUPLEX
           TYPE    TYPE    FULL   TRACKS  TRACKS              DATE                                          ALT

 M195AW    3490   DAILY     NO    ******  ******    ***      98/07/29      NO    NO     YES    NO     Y    M195A1
 M195A2    3490   DAILY     NO    ******  ******    ***      98/07/29      NO    NO     YES    NO     Y    M195AT
 M195A3    3490   DAILY     NO    ******  ******    ***      98/07/29      NO    NO     YES    NO     Y    M195A1
 ----- END OF - BACKUP VOLUME - LISTING -----
```

7. **How do I limit the output to a particular level of data set name, volume, and age?**

In the following example we show how it is possible to specify the data set name qualifiers, the volser of the volume, and the age of the data sets.

In the following command we ask for data sets that have a high-level qualifier of MHLRES5, and a second-level qualifier of PAY1, are migrated to ML1 volume HSM14D, and have not been referenced for one day.

```
HSEND LIST LEVEL(MHLRES5.PAY1) MCDS SELECT(VOLUME(HSM14D) AGE(1))
```

The output that is written to a DASD data set looks like this:

```
- DFSMSHSM CONTROL DATASET - MIGRATED DATASET-- LISTING ----- AT 14:22: 48 ON 98/07/30 FOR SYSTEM=SC54

 DATASET NAME                              MIGRATED  LAST REF MIGRATED  TRKS   QTY   TIMES  DS SDSP   QTY    LAST MIG
                                           ON VOLUME   DATE     DATE   ALLOC 2K BLKS  MIG  ORG  DS  16K BLKS  VOLUME

 MHLRES5.PAY1.SALARY1                        HSM14D  98/07/29 98/07/30 000001 0000001 001   PS  NO   ******   ******
- DFSMSHSM CONTROL DATASET - SUMMARY-- LISTING ----- AT 14:22:48 ON 98/ 07/30 FOR SYSTEM=SC54




                                          MIGRATED       TRACKS      K-BYTES
                                          DATA SETS     MIGRATED     MIGRATED

                                        0000000001      000000001     00000001

 ----- END OF - MIGRATED DATASET - LISTING -----
```

8. **If I want to list all of the data sets that have been either migrated or recalled, which command do I use?**

   If you want to obtain a list of all the data sets that have an entry in the MCDS, even though they may already have been recalled to a primary volume, issue the following command:

```
HSEND LIST DATASETNAME MCDS INCLUDEPRIMARY ODS(MHLRES5.DSN)
```

The output from this command when directed to an output data set looks like this:

```
- DFSMSHSM CONTROL DATASET - MIGRATED DATASET-- LISTING ----- AT 15:5 1:56 ON 98/07/30 FOR SYSTEM=SC54

 DATASET NAME                              MIGRATED  LAST REF MIGRATED  TRKS   QTY   TIMES  DS SDSP   QTY    LAST MIG
                                           ON VOLUME   DATE     DATE   ALLOC 2K BLKS  MIG  ORG  DS  16K BLKS  VOLUME

 MHLRES5.PAY1.SALARYA                        HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY1                        HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY2                        HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY3                        HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY4                        HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY5                        HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY6                        ONLINE  98/07/30 98/07/30 000001 0000000 001   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY7                        ONLINE  98/07/30 98/07/30 000001 0000000 001   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY8                        ONLINE  98/07/30 98/07/30 000001 0000000 001   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY9                        ONLINE  98/07/30 98/07/30 000001 0000000 001   PS  NO   ******   ******
- DFSMSHSM CONTROL DATASET - SUMMARY-- LISTING ----- AT 15:51:56 ON 9 8/07/30 FOR SYSTEM=SC54

                                          MIGRATED       TRACKS      K-BYTES
                                          DATA SETS     MIGRATED     MIGRATED

                                        0000000006      000000006     00000004

 ----- END OF - MIGRATED DATASET - LISTING -----
```

9. **Can I list information recorded in the BCDS about data sets that have been backed up and use the LEVEL parameter to restrict the amount of output?**

   If you were to use this command, you would list out all data sets that have the high-level qualifier HSMACT and have an entry in the BCDS:

```
HSEND LIST LEVEL(HSMACT) BCDS ODS(MHLRES5.DSN3)
```

The output (partial) that is written to a DASD data set looks like this:

```
- DFSMSHSM CONTROL DATASET - BACKUP DATASET-- LISTING ----- AT 15:56:49  ON 98/07/30 FOR SYSTEM=SC54


DSNAME = HSMACT.H1.BAKLOG.D98204.T183225          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.V385717.HSMACT.H1.I8210          M195A2  HSM14C  98/07/29  17:57:38  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98208.T182753          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.T365717.HSMACT.H1.I8210          M195A3  HSM14A  98/07/29  17:57:35  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98208.T200851          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.U365717.HSMACT.H1.I8210          M195A2  HSM14C  98/07/29  17:57:36  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98209.T104716          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.T385717.HSMACT.H1.I8210          M195A2  HSM14C  98/07/29  17:57:37  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98209.T104822          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.U513317.HSMACT.H1.I8210          M195AW  HSM14B  98/07/29  17:33:51  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98209.T105715          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.W385717.HSMACT.H1.I8210          M195A3  HSM14A  98/07/29  17:57:38  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98209.T132129          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.U405717.HSMACT.H1.I8210          M195A2  HSM14C  98/07/29  17:57:40  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98209.T134803          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.U415717.HSMACT.H1.I8210          M195A2  HSM14C  98/07/29  17:57:41  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98210.T010044          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.T563317.HSMACT.H1.I8210          M195AW  HSM14B  98/07/29  17:33:55  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98210.T154052          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.V415717.HSMACT.H1.I8210          M195A3  HSM14A  98/07/29  17:57:41  YES  000   001    NO   NO    NO


DSNAME = HSMACT.H1.BAKLOG.D98210.T164542          BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME              BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                              VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF
```

```
    HSM.BACK.T583317.HSMACT.H1.I8210             M195AW  HSM14B  98/07/29  17:33:58  YES  000   001    NO   NO    NO

DSNAME = HSMACT.H1.CMDLOG.D98204.T183225              BACKUP FREQ = ***, MAX BACKUP VERSIONS = ***

    BACKUP VERSION DATA SET NAME                 BACKUP   FROM    BACKUP    BACKUP   SYS  GEN   VER   UNS/ RACF  BACKUP
                                                 VOLUME  VOLUME    DATE      TIME    CAT  NMBR  NMBR  RET  IND   PROF

    HSM.BACK.U345717.HSMACT.H1.I8210             M195A3  HSM14A  98/07/29  17:45:35  YES  000   001    NO   NO    NO


----- END OF - BACKUP DATASET - LISTING -----
```

10. **Can I get a list of the ML1 volumes to which data sets with a high-level qualifier of MHLRES5 are migrated?**

    If you use the following command, you will see to which ML1 volumes your data sets have been migrated:

```
HSEND LIST LEVEL(MHLRES5) MCDS SELECT(ML1) ODS(MHLRES5.DSN4)
```

    The output that is written to a DASD data set looks like this:

```
- DFSMSHSM CONTROL DATASET - MIGRATED DATASET-- LISTING ----- AT 16:21: 54 ON 98/07/30 FOR SYSTEM=SC54

 DATASET NAME                                MIGRATED  LAST REF MIGRATED  TRKS   QTY   TIMES  DS SDSP   QTY    LAST MIG
                                             ON VOLUME   DATE     DATE   ALLOC 2K BLKS  MIG  ORG  DS  16K BLKS  VOLUME

 MHLRES5.PAY1.SALARYA                          HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY1                          HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY2                          HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY3                          HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY4                          HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
 MHLRES5.PAY1.SALARY5                          HSM14D  98/07/30 98/07/30 000001 0000001 002   PS  NO   ******   ******
- DFSMSHSM CONTROL DATASET - SUMMARY-- LISTING ----- AT 16:21:54 ON 98/ 07/30 FOR SYSTEM=SC54



                                          MIGRATED       TRACKS      K-BYTES
                                          DATA SETS     MIGRATED     MIGRATED

                                       0000000006      000000006     00000004

 ----- END OF - MIGRATED DATASET - LISTING -----
```

11. **How can I list the contents of a primary volume at the time the last incremental backup was taken?**

    This information is taken from the backup VTOC copy data sets. The following command allows you to list out the contents of the latest incremental backup for primary volume HSM14A:

```
HSEND LIST PVOL(HSM14A) BCDS BACKUPCONTENTS ODS(MHLRES5.DSN6)
```

    The output that is written to the data set looks like this:

```
-- DFSMSHSM CONTROL DATASET -PRIMARY VOLUME-BCDS-- BCONTENTS    --- AT  16:42:54 ON 98/07/30 FOR SYSTEM=SC54
 CONTENTS OF BACKUP VTOC COPY # 00 FOR PRIMARY VOLUME HSM14A
DATASET NAME                                 ORG  MULTI  CREATED   REF ERENCED  EXP DATE  RACF  PSWD   CHANGED
 HSM.BCDS                                     VS    ***   98/07/23  98/07/29    00/00/00  ***   ***    YES
 HSM.BCDS.DATA                                VS    ***   98/07/23  98/07/29    00/00/00  ***   ***    YES
 HSM.BCDS.INDEX                               VS    ***   98/07/23  00/00/00    00/00/00  ***   ***    NO
 HSM.MCDS                                     VS    ***   98/07/23  98/07/29    00/00/00  ***   ***    YES
 HSM.MCDS.DATA                                VS    ***   98/07/23  98/07/29    00/00/00  ***   ***    YES
 HSM.MCDS.INDEX                               VS    ***   98/07/23  00/00/00    00/00/00  ***   ***    NO
 HSM.OCDS                                     VS    ***   98/07/23  98/07/29    00/00/00  ***   ***    YES
 HSM.OCDS.DATA                                VS    ***   98/07/23  98/07/29    00/00/00  ***   ***    YES
 HSM.OCDS.INDEX                               VS    ***   98/07/23  00/00/00    00/00/00  ***   ***    NO
 HSMACT.H1.BAKLOG.D98208.T182753              PS    NO    98/07/27  98/07/27    00/00/00  NO    NO     YES
 HSMACT.H1.BAKLOG.D98209.T105715              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
 HSMACT.H1.BAKLOG.D98210.T154052              PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     YES
 HSMACT.H1.BAKLOG.D98210.T165212              PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98204.T183225              PS    NO    98/07/23  98/07/23    00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98208.T182753              PS    NO    98/07/27  98/07/27    00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98208.T200850              PS    NO    98/07/27  98/07/27    00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98209.T104822              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98209.T105715              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
```

```
HSMACT.H1.CMDLOG.D98209.T134759              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
HSMACT.H1.CMDLOG.D98210.T165212              PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     YES
HSMACT.H1.DMPLOG.D98204.T183225              PS    NO    98/07/23  98/07/23    00/00/00  NO    NO     YES
HSMACT.H1.DMPLOG.D98209.T104822              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
HSMACT.H1.MIGLOG.D98209.T105715              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
HSMACT.H1.MIGLOG.D98209.T132129              PS    NO    98/07/28  98/07/28    00/00/00  NO    NO     YES
MHLTST4.COMPINFO                             PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     NO
MHLTST4.C2SYSBS                              PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     NO
MHLTST4.C3RECYCL                             PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     NO
MHLTST4.SC47.SPFLOG1.LIST                    PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     NO
MHLTST4.SYSOUT                               PS    NO    98/07/29  98/07/29    00/00/00  NO    NO     NO
----- END OF - PRIMARY VOLUME - LISTING -----
```

12. **How do I list out the attributes of a dump class?**

    If you do not know the name of the dump class and want to list all dump classes, then omit the dump class name in the following command. If you know the name of the dump class, you can issue the following command. In this example the dump class name is SYSPLEX2:

```
HSEND LIST DUMPCLASS(SYSPLEX2) ODS(MHLRES3.DSN1)
```

    The output that is written to the DASD data set looks like this:

```
--- DFSMSHSM CONTROL DATASET -DUMP CLASS-BCDS--- LISTING --- AT 17:48:3 2 ON 98/07/30 FOR SYSTEM=SC47

DUMP      UNIT     AUTO   DATASET  RESET   CLASS                          TAPE   VTOC
CLASS     TYPE     REUSE  RESTORE  CHANGE  DISABLE  DAY  FREQ   RETPD     EXPDT  COPIES  STACK  DISPOSITION
SYSPLEX2  3490      YES    YES      NO      NO       03   006    000013   *******  002      **  ********************
----- END OF - DUMP CLASS  - LISTING -----
```

13. **If I want to list all the dump volumes that I have created over time, which command do I use?**

    Depending on the dump policies at your installation the output from this command can be quite large. To list out information about all dump volumes in the system, issue the following command:

```
HSEND LIST DVOL BCDS ODS(MHLRES3.DSN)
```

    The output (partial) that is written to the DASD data set looks like this:

```
-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 17: 44:04 ON 98/07/30 FOR SYSTEM=SC47

DUMP   VOL    UNIT     FILE  SOURCE              DUMPED     DUMPED                          SET OF DUMP
VOLSER STATUS TYPE     SEQ   VOLSER SMS CLASS    DATE       TIME     EX P DATE IDRC LIBRARY   VOLSERS
PLX474 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACMC1 N           1998/07/29 21:16:24                        PLX466 PLX474 PLX495
PLX475 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACDB1 N           1998/07/29 20:51:46                        PLX468 PLX475 PLX479
                                                                                           PLX487
PLX476 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACCI1 N           1998/07/29 20:49:30                        PLX469 PLX476
PLX477 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    TOTDB3 N           1998/07/29 21:06:54                        PLX477 PLX485 PLX493
PLX478 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACOR1 N           1998/07/29 21:12:26                        PLX471 PLX478 PLX486
PLX479 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACDB1 N           1998/07/29 20:51:46                        PLX468 PLX475 PLX479
                                                                                           PLX487
PLX480 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    TOTMQ1 N           1998/07/29 21:13:04                        PLX480
PLX481 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACIM1 N           1998/07/29 21:15:23                        PLX481 PLX498
PLX482 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    TOTDB2 N           1998/07/29 21:06:51                        PLX467 PLX470 PLX482
PLX483 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    TOTMQ2 N           1998/07/29 21:13:21                        PLX483
PLX484 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    BOOK04 N           1998/07/29 20:43:45                        PLX458 PLX472 PLX473
                                                                                           PLX484
PLX485 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    TOTDB3 N           1998/07/29 21:06:54                        PLX477 PLX485 PLX493
PLX486 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACOR1 N           1998/07/29 21:12:26                        PLX471 PLX478 PLX486
PLX487 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
                       01    DACDB1 N           1998/07/29 20:51:46                        PLX468 PLX475 PLX479
                                                                                           PLX487
PLX488 UNEXP  3490                    SYSPLEX2                      19 98/08/11 Y  *NO LIB*
```

```
                    01    DB2001 N           1998/07/29 21:12:16                        PLX488 PLX492 PLX496
PLX489 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DACOD1 N           1998/07/29 21:14:45                        PLX489 PLX490 PLX499
PLX490 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DACOD1 N           1998/07/29 21:14:45                        PLX489 PLX490 PLX499
PLX492 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DB2001 N           1998/07/29 21:12:16                        PLX488 PLX492 PLX496
PLX493 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    TOTDB3 N           1998/07/29 21:06:54                        PLX477 PLX485 PLX493
PLX495 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DACMC1 N           1998/07/29 21:16:24                        PLX466 PLX474 PLX495
PLX496 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DB2001 N           1998/07/29 21:12:16                        PLX488 PLX492 PLX496
PLX498 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DACIM1 N           1998/07/29 21:15:23                        PLX481 PLX498
PLX499 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                    01    DACOD1 N           1998/07/29 21:14:45                        PLX489 PLX490 PLX499
----- END OF - DUMP VOLUME - LISTING -----
```

14. **How can I find out to which dump volumes one of my primary volumes has been dumped?**

    If you know the primary volume, in this case DACMC1, use the following command:

```
HSEND LIST PRIMARY(DACMC1) BCDS ALLDUMPS ODS(MHLRES3.DSN2)
```

    The output that is written to the DASD data set looks like this:

```
******************************************************
- DFSMSHSM CONTROL DATASET - PRIMARY VOLUME-BCDS--- ALLDUMPS----- AT 17 :53:26 ON 98/07/30 FOR SYSTEM=SC47

 SOURCE                                          SET OF DUMP
 VOLSER GEN SMS  DUMPED     TIME    CLASS     EXP DATE   VOLSERS

 DACMC1  00 NO  98/07/29  21:16:24  SYSPLEX2  98/08/11  PLX466 PLX474 PLX495 ****** ****** ****** ****** ****** ******
 DACMC1  01 NO  98/07/22  21:31:42  SYSPLEX2  98/08/04  PLX243 PLX248 PLX231 ****** ****** ****** ****** ****** ******
 ----- END OF - PRIMARY VOLUME - LISTING -----
```

15. **How can I find out which data sets were dumped to a specific dump volume?**

    This information is in the dump volume VTOC copy data set that is created. In the previous example we know that one of the dump volumes that output from primary volume DACMC1 went to is dump volume PLX466. Specifying the dump volume volser (PLX466) in the following command lists out all the data sets in the specified dump volume:

```
HSEND LIST DUMPVOLUME(PLX466) DCONTENTS ODS(MHLRES3.DSN3)
```

    The output that is written to the data set looks like this:

```
-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 17: 58:07 ON 98/07/30 FOR SYSTEM=SC47

DUMP   VOL    UNIT     FILE  SOURCE              DUMPED     DUMPED                          SET OF DUMP
VOLSER STATUS TYPE     SEQ   VOLSER SMS CLASS    DATE       TIME     EX P DATE IDRC LIBRARY   VOLSERS
PLX466 UNEXP  3490                     SYSPLEX2                    19 98/08/11 Y  *NO LIB*
                      01    DACMC1 N            1998/07/29 21:16:24                        PLX466 PLX474 PLX495
DUMP COPY DATA SET NAME = TOTHSM.DMP.SYSPLEX2.VDACMC1.D98210.T241621
CONTENTS OF VTOC COPY FOR SOURCE VOLUME  DACMC1
DATASET NAME                                ORG  MULTI  CREATED   REF ERENCED  EXP DATE  RACF  PSWD   CHANGED
ALGUEIR.DB2.OUTPUT                          PS    NO    97/07/14  97/ 07/24    00/00/00  NO    NO     YES
ALGUEIR.ISPF.ISPPROF                        PO    NO    97/07/09  97/ 07/30    00/00/00  NO    NO     YES
ALGUEIR.LOG.MISC                            PS    NO    97/07/29  97/ 07/29    00/00/00  NO    NO     YES
ARTURO.DB2.OUTPUT                           PS    NO    97/07/17  97/ 07/24    00/00/00  NO    NO     YES
ARTURO.ISPF.ISPPROF                         PO    NO    97/07/17  97/ 07/25    00/00/00  NO    NO     YES
ARTURO.SPUFI.OUTPUT                         PS    NO    97/07/17  97/ 07/17    00/00/00  NO    NO     YES
BFS.LS130.LFS.ACCSCTL                       PS    YES   97/06/30  00/ 00/00    00/00/00  NO    NO     YES
CAT.MASTER                                  VS    ***   97/06/26  98/ 07/28    99/12/31  ***   ***    YES
CAT.MASTER.CATINDEX                         VS    ***   97/06/26  00/ 00/00    99/12/31  ***   ***    NO
CAT.SMPE                                    VS    ***   97/06/26  98/ 07/28    99/12/31  ***   ***    YES
```

16. **In the output from the LIST DUMPVOLUME, the dump tape data set name is given. Can I use this dump tape data set name to restore a volume outside DFSMShsm?**

Yes. Remember that DFSMShsm performs only one volume restore at a time. You can use the dump tape data set name in a job that invokes DFSMSdss RESTORE directly.

17. **How can I list the information about ML1 volumes that is recorded in the MCDS?**

    To obtain this information you use the following command:

```
HSEND LIST ML1 MCDS ODS(MHLRES3.DSN4)
```

    The output that is written to the DASD data set looks like this:

```
- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-MCDS--- LISTING ----- AT 18 :05:25 ON 98/07/30 FOR SYSTEM=SC47

 VOLSER DEVICE   VOLUME THRESHOLD FRAG HOSTID AUTO-   AUTO SDSP  MIN MIGRATED      DATE---TIME  SPACE-MGMT BUDEV DUMP
        TYPE     TYPE   HI--LOW  INDX MIG--BACK-DUMP RECL AVAIL AGE DS---TRKS    LAST MIGRATED  TYPE/AGE  CATGY CLASS

 TOTHS2   3390   LEV 1   090 ***  .193 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 TOTHS3   3390   LEV 1   090 ***  .415 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 TOTHS4   3390   LEV 1   090 ***  .243 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 TOTHS5   3390   LEV 1   090 ***  .214 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 ----- END OF - MIGRATE VOLUME - LISTING -----
```

18. **If I need information about all my migration volumes and want to use the MCDS and BCDS, which command should I use?**

    When you specify the BOTH keyword, the LIST command uses the MCDS and the BCDS as its source for information. The command syntax is:

```
HSEND LIST MIGRATIONVOLUME BOTH ODS(MHLRES3.DSN5)
```

    The output that is written to the data set looks like this:

    Note: Because we are not using ML2 in our system, no ML2 volumes are listed here.

```
- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-MCDS--- LISTING ----- AT 18 :05:52 ON 98/07/30 FOR SYSTEM=SC47

 VOLSER DEVICE   VOLUME THRESHOLD FRAG HOSTID AUTO-   AUTO SDSP  MIN MIGRATED      DATE---TIME  SPACE-MGMT BUDEV DUMP
        TYPE     TYPE   HI--LOW  INDX MIG--BACK-DUMP RECL AVAIL AGE DS---TRKS    LAST MIGRATED  TYPE/AGE  CATGY CLASS

 TOTHS2   3390   LEV 1   090 ***  .193 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 TOTHS3   3390   LEV 1   090 ***  .415 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 TOTHS4   3390   LEV 1   090 ***  .243 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
 TOTHS5   3390   LEV 1   090 ***  .214 **** **** P     ***  NO   *** 0000 000000 00/00/00 00:00  *** ***  ****  SYSPLEX1
----- END OF - MIGRATE VOLUME - LISTING -----

- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-BCDS--- LISTING ----- AT 18 :05:52 ON 98/07/30 FOR SYSTEM=SC47

                                                    CATALOG  LAST BACKED UP
 VOLSER SMS OWNED BY VSAM CATALOG                  ON VOLSER   DATE   TIME   DUMPCLASS  DUMPED     TIME   EXP DATE

 TOTHS2 NO  NOT OWNED BY A VSAM CATALOG                    00/00/00 00:00  SYSPLEX1  98/07/27  21:29:15 98/08/09
                                                                           SYSPLEX   97/07/21  23:58:05 97/08/03
 TOTHS3 NO  NOT OWNED BY A VSAM CATALOG                    00/00/00 00:00  SYSPLEX1  98/07/27  20:50:47 98/08/09
                                                                           SYSPLEX   97/07/21  22:56:23 97/08/03
 TOTHS4 NO  NOT OWNED BY A VSAM CATALOG                    00/00/00 00:00  SYSPLEX1  98/07/27  21:30:19 98/08/09
                                                                           SYSPLEX   97/07/22  00:00:32 97/08/03
 TOTHS5 NO  NOT OWNED BY A VSAM CATALOG                    00/00/00 00:00  SYSPLEX1  98/07/27  20:48:34 98/08/09
----- END OF - MIGRATE VOLUME - LISTING -----
```

19. **Is it possible to list information about my primary volumes?**

    Yes. The PRIMARYVOLUME or PVOL parameter provides information for primary volumes. You can request information about all primary volumes, or a specific primary volume, whether they are SMS-managed or not. You can also request that information be listed from the MCDS, BCDS, or both. In the following LIST command, we only want information for volume HSM14A. To list all primary volumes omit the volser parameter:

```
HSEND LIST PVOL(HSM14A) MCDS ODS(MHLRES5.DSN7)
```

The output that is written to the data set looks like this:

```
- DFSMSHSM CONTROL DATASET - PRIMARY VOLUME-MCDS--- LISTING ----- AT 18 :23:47 ON 98/07/30 FOR SYSTEM=SC54

 VOLSER DEVICE   VOLUME THRESHOLD FRAG HOSTID AUTO-   AUTO SDSP  MIN MIGRATED      DATE---TIME  SPACE-MGMT BUDEV DUMP
        TYPE     TYPE   HI--LOW  INDX MIG--BACK-DUMP RECL AVAIL AGE DS---TRKS    LAST MIGRATED  TYPE/AGE  CATGY CLASS

 HSM14A   3390   P SMS   095 080  .002 **** **** ****  ***  ***  *** 0000 ****** 98/07/30 15:50  MIG ***  ****  ONSITE
                                                                                                              OFFSITE
 ----- END OF - PRIMARY VOLUME - LISTING -----
```

20. **How can I generate a list of all tape volumes that I have in my DFSMShsm installation?**

    A list of all tape volumes that are defined to DFSMShsm can be quite exhaustive, so we recommend that you use an output data set. The information about tape volumes is recorded in the OCDS in a TTOC. The information that is returned from the following command includes all the tape information but does not show the individual data set information:

```
HSEND LIST TTOC ODS(MHLRES3.TTOC)
```

The output (partial) that is written to the data set looks like this:

```
 DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 18:27:39 O  N 98/07/30 FOR SYSTEM=SC47

 VOLSER    UNIT    VOL      REUSE     VALID   PCT    VOL    RACF  PREV    SUCC    NUM  ONE   ALT     LIB       STORAGE
           NAME    TYPE   CAPACITY    BLKS    VALID  STATUS       VOL     VOL     REC  FILE  VOL                GROUP
 MXX001    3490    SPILL  00050400  00042456  085    FULL    YES  *NONE*  *NONE*  042  YES   *NONE*  ITSOATL   SGLIB
 MXX003    3490    SPILL  00050400  00014874  030    FULL    YES  *NONE*  *NONE*  008  YES   *NONE*  ITSOATL   SGLIB
 MXX022    3490    SPILL  00050400  00046290  092    FULL    YES  *NONE*  MXX023  022  YES   *NONE*  ITSOATL   SGLIB
 MXX023    3490    SPILL  00050400  00033884  068    FULL    YES  MXX022  *NONE*  014  YES   *NONE*  ITSOATL   SGLIB
 MXX186    3490    SPILL  00050400  00045928  092    FULL    YES  *NONE*  *NONE*  024  YES   *NONE*  ITSOATL   SGLIB
 MXX210    3490    SPILL  00050400  00020581  041    FULL    YES  MXX475  *NONE*  007  YES   *NONE*  ITSOATL   SGLIB
 MXX475    3490    SPILL  00050400  00043904  088    FULL    YES  *NONE*  MXX210  038  YES   *NONE*  ITSOATL   SGLIB
 MXX016    3490    D(01)  00050400  00032469  065    FULL    YES  *NONE*  *NONE*  005  YES   *NONE*  ITSOATL   SGLIB
 MXX026    3490    D(01)  00050400  00040136  080    FULL    YES  *NONE*  *NONE*  018  YES   *NONE*  ITSOATL   SGLIB
 MXX037    3490    D(01)  00050400  00022174  044    FULL    YES  *NONE*  *NONE*  009  YES   *NONE*  ITSOATL   SGLIB
 MXX080    3490    D(01)  00050400  00008802  018    FULL    YES  *NONE*  *NONE*  017  YES   *NONE*  ITSOATL   SGLIB
 MXX084    3490    D(01)  00050400  00000000  000    FULL    YES  *NONE*  *NONE*  008  YES   *NONE*  ITSOATL   SGLIB
 MXX172    3490    D(01)  00050400  00088064  100    FULL    YES  *NONE*  MXX175  001  YES   *NONE*  ITSOATL   SGLIB
 MXX175    3490    D(01)  00050400  00023211  046    FULL    YES  MXX172  *NONE*  011  YES   *NONE*  ITSOATL   SGLIB
```

21. **How can I list the data sets that have been copied to a particular volume?**

    In the output from the previous command, the first volume listed has a volser of MXX001. If you specify the MXX001 volser and issue the following command, you will get data set information returned to you:

```
HSEND LIST TTOC(MXX001) ODS(MHLRES3.TTOC)
```

The output (partial) that is written to the data set looks like this:

```
 DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 18:30:45 O  N 98/07/30 FOR SYSTEM=SC47

 VOLSER    UNIT    VOL      REUSE     VALID   PCT    VOL    RACF  PREV    SUCC    NUM  ONE   ALT     LIB       STORAGE
           NAME    TYPE   CAPACITY    BLKS    VALID  STATUS       VOL     VOL     REC  FILE  VOL                GROUP
 MXX001    3490    SPILL  00050400  00042456  085    FULL    YES  *NONE*  *NONE*  042  YES   *NONE*  ITSOATL   SGLIB

             DATA SET NAME                        NUM BLOCKS  RELATIVE FBID  VSAM  RACF   LAST REF DATE  EXP DATE
    TOTHSM.BACK.T434907.LOGR.SYSPLEX.I8114          000169        0001       YES   NO       ********     ********
    TOTHSM.BACK.T544907.DONNAS.A0000018.I8114       000001        0002       NO    NO       ********     ********
    TOTHSM.BACK.T085107.SHARE.HSM.I8114             000003        0003       NO    NO       ********     ********
    TOTHSM.BACK.T025207.LOGR.SYSPLEX.I8114          000171        0004       YES   NO       ********     ********
    TOTHSM.BACK.T055207.LOGR.SYSPLEX.I8114          000170        0005       YES   NO       ********     ********
    TOTHSM.BACK.T085207.LOGR.SYSPLEX.I8114          000170        0006       YES   NO       ********     ********
    TOTHSM.BACK.T195207.OS390CB.R6CB01.I8114        000088        0008       NO    NO       ********     ********
    TOTHSM.BACK.T215207.RALPHR.BRODCAST.I8114       000001        0009       NO    NO       ********     ********
    TOTHSM.BACK.T325207.SILVIO.SC62.I8114           000003        0011       NO    NO       ********     ********
    TOTHSM.BACK.T475207.HAIMO.UQ13783.I8114         000002        0012       NO    NO       ********     ********
    TOTHSM.BACK.T575207.HAIMO.UQ09719.I8114         000026        0013       NO    NO       ********     ********
---- EDITED DATA SET LISTING ----
```

22. **Is there a way to list TTOC information for a given volume but not get a list of data sets?**

    Specify the TTOC parameter to list tape volume information from the OCDS without listing all the data set related information for the volume. If you do not want a list of data sets on the volume, specify the NODSI parameter to the LIST command:

```
HSEND LIST TTOC(MXX001) NODSI ODS(MHLRES3.DSN8)
```

    The output that is written to the data set looks like this:

```
DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 18:34:26 O  N 98/07/30 FOR SYSTEM=SC47

VOLSER    UNIT    VOL      REUSE     VALID   PCT    VOL    RACF  PREV    SUCC    NUM  ONE   ALT     LIB       STORAGE
          NAME    TYPE   CAPACITY    BLKS    VALID  STATUS       VOL     VOL     REC  FILE  VOL               GROUP
MXX001    3490    SPILL  00050400  00042456  085    FULL    YES  *NONE*  *NONE*  042  YES   *NONE*  ITSOATL   SGLIB
----- END OF - TAPE VOLUME TTOC   - LISTING -----
```

23. **Is there a way to obtain information about the status of tapes for DFSMShsm use such as partially filled and empty tapes?**

    The LIST command supports an optional parameter, SELECT. This parameter can be used in a number of ways to list only those TTOC entries in which you are interested. A number of subparameters can be used. We show you some of the SELECT options that you can use (without including the output from each command) just to give you an idea of the kind of information that is available.

    The following command generates a list of all those backup volumes that are empty along with those that have some data on them but are not marked full (the tapes you could then mount to begin backup):

```
HSEND LIST TTOC SELECT(BACKUP NOTFULL) ODS(MHLRES3.DSNA)
```

    The following command generates a list of ML2 volumes that were taken away from another function because the recall function needed a data set on the volume:

```
HSEND LIST TTOC SELECT(ML2 RECALLTAKEAWAY) ODS(MHLRES3.DSNB)
```

24. **How can I obtain a list of all DFSMShsm users that have been authorized by the DFSMShsm AUTH command?**

    Use the following command to quickly identify users that should or should not have DFSMShsm authorization:

```
HSEND LIST USER ODS(MHLRES3.DSN9)
```

    The output that is written to the data set looks like this:

```
DFSMSHSM CONTROL DATASET - USER-- LISTING --- AT 18:39:25 ON 98/07/30 FOR SYSTEM=SC47

  USERID   AUTH

 CICSRS2   USER
 CICSRS3   USER
```

```
DODELL    USER
HAIMO     CNTL
HGSRESA   USER
MHLRES3   CNTL
MHLRES4   CNTL
SMSOPT    USER
VAINI     USER
----- END OF - USER - LISTING -----
```

In the command output a number of users have an authority of CNTL. We recommend that this authority be assigned only as required and that the majority of users be assigned USER authority.

5.2 Query Command

As DFSMShsm processes its units of work, it maintains information in its address space. As the LIST command obtains its information from the CDSs there is another command that will interrogate the address space. This command is the QUERY command, and it can be used to display real-time information about your DFSMShsm system. The QUERY command returns information that includes:

- The current SETSYS parameters
- The current ABARS parameters
- The status of outstanding DFSMShsm requests
- Volume space usage
- The status of each volume and data set subtask and long-running commands
- The progress of automatic functions
- The parameters used at DFSMShsm startup

1. **How do I issue the QUERY command?**

   You can issue the QUERY command by using the HSEND command in TSO or the MVS MODIFY command on a system console:

```
HSEND QUERY request
MODIFY HSM54,QUERY request
```

   You can also issue the QUERY command from a REXX program.

   When you issue the QUERY command you must also specify one of the optional parameters in order to produce any output. It is also possible to place the QUERY command in your DFSMShsm startup procedure, but remember that some of the information that may be displayed may be incorrect as DFSMShsm is not yet fully initialized.

2. **Where will the output from the command be returned?**

   If you issue the HSEND command, the output is directed back to your TSO terminal and to the DFSMShsm log. If you issue the MODIFY command, the information is returned to the system log.

3. **What is the best version of the command to use?**

   It depends on the output that you expect to receive and what you want to do with it. Remember that if there is a lot of output associated with the command and it is displayed at your terminal, you may find that not all the information is displayed on one screen. Or you may find that you miss the

information first time around and have to issue the command again. In the examples that follow we use the MVS MODIFY command so that we can capture the information on the system log and perform searches for information or messages in which we are interested.

4. **Are there abbreviations for the commands?**

   In our examples in most cases we use F for the MVS MODIFY command, and Q for the DFSMShsm QUERY command. We also use abbreviations for the optional parameters.

5. **How do I find detailed information about the QUERY command syntax?**

   In addition to referring to the *DFSMShsm Storage Administration Reference* for a comprehensive description of the command and optional parameters, you can use the ARC0101I message to check the full command syntax.

6. **From where is all the information gathered?**

   The information is gathered from control records held within the DFSMShsm address space.

7. **Why can't I use the LIST command to get at this information?**

   The LIST command operates on the CDSs, not the DFSMShsm address space.

8. **I have more than one DFSMShsm system sharing CDSs and the journal. On which system should I issue the command?**

   The information returned will apply to the system where you issue the command. Therefore if you have more than one system, you may have to issue the command on more than one system. Some of the information returned applies to the DFSMShsm installation in its entirety, but some information is applicable only to the system where you issue the command.

9. **How can I list what the current ABARS control parameters that are in effect?**

   If you want to query the current DFSMShsm control parameters that apply to aggregate backup and recovery, issue the following command:

```
F HSM54,Q AB
```

   The information is returned in the following messages:

```
ARC0101I QUERY ABARS COMMAND STARTING
ARC6008I AGGREGATE BACKUP/RECOVERY PROCNAME = ABARS54
ARC6009I AGGREGATE BACKUP/RECOVERY MAXADDRESSSPACE = 02
ARC6366I AGGREGATE BACKUP/RECOVERY UNIT NAME = 3490
ARC6368I AGGREGATE BACKUP/RECOVERY ACTIVITY LOG
ARC6368I (CONT.) MESSAGE LEVEL IS FULL
ARC6371I AGGREGATE RECOVERY ML2 TAPE UNIT NAME = 3490
ARC6372I NUMBER OF ABARS I/O BUFFERS = 02
ARC6373I ABARS ACTIVITY LOG OUTPUT TYPE = DASD
ARC6033I AGGREGATE RECOVERY UNIT NAME = 3490
ARC6036I AGGREGATE BACKUP OPTIMIZE = 3
ARC6036I AGGREGATE RECOVERY TGTGDS = SOURCE
ARC6036I AGGREGATE RECOVERY ABARSVOLCOUNT = *NONE*
ARC6036I AGGREGATE RECOVERY PERCENTUTILIZED = 090
ARC6036I AGGREGATE BACKUP/RECOVERY ABARSDELETEACTIVITY
ARC6036I (CONT.) = NO
ARC6036I AGGREGATE BACKUP/RECOVERY ABARSTAPES = STACK
ARC0101I QUERY ABARS COMMAND COMPLETED
```

You can use other optional QUERY command parameters to display information pertinent to the ABARS environment. Be aware that this is not the only ABARS parameter.

10. **How can I display what is currently running in DFSMShsm?**

    To display the status of each DFSMShsm volume and data set subtask, long-running commands, and the virtual storage usage above and below 16 MB, issue the following command:

```
F HSM54,Q ACT
```

The information is returned in the following messages:

```
ARC0101I QUERY ACTIVE COMMAND STARTING
ARC0144I AUDIT=NOT HELD AND INACTIVE, LIST=NOT HELD
ARC0144I (CONT.) AND INACTIVE, RECYCLE=NOT HELD AND INACTIVE,
ARC0144I (CONT.) REPORT=NOT HELD AND INACTIVE
ARC0160I MIGRATION=NOT HELD, AUTOMIGRATION=NOT HELD,
ARC0160I (CONT.) RECALL=NOT HELD, TAPERECALL=NOT HELD, DATA SET
ARC0160I (CONT.) MIGRATION=INACTIVE, VOLUME MIGRATION=INACTIVE, DATA
ARC0160I (CONT.) SET RECALL=INACTIVE
ARC0163I BACKUP=NOT HELD, AUTOBACKUP=NOT HELD,
ARC0163I (CONT.) RECOVERY=NOT HELD, TAPEDATASETRECOVERY=NOT HELD, DATA
ARC0163I (CONT.) SET BACKUP=INACTIVE, VOLUME BACKUP=INACTIVE, DATA SET
ARC0163I (CONT.) RECOVERY=INACTIVE, VOLUME RECOVERY=INACTIVE
ARC0642I DUMP=NOT HELD, AUTODUMP=NOT HELD, VOLUME
ARC0642I (CONT.) DUMP=INACTIVE, VOLUME RESTORE=INACTIVE, DATA SET
ARC0642I (CONT.) RESTORE=INACTIVE
ARC0142I MOVEMENT OF BACKUP VERSIONS, CURRENTLY IN
ARC0142I (CONT.) PROCESS
ARC0437I - TAPECOPY NOT HELD AND INACTIVE
ARC0437I - TAPEREPL NOT HELD AND INACTIVE
ARC0415I EXPIREBV=NOT HELD AND INACTIVE, LAST STORED
ARC0415I (CONT.) BACKUP VERSION KEY=, LAST STORED ABARS VERSION KEY=,
ARC0415I (CONT.) LAST PLANNED END KEY=
ARC0460I PRIVATE AREA LIMIT=8172K, UNALLOCATED=5928K,
ARC0460I (CONT.) LARGEST FREE AREAS=5884K, 16K
ARC0460I EXTENDED PRIVATE AREA LIMIT=1887M,
ARC0460I (CONT.) UNALLOCATED=1863M, LARGEST FREE AREAS=1863M, 104K
ARC6018I AGGREGATE BACKUP/RECOVERY = INACTIVE
ARC6019I AGGREGATE BACKUP = NOT HELD, AGGREGATE
ARC6019I (CONT.) RECOVERY = NOT HELD
ARC0101I QUERY ACTIVE COMMAND COMPLETED
```

11. **If I want to find out how DFSMShsm automatic functions are progressing and the number of volumes to be processed, which command should I use?**

    To find out exactly how the automatic functions are progressing in terms of volumes processed and to be processed, issue the following command:

```
F HSM54,Q AUTOP
```

    The information is returned in the following messages:

```
ARC0101I QUERY AUTOPROGRESS COMMAND STARTING
ARC0247I PRIMARY SPACE MANAGEMENT IS CURRENTLY
ARC0247I (CONT.) PROCESSING DFSMSHSM MANAGED VOLUMES
ARC0246I SMS VOLUMES RESTRICTED TO PROCESSING BY THIS
ARC0246I (CONT.) PROCESSING UNIT: NOT PROCESSED=0, TOTAL=0, SMS
ARC0246I (CONT.) VOLUMES NOT RESTRICTED TO PROCESSING BY ANY
ARC0246I (CONT.) PROCESSING UNIT: NOT PROCESSED=0, TOTAL=4, NON-SMS
ARC0246I (CONT.) VOLUMES: NOT PROCESSED=0, TOTAL=0
ARC0101I QUERY AUTOPROGRESS COMMAND COMPLETED
```

    The same command gives the following result when automatic functions are not in progress:

```
ARC0101I QUERY AUTOPROGRESS COMMAND STARTING
ARC0247I NO AUTOMATIC FUNCTION IS CURRENTLY PROCESSING
ARC0247I (CONT.) DFSMSHSM MANAGED VOLUMES
ARC0101I QUERY AUTOPROGRESS COMMAND COMPLETED
```

12. **If I just wanted to get information with respect to my backup environment, which command could I use?**

    To find out your current automatic backup and dump parameters, specify the BACK parameter. In our example the DAILY(6) parameter tells DFSMShsm to list the information for the sixth day of the month. If DAILY is specified without a day, DFSMShsm lists all daily backup volumes.

```
F HSM54,QUERY BACK(DAILY(6))
```

    The information is returned in the following messages:

```
ARC0101I QUERY BACKUP COMMAND STARTING
ARC0638I MAXDUMPTASKS=01, ADSTART=(0400 0500 0600),
ARC0638I (CONT.) DUMPIO=(1,4), VOLUMEDUMP=(NOCC)
ARC0273I DUMP CYCLE LENGTH=7 DAY(S), CYCLE=NNNNNNY,
ARC0273I (CONT.) TODAY IS DAY=1, CYCLE START DATE=95/01/06, LEVEL
ARC0273I (CONT.) FUNCTIONS NOT ELIGIBLE TO BE STARTED, CURRENT TIME IS
ARC0273I (CONT.) NOT WITHIN AUTODUMP START WINDOW
ARC0274I BACKUP=YES(TAPE(3490)), SPILL=NO,
ARC0274I (CONT.) MAXDSRECOVERTASKS=01
ARC0154I MAXBACKUPTASKS=03, ABSTART= (1400 1500 1600),
ARC0154I (CONT.) VERSIONS=001, FREQUENCY=000, SKIPABPRIMARY=NO, BACKUP
ARC0154I (CONT.) PREFIX=HSM, INCREMENTALBACKUP=ORIGINAL,
ARC0154I (CONT.) PROFILEBACKUP=YES, INUSE=(RETRY=NO, DELAY=015,
ARC0154I (CONT.) SERIALIZATION=REQUIRED)
ARC0271I BACKUP CYCLE LENGTH=07 DAY(S), CYCLE=YYYYYYN,
ARC0271I (CONT.) TODAY IS DAY=01, VOLUME LIMIT/DAY=0001, TOTAL BACKUP
ARC0271I (CONT.) VOLUMES=00003, CYCLE START DATE=95/01/06
ARC0101I QUERY BACKUP COMMAND COMPLETED
```

13. **How can I get volume information related to my backup environment?**

    You can get volume information by using a format of the following command:

```
QUERY BACKUP(ALL | DAILY(nn) | SPILL | UNASSIGNED)
```

    To list out all backup parameters and additionally the daily backup volumes, issue the following command:

```
F HSM54,QUERY BACK(DAILY)
```

    The information is returned in the following messages:

```
ARC0101I QUERY BACKUP COMMAND STARTING
ARC0638I MAXDUMPTASKS=01, ADSTART=(0400 0500 0600),
ARC0638I (CONT.) DUMPIO=(1,4), VOLUMEDUMP=(NOCC)
ARC0273I DUMP CYCLE LENGTH=7 DAY(S), CYCLE=NNNNNNY,
ARC0273I (CONT.) TODAY IS DAY=1, CYCLE START DATE=95/01/06, LEVEL
ARC0273I (CONT.) FUNCTIONS NOT ELIGIBLE TO BE STARTED, CURRENT TIME IS
ARC0273I (CONT.) NOT WITHIN AUTODUMP START WINDOW
ARC0274I BACKUP=YES(TAPE(3490)), SPILL=NO,
ARC0274I (CONT.) MAXDSRECOVERTASKS=01
ARC0154I MAXBACKUPTASKS=03, ABSTART= (1400 1500 1600),
ARC0154I (CONT.) VERSIONS=001, FREQUENCY=000, SKIPABPRIMARY=NO, BACKUP
ARC0154I (CONT.) PREFIX=HSM, INCREMENTALBACKUP=ORIGINAL,
ARC0154I (CONT.) PROFILEBACKUP=YES, INUSE=(RETRY=NO, DELAY=015,
ARC0154I (CONT.) SERIALIZATION=REQUIRED)
ARC0271I BACKUP CYCLE LENGTH=07 DAY(S), CYCLE=YYYYYYN,
ARC0271I (CONT.) TODAY IS DAY=01, VOLUME LIMIT/DAY=0001, TOTAL BACKUP
ARC0271I (CONT.) VOLUMES=00003, CYCLE START DATE=95/01/06
ARC0164I DAY=06     VOLS = M195A3-A M195A2-A M195AW-A
ARC0101I QUERY BACKUP COMMAND COMPLETED
```

Notice the addition of ARC0164I to indicate the backup volumes.

14. **How do I display the current backup version of the CDSs and journal and find out their backup names?**

    Issue the following command to find out all the information you need with respect to the backup technique and status of the CDSs and journal backup copies:

```
F HSM54,Q CDSV
```

The information is returned in the following messages:

```
ARC0101I QUERY CDSVERSIONBACKUP COMMAND STARTING
ARC0375I CDSVERSIONBACKUP,
ARC0375I (CONT.) MCDSBACKUPDSN=HSM.MCDS.BACKUP,
ARC0375I (CONT.) BCDSBACKUPDSN=HSM.BCDS.BACKUP,
ARC0375I (CONT.) OCDSBACKUPDSN=HSM.OCDS.BACKUP,
ARC0375I (CONT.) JRNLBACKUPDSN=HSM.JRNL.BACKUP
ARC0376I BACKUPCOPIES=0004, BACKUPDEVICECATEGORY=DASD,
ARC0376I (CONT.) LATESTFINALQUALIFIER=D0000006, DATAMOVER=DSS
ARC0101I QUERY CDSVERSIONBACKUP COMMAND COMPLETED
```

15. **How can I display the current space usage of the CDSs and the journal and the serialization technique currently being used?**

    Use this command:

```
F HSM54,QUERY CDS
```

The information is returned in the following messages:

```
ARC0101I QUERY CONTROLDATASETS COMMAND STARTING
ARC0947I CDS SERIALIZATION TECHNIQUE IS RESERVE
ARC0148I MCDS TOTAL SPACE=0012960 K-BYTES, CURRENTLY
ARC0148I (CONT.) ABOUT 005% FULL, WARNING THRESHOLD=080%, TOTAL
ARC0148I (CONT.) FREESPACE=099%
ARC0948I MCDS INDEX TOTAL SPACE=0000042 K-BYTES,
ARC0948I (CONT.) CURRENTLY ABOUT 004% FULL, WARNING THRESHOLD=080%
ARC0148I BCDS TOTAL SPACE=0012960 K-BYTES, CURRENTLY
ARC0148I (CONT.) ABOUT 005% FULL, WARNING THRESHOLD=080%, TOTAL
ARC0148I (CONT.) FREESPACE=098%
ARC0948I BCDS INDEX TOTAL SPACE=0000042 K-BYTES,
ARC0948I (CONT.) CURRENTLY ABOUT 004% FULL, WARNING THRESHOLD=080%
ARC0148I OCDS TOTAL SPACE=0012960 K-BYTES, CURRENTLY
ARC0148I (CONT.) ABOUT 005% FULL, WARNING THRESHOLD=080%, TOTAL
ARC0148I (CONT.) FREESPACE=099%
ARC0948I OCDS INDEX TOTAL SPACE=0000042 K-BYTES,
ARC0948I (CONT.) CURRENTLY ABOUT 004% FULL, WARNING THRESHOLD=080%
ARC0148I JOURNAL TOTAL SPACE=0004305 K-BYTES,
ARC0148I (CONT.) CURRENTLY ABOUT 006% FULL, WARNING THRESHOLD=080%,
ARC0148I (CONT.) TOTAL FREESPACE=093%
ARC0101I QUERY CONTROLDATASETS COMMAND COMPLETED
```

16. **How can I query DFSMShsm requests?**

    A number of options are available to display the requests associated with data sets, and userids and to list all requests.

    The following command lists all requests for data set VIALLI.KING:

```
F HSM54,Q DATASETNAME(VIALLI.KING)
```

    The following command lists all requests associated with user VIALLI:

```
F HSM54,Q USER(VIALLI)
```

    Note: If you omit the user ID, DFSMShsm lists requests for all users.

    The following command lists request number 45:

```
F HSM54,Q REQUEST(45)
```

    Note: If you omit the request number, DFSMShsm lists all requests.

17. **How do I display all my ML2 volumes regardless of whether they are tape or DASD?**

    If you are using DASD for ML2, the following command shows you the key ranges associated with the DASD volumes and all ML2 tape volumes that are currently selected for use for migration and secondary space management:

```
F HSM54,Q MIGRATIONLEVEL2
```

Note: ML2 volumes that are currently selected by recycle are not displayed.

18. **Is it possible to display the non-SMS recall pools that have been set up at my installation?**

    If you are using recall pools, issue the following command to interrogate the recall pool environment:

```
F HSM54,QUERY P
```

19. **What command can I use to show the non-SMS data sets that might not be migrating because the SETMIG command has been used?**

    For data sets that are being prevented from migrating because a SETMIG LEVEL(hlq) command has been issued, use the following command:

```
F HSM54,QUERY RET
```

    The information is returned in the following messages:

```
ARC0101I QUERY RETAIN COMMAND STARTING
ARC0175I LEVEL QUALIFIER AND MIGRATION RESTRICTION TYPE
ARC0176I QUALIFIER=SYS1. RESTRICTION TYPE=NOMIGRATION
ARC0176I QUALIFIER=SYSCTLG. RESTRICTION
ARC0176I (CONT.) TYPE=NOMIGRATION
ARC0176I QUALIFIER=HSM. RESTRICTION TYPE=NOMIGRATION
ARC0101I QUERY RETAIN COMMAND COMPLETED
```

20. **How do I display the DFSMShsm control parameters?**

    As the SETSYS parameters may have been altered since you started DFSMShsm and the ARCCMDxx member may no longer reflect what is currently set on the system, you will want to check what is in effect now. If you are unsure as to what SETSYS parameters are currently in place, issue the following command:

```
F HSM54,Q SETSYS
```

    The information is returned in a series of messages that reflect your current DFSMShsm environment.

21. **Is there any way in which I can display the space usage of my non-SMS primary volumes and my ML1 volumes?**

    The command that allows you to list the space usage of volumes can be applied to all your volumes or a specific volser coded. Use the following command to list volume HSM14D:

```
F HSM54,Q SPACE(HSM14D)
```

    The information is returned in the following messages:

```
ARC0400I VOLUME HSM14D IS 94% FREE, 000051 FREE
ARC0400I (CONT.) TRACK(S), 03163 FREE CYLINDER(S), FRAG .006
ARC0401I LARGEST EXTENTS FOR HSM14D ARE CYLINDERS
ARC0401I (CONT.) 3132    28     3, TRACKS   46990     435      57
ARC0401I (CONT.) 14
ARC0402I VTOC FOR HSM14D IS 0090 TRACKS(04500 DSCBS),
ARC0402I (CONT.) 04487 FREE DSCBS(99% OF TOTAL)
```

Note: If we had omitted the volser, DFSMShsm would have returned this information for all non-SMS primary and ML1 volumes. Additionally no ARC0101I message would be issued.

22. **If I am unsure which PARMLIB member was used at the start of DFSMShsm, is there a way that I can obtain this information?**

    For a list of the parameters that were specified on the PROC statement in the DFSMShsm startup procedure, issue the following command:

```
F HSM54,Q STAR
```

    The information is returned in the following messages:

```
ARC0101I QUERY STARTUP COMMAND STARTING
ARC0143I PARMLIB MEMBER=ARCCMD54, DFSMSHSM AUTHORIZED
ARC0143I (CONT.) USERID=HSM, HOSTID=1, PRIMARY HOST=YES, LOGSW=YES,
ARC0143I (CONT.) STARTUP=YES, EMERGENCY=NO, CDSQ=NO, CDSR=NO, PDA=YES,
ARC0143I (CONT.) RESTART=NOT SPECIFIED, CDSSHR=NO
ARC0101I QUERY STARTUP COMMAND COMPLETED
```

23. **Is there a quick way that I can check on the DFSMShsm activity that has taken place today?**

    DFSMShsm keeps track of activity on a daily basis in the daily statistics record (DSR). To display the information at any time throughout the day, issue the following command:

```
F HSM54,Q STAT
```

    The information is returned in the following messages:

```
ARC0101I QUERY STATISTICS COMMAND STARTING
ARC0155I DFSMSHSM STATISTICS FOR 98/07/31
ARC0156I STARTUPS=03, SHUTDOWNS=00, ABENDS=00,
ARC0156I (CONT.) MWES=0021, CPU TIME=01578.37 SECONDS
ARC0157I DS MIGRATE L1=000001, DS MIGRATE L2=000000,
ARC0157I (CONT.) DS EXTENT REDUCTIONS=000000, DS MIGRATE FAIL=000,
ARC0157I (CONT.) TRKS MIGRATE=000021, KBYTES MIGRATE=000000424
ARC0158I DS RECALL L1=000000, DS RECALL L2=000000, DS
ARC0158I (CONT.) RECALL FAIL=000,  KBYTES RECALL=000000000
ARC0159I DS BACKUP=000001, DS BACKUP FAIL=001, DS
ARC0159I (CONT.) RECOVER=000000, DS RECOVER FAIL=000
ARC0641I VOL DUMP=0, VOL DUMP FAIL=0, VOL RESTORE=0,
ARC0641I (CONT.) VOL RESTORE FAIL=0, DS RESTORE=0, DS RESTORE FAIL=0
ARC0145I DS DELETED=000000, DS DELETE FAILED=000
ARC0146I RECYCLED BACKUP VOLUMES=0000, DS=000000,
ARC0146I (CONT.) BLOCKS=000000
ARC0146I RECYCLED MIGRATION VOLUMES=0000, DS=000000,
ARC0146I (CONT.) BLOCKS=000000
ARC0101I QUERY STATISTICS COMMAND COMPLETED
```

24. **I have been getting some strange error messages from DFSMShsm lately. Is there any way of getting more information about errors before we call support?**

    DFSMShsm automatically records some errors that it receives and records others by using the diagnosis command, TRAP. To see which TRAP activity has taken place, issue the following command:

```
F HSM54,Q T
```

    The information is returned in the following messages:

```
ARC0101I QUERY TRAPS COMMAND STARTING
ARC0205I TRAP IN MODULE ARCRSTR FOR CODE 00002,
ARC0205I (CONT.) TIMES=0012, TYPE=BY OCCURRENCE
ARC0205I TRAP IN MODULE ARCMDSUV FOR CODE 00068,
ARC0205I (CONT.) TIMES=0004, TYPE=BY OCCURRENCE
ARC0205I TRAP IN MODULE ARCMDSS FOR CODE 00068,
ARC0205I (CONT.) TIMES=0004, TYPE=BY OCCURRENCE
ARC0205I TRAP IN MODULE ARCMDSUV FOR CODE 00019,
ARC0205I (CONT.) TIMES=0002, TYPE=BY OCCURRENCE
ARC0205I TRAP IN MODULE ARCRSTR FOR CODE 00004,
ARC0205I (CONT.) TIMES=0001, TYPE=BY OCCURRENCE
ARC0101I QUERY TRAPS COMMAND COMPLETED
```

25. **How can I display the non-SMS volume pools that I have defined ?**

    Issue the following command to see the volume pool information:

```
F HSM54,Q VOLUMEPOOL
```

26. **DFSMShsm seems to be taking a long time to do anything. Is there any way that I can display the amount of non daily work that DFSMShsm has waiting?**

    DFSMShsm manages its work by building management work elements (MWEs). The MWEs are scheduled to special work queues associated with the function that they have been asked to perform. To see the oustanding MWEs or work that DFSMShsm has to do, issue the following command:

```
F HSM54,Q W
```

    The information is returned in the following messages:

```
ARC0101I QUERY WAITING COMMAND STARTING
ARC0168I WAITING MWES: MIGRATE=0000, RECALL=0001,
ARC0168I (CONT.) DELETE=0000, BACKUP=0001, RECOVER=0000, COMMAND=0002,
ARC0168I (CONT.) ABACKUP=0000, ARECOVER=0000, TOTAL=0000
ARC0101I QUERY WAITING COMMAND COMPLETED
```

    Note: The output excludes requests generated as a result of primary and secondary space management, automatic backup and automatic dump.

5.3 Report Command

The REPORT command is used to gather and consolidate statistics about DFSMShsm′s operations and functions. DFSMShsm during its processing stores daily function and volume related information in the MCDS. The REPORT command allows you to gather information that is generated:

- At a function level for:
  - Backup
  - Migration
  - Delete
  - Recall
  - Recover
  - Recycle
  - Spill
- At a daily level
- At a volume level
- For statistics before or after a certain date
- For statistics between certain dates
- For a summary of all statistics reports

In this section we show you several REPORT commands that you can use, to give you an idea of the type of information that is available.

1. **Who can issue the REPORT command?**

The REPORT command is submitted primarily by a DFSMShsm-authorized storage administrator who uses the DFSMShsm HSENDCMD command from a TSO terminal. In the examples that follow we use the abbreviated command, HSEND, to issue our report commands.

2. **Is there anything I need to do to make sure that I am keeping these statistics?**

   There is a SETSYS parameter that affects how long DFSMShsm keeps daily and volume related statistics. The following command direct DFSMShsm to keep the statistics for seven days and to delete them during secondary space management:

```
SETSYS MIGRATIONCLEANUPDAYS(14 7)
```

   You need to consider over which time period you want to keep statistics so you can prevent deleting needed information.

3. **Where does DFSMShsm get its information from for the daily statistics report?**

   DFSMShsm obtains its information from a record in the MCDS called the daily statistics record (DSR). These records are written to the MCDS each hour. Any activity in the hosts, other than the host from which the REPORT command is issued, since the last update is not reported.

4. **Where does DFSMShsm get its information from for the volume statistics report?**

   DFSMShsm obtains its information from a record in the MCDS called the volume statistics record (VSR). These records are written to the MCDS each hour. Any activity in the hosts, other than the host from which the REPORT command is issued, since the last update is not reported.

5. **How many REPORT commands can I issue at any one time?**

   DFSMShsm allows only one REPORT command to run at a time.

6. **Where can I send the output of the REPORT command?**

   The REPORT command supports direction of the output to either a SYSOUT class or to an output data set. If you want to obtain a printed listing, use the following command to direct the output to your SYSOUT class associated with print:

```
REPORT request SYSOUT(class)
```

   If you want to direct the output to DASD, use the following command:

```
REPORT request ODS(dsname)
```

   When directing the output to DASD, you must specify the fully qualified data set name. If it does not exist, DFSMShsm dynamically allocates it. If it does exist and has the correct attributes, DFSMShsm appends the output to the end of the data set.

7. **How can I obtain a summary of all function-related activity that has taken place today?**

   Issue the following command:

```
HSEND REPORT DAILY FUNCTION ODS(MHLRES5.DAILY.STATS.AUG03)
```

   The output that is written to the data set looks like this:

```
DFSMSHSM STATISTICS REPORT ------- AT 18:06:03 ON 98/08/03 FOR SYSTEM=SC54

DAILY STATISTICS REPORT FOR 98/08/03

STARTUPS=005, SHUTDOWNS=005, ABENDS=000, WORK ELEMENTS PROCESSED=000016, BKUP VOL RECYCLED=00000, MIG VOL RECYCLED=00000
DATA SET MIGRATIONS BY VOLUME REQUEST= 0000000, DATA SET MIGRATIONS BY DATA SET REQUEST= 00000, BACKUP REQUESTS= 0000002
EXTENT REDUCTIONS= 0000000
FULL VOLUME DUMPS= 000000 REQUESTED, 00000 FAILED; DUMP COPIES= 000000 REQUESTED, 00000 FAILED
FULL VOLUME RESTORES= 000000 REQUESTED, 00000 FAILED; DATASET RESTORES= 000000 REQUESTED, 00000 FAILED

                    NUMBER  ------READ--------  -----WRITTEN------  ------REQUESTS---- AVERAGE ------AVERAGE TIME-------
   HSM FUNCTION    DATASETS TRK/BLK    BYTES    TRK/BLK    BYTES    SYSTEM USER FAILED   AGE   QUEUED WAIT PROCESS TOTAL

MIGRATION
 PRIMARY - LEVEL 1 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
 SUBSEQUENT MIGS   0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
 PRIMARY - LEVEL 2 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
RECALL
 LEVEL 1 - PRIMARY 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
 LEVEL 2 - PRIMARY 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
DELETE
 MIGRATE DATA SETS 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
 PRIMARY DATA SETS 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
BACKUP
 DAILY BACKUP      0000002  0000000  000000000K 0000000  000000000K 000002 00000 00000  00005  0000  00000  05632  05632
 SUBSEQUENT BACKUP 0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
 DELETE BACKUPS    0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
RECOVER
 BACKUP - PRIMARY  0000000  0000000  000000000K 0000000  000000000K 000000 00000 00000  00000  0000  00000  00000  00000
RECYCLE
 BACKUP - SPILL    0000000  0000000             0000000             000000 00000 00000  00000  0000  00000  00000  00000
 MIG L2 - MIG L2   0000000  0000000             0000000             000000 00000 00000  00000  0000  00000  00000  00000
```

   If OW31281 is installed on your system, the header of the REPORT output also contains the number of mounts saved by RECALL and RECOVER. This information is useful for addressing the changes in mounts when using larger capacity tapes like the Magstar.

8. **How do I obtain information for periods other than the current day?**

   If the DRSs are still available and have not been deleted, you can use a modification of the following command to specify the period in which you are interested:

```
REPORT DAILY FROMDATE(98/07/24) FUNCTION TODATE(98/07/31) ODS(MHLRES5.DS)
```

   The output is similar to the DSR. The difference is that you now get information at a function level:

   - A function summary for the period July 24 to July 31 inclusive
   - A function summary report totaling all activity for the requested period

9. **How do I request information detailing the statistics for a particular function?**

   The options that are available to you on the FUNCTION paramater are:

   - BACKUP
   - DELETE

- MIGRATION
- RECALL
- RECOVER
- RECYCLE
- SPILL

If you were to issue the following command you would obtain a report detailing the data sets that have been backed up to daily volumes:

```
HSEND REPORT DAILY FUNCTION(BACKUP) ODS(MHLRES5.BACKUP.REPORT)
```

The output is similar to that produced previously but restricted to the backup function. Some statistics are always reported, and these will appear before the requested function report.

10. **How do I list the statistics related to volume activity?**

For all volumes that DFSMShsm manages, you issue the following command:

```
HSEND REPORT VOLUMES FUNCTION ODS(MHLRES5.VOLUME.STATS)
```

The output that is written to the data set looks like this:

```
 DFSMSHSM STATISTICS REPORT ------- AT 18:16:41 ON 98/08/03 FOR SYSTEM=SC54

 VOLUME STATISTICS REPORT FOR VOLUME HSM14B

 UNIT TYPE = 3390    , HSM VOLUME TYPE = PRIMARY
 MIGRATED DATA SETS BY VOLUME REQUEST=000000, DATA SET MIGRATIONS BY DATA SET REQUEST=000000, DATA SETS PROCESSED=  N/A
          MINIMUM AGE = 000, TOTAL TRACKS = 00000334, FREE TRACKS = 00049751, FRAGMENT INDEX = .002
 VOLUME DUMP=  NOT DONE;  DUMP COPIES=000000  REQUESTED, FAILED=000000
 VOLUME RESTORE=  NOT DONE;  DATASET RESTORES=000000  REQUESTED, FAILED=000000

                     NUMBER  ------READ--------  -----WRITTEN------  ------REQUESTS---- AVERAGE ------AVERAGE TIME-------
    HSM FUNCTION    DATASETS TRK/BLK    K-BYTES  TRK/BLK    K-BYTES  SYSTEM USER FAILED   AGE   QUEUED WAIT PROCESS TOTAL

 MIGRATION
  PRIMARY - LEVEL 1 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
  SUBSEQUENT MIGS   0000000  0000000  000000000  0000000  000000000  000000 00000 00000  00000  0000  00000  00000  00000
  PRIMARY - LEVEL 2 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 RECALL
  LEVEL 1 - PRIMARY 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
  LEVEL 2 - PRIMARY 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 DELETE
  MIGRATE DATA SETS 0000000  0000000  000000000  0000000  000000000  000000 00000 00000  00000  0000  00000  00000  00000
  PRIMARY DATA SETS 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 BACKUP
  DAILY BACKUP      0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
  SUBSEQUENT BACKUP 0000000  0000000  000000000  0000000  000000000  000000 00000 00000  00000  0000  00000  00000  00000
  DELETE BACKUPS    0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 RECOVER
  BACKUP - PRIMARY  0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 RECYCLE
  BACKUP - SPILL    0000000  0000000             0000000             000000 00000 00000  00000  0000  00000  00000  00000
  MIG L2 - MIG L2   0000000  0000000             0000000             000000 00000 00000  00000  0000  00000  00000  00000


1
 VOLUME STATISTICS REPORT SUMMARY

 MIGRATED DATA SETS BY VOLUME REQUEST=000000, DATA SET MIGRATIONS BY DATA SET REQUEST=000000, DATA SETS PROCESSED=  N/A
          MINIMUM AGE = 000, TOTAL TRACKS = 00000334, FREE TRACKS = 00049751, FRAGMENT INDEX =  N/A
 FULL VOLUME DUMPS= 000000 REQUESTED, 00000 FAILED; DUMP COPIES= 000000 REQUESTED, 00000 FAILED
 FULL VOLUME RESTORES= 000000 REQUESTED, 00000 FAILED; DATASET RESTORES= 000000 REQUESTED, 00000 FAILED

                     NUMBER  ------READ--------  -----WRITTEN------  ------REQUESTS---- AVERAGE ------AVERAGE TIME-------
    HSM FUNCTION    DATASETS TRK/BLK    K-BYTES  TRK/BLK    K-BYTES  SYSTEM USER FAILED   AGE   QUEUED WAIT PROCESS TOTAL

 MIGRATION
  PRIMARY - LEVEL 1 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
  SUBSEQUENT MIGS   0000000  0000000  000000000  0000000  000000000  000000 00000 00000  00000  0000  00000  00000  00000
  PRIMARY - LEVEL 2 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 RECALL
  LEVEL 1 - PRIMARY 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
  LEVEL 2 - PRIMARY 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 DELETE
```

```
 MIGRATE DATA SETS 0000000  0000000  000000000  0000000  000000000  000000 00000 00000  00000  0000  00000  00000  00000
 PRIMARY DATA SETS 0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
BACKUP
 DAILY BACKUP      0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
 SUBSEQUENT BACKUP 0000000  0000000  000000000  0000000  000000000  000000 00000 00000  00000  0000  00000  00000  00000
 DELETE BACKUPS    0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
RECOVER
 BACKUP - PRIMARY  0000000    N/A       N/A       N/A       N/A     000000 00000 00000  00000  0000  00000  00000  00000
RECYCLE
 BACKUP - SPILL    0000000  0000000             0000000            000000 00000 00000  00000  0000  00000  00000  00000
 MIG L2 - MIG L2   0000000  0000000             0000000            000000 00000 00000  00000  0000  00000  00000  00000
```

To specify that you want the statistics for a specific volume, use the following command:

```
HSEND REPORT VOLUMES(HSM14B) FUNCTION ODS(MHLRES5.VOL14B.STATS)
```

The output that is generated this time includes only statistics for the particular volume that you specify. A summary report is not generated for all volumes that DFSMShsm manages.

We have not shown all the REPORT command parameters. The information that is available within the DSRs VSRs is enough to give you a good idea of the activity. You may be able to spot patterns for particular days and functions, or identify unusually high activity against specific volumes and take some action to spread the work more evenly.

5.4 Using DCOLLECT

Another method of gathering data related to space utilization and capacity planning is to use DCOLLECT, an AMS function run in a batch environment. With DCOLLECT you can obtain information about:

- Active data sets
- Volumes
- Migrated data sets
- Backup data sets
- DASD capacity planning
- Tape capacity planning

For a detailed description of DCOLLECT, see *DFSMS/MVS Access Method Services for ICF*.

In this section we show how you can produce an output data set that can later be used as input to a report creation package.

1. **How do I invoke the DCOLLECT function?**

   DCOLLECT can be called in three ways:

   - As part of user JCL, using IDCAMS DCOLLECT
   - By directly invoking the ARCUTIL utility, using JCL
   - By invoking a user-written program that accesses the ARCUTIL module

2. **What is ARCUTIL?**

   ARCUTIL is a DFSMShsm-related program that you can use to capture a copy of DFSMShsm-specific information.

3. **What is the relationship between ARCUTIL and DCOLLECT?**

   If DFSMShsm information is requested from within a DCOLLECT batch job, the ARCUTIL load module is called, almost transparent to the end user. The parameters, although specified in a different place, are the same as you would use on the DCOLLECT SYSIN DD card.

4. **What does the JCL look like for DCOLLECT and ARCUTIL?**

   The sample DCOLLECT JCL shown in Figure 72 calls ARCUTIL under the covers:

```
//DCOLLECT   JOB   ,'DCOLLECT RUN',CLASS=Z,MSGCLASS=H,REGION=4M
//*
//STEP1   EXEC   PGM=IDCAMS
//*
//SYSPRINT  DD  SYSOUT=*
//ARCSNAP   DD  SYSOUT=*
//MCDS      DD  DSN=HSM.MCDS,DISP=SHR
//BCDS      DD  DSN=HSM.BCDS,DISP=SHR
//DCOUT     DD  DSN=MHLRES5.DCOLLECT.OUTPUT,
//               DISP=(NEW,CATLG,DELETE),
//               SPACE=(1,(859,429)),AVGREC=K,
//               DSORG=PS,RECFM=VB,LRECL=264
//SYSIN     DD  *
     DCOLLECT -
           OUTFILE(DCOUT) -
           NODATAINFO -
           NOVOLUMEINFO -
           MIGRATEDATA -
           BACKUPDATA -
           CAPPLANDATA -
           MIGRSNAPERR

/* END OF DCOLLECT COMMAND
```

Figure 72. Sample DCOLLECT JCL to Call ARCUTIL

If you want to invoke ARCUTIL directly, using the same DCOLLECT specified parameters to capture DFSMShsm information, use the JCL in Figure 73.

```
 frame=box place=inline scale='1.0' width=column
//JOB2    JOB 'ARCUTIL RUN',REGION=4M
//STEP2   EXEC PGM=ARCUTIL,PARM='DCOLLECT MIGD CAPD BACD MSERR'
//ARCSNAP SYSOUT=*
//ARCTEST SYSOUT=*
//ARCDATA DD DSN=MY.COLLECT.DATA,DISP=(,CATLG),
//           SPACE=(CYL,(5,10)),UNIT=SYSDA
//MCDS    DD DSN=HSM.MCDS,DISP=SHR
//BCDS    DD DSN=HSM.BCDS,DISP=SHR
```

Figure 73. Sample ARCUTIL JCL

This JCL gathers exactly the same information as the IDCAMS DCOLLECT run, but this time you are directly invoking ARCUTIL.

The DCOLLECT parameters are specified on the EXEC PARM statement. DCOLLECT must be the first command specified and can be followed by any of the other DCOLLECT keywords.

5. **Can I call DCOLLECT in any other way?**

   Yes. You can use ISMF to build a DCOLLECT job that will execute under IDCAMS.

   From the ISMF Primary Option Menu, select option C to get to Data Collection Entry panel (Figure 56).

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                          DATA COLLECTION ENTRY PANEL             Page 1 of 2
Command ===>

Select Data Collection options:
  Data Set Information  . . . Y           (Y or N; Y requires volume(s) or
  Volume Information  . . . . Y           storage group(s) on next page)
  Migration Data  . . . . . . Y           (Y or N)
  Backup Data . . . . . . . . Y           (Y or N)
  Capacity Planning Data  . . Y           (Y or N)
  SMS Data  . . . . . . . . . Y           (Y or N)
Specify Output Data Set:                  (1 to 44 Characters)
  Data Set Name . . . . . . . DCOLLECT
  Optional Password . . . . .             (Ignored if SMS-managed data set)
  Replace Contents  . . . . . N           (Y or N)
  Number of Data Sets . . . . 1           (1 to 99999999; new data set only)
Specify Input Data Set:                   (1 to 44 Characters)
  Migration Data Set Name . . 'HSM.MCDS'
  Backup Data Set Name  . . . 'HSM.BCDS'
  CDS Name  . . . . . . . . . 'ACTIVE'
```

Figure 74. Data Collection Entry Panel (Page 1 of 2)

From this panel you can make selections based on the information that you want to capture. You can also capture data from your active SMS environment.

6. **Is the output that is generated in the format of a report?**

   No. These options generate a sequential data set that can then be used as input to a reporting mechanism of your choice.

7. **Which reporting mechanisms can I use?**

   You have number of choices. For example, you can:

   - Access the records of the DCOLLECT data set with a user-written program
   - Use IBM DFSORT

8. **Are there any examples of the reports that I can use and modify?**

   Yes. Samples are provided to allow you to create a simple report. Refer to the following for more information:

   - In PDS HSM.SAMPLE.TOOL, member DCOLREXX provides a sample REXX EXEC.
   - The *dpAM R2 Collect Feature Customization and Reference* has some simple reports that you can use.

- The *DFSORT Application Programming Guide* has are some samples that show you how to use ICETOOL to create reports.

9. **Where can I get more information about the DCOLLECT functions and the record layout of the sequential data set that is created?**

   For a comprehensive description of DCOLLECT, see *DFSMS/MVS V1R4 Access Method Services for ICF*.

5.5 Using the HSM Monitor/Tuner

The Challenge

You would like to monitor and control your DFSMShsm systems from a single workstation. You are wondering whether your automatic space management has been running successfully, and, if it has not, you are trying to figure out what to do. You want to detect potential DFSMShsm problems on the basis of thresholds you set and have corrective action taken automatically.

The Solution

The DFSMS Optimizer Version 1 Release 2 became available in February 1998 and has been enhanced to satisfy your requirements. The DFSMS Optimizer uses historical and real-time data to provide an overall picture of data usage on each system in a sysplex environment. It enables you to understand how you are utilizing storage today. With this information, you can make informed decisions about how to manage storage in the future.

The HSM Monitor/Tuner component of the DFSMS Optimizer keeps you informed about the status of DFSMShsm, provides control and tuning capability, allows selective playback of DFSMShsm activity, and enables you to terminate active DFSMShsm tasks through the graphical user interface (GUI).

Improvements in the DFSMS Optimizer Version 1 Release 2 have been made primarily to the HSM Monitor/Tuner and the Charting Facility. Here are the enhancements introduced in this release of the DFSMS Optimizer:

- Improved event recording is incorporated into the new release to gather information from DFSMShsm on a real-time basis. The information is displayed in new workstation panels. Furthermore, the information is used to drive enhanced event triggering from the host.
- The HSM Monitor/Tuner workstation component and the Charting Facility provide multiple platform support for Windows 95, Windows NT, and OS/2.
- The command feature of the HSM Monitor/Tuner has been improved so that commands issued at the workstation receive a response at the workstation. A new feature enables termination of selected DFSMShsm tasks from the workstation.
- The HSM Monitor/Tuner can anticipate and automate corrective actions through customizable triggers that invoke host-resident user-written REXX routines.
- New logging functions provide historical review of DFSMShsm activity by means of a workstation playback file.
- The Charting Faciltiy provides you with a GUI to download the DFSMS Optimizer chart files and view and print color charts.

So, Where to Start?

The redbook entitled *DFSMS Optimizer: The New HSM Monitor/Tuner* (SG24-5248), is written for both new and experienced users of the DFSMS Optimizer at the Version 1 Release 2 level. It provides planning, implementation, and usage guidance for customers who want to exploit the features of the DFSMS Optimizer Version 1 Release 2 HSM Monitor/Tuner and Charting Facility. It uses a step-by-step approach, taking you through the sequence of tasks necessary to get the DFSMS Optimizer Monitor/Tuner and Charting Facility up and running on your system. Installation on the MVS host system and the supported client platforms (OS/2, Windows 95, and Windows NT) is documented. Migration considerations to the DFSMS Optimizer Version 1 Release 2 from a previous level of the product are discussed. The book contains detailed descriptions about how to use the HSM Monitor/Tuner from the supported platforms to monitor and control DFSMShsm. Many useful hints and tips that will help you complete the installation and customization tasks outlined in the book are included.

New users of the DFSMS Optimizer who have some experience with DFSMShsm will find valuable information about installing, customizing, and exploiting the functions of the DFSMS Optimizer Version 1 Release 2 workstation component (DFSMS Optimizer Monitor/Tuner and Charting Facility). Experienced DFSMS Optimizer users will find useful information about migrating to this release of the product, using the new functions of the HSM Monitor/Tuner and REXX routines to automate responses to DFSMShsm events.

DFSMS Optimizer: The New Monitor/Tuner does not include information about how to build the DFSMS Optimizer database. Refer to the *DFSMS Optimizer Usage Guide*, SG24-2235, for a description of the tasks required to set up your database and collect data for use by the DFSMS Optimizer.

Chapter 6. DFSMShsm Tape Management

How DFSMShsm tape management is performed at your site is determined by a number of factors, such as:

- How DFSMShsm interacts with DFSMSrmm and other tape management systems
- Tape pool types that you can use
- Processing backup and migration tapes
- How tapes are marked at the end of a function
- Allowing SMS to manage your tape media

We describe the DFSMShsm tape requirements in a tape management system and a non-tape management system. We highlight the security measures that should be implemented for DFSMShsm tape processing with DFSMSrmm and describe enhancements that have been provided for tape processing. Optimum tape utilization is described with the use of tape emulation capabilities. Customization of the DFSMSrmm VRSs is provided as are examples for vault movement to your vault or to offsite data storage.

6.1 DFSMShsm Tape Media

DFSMShsm tape processing is supported by both single-file cartridge and multifile tape reel and requires both to be in standard label format.

Note: With DFSMS/MVS Version 1 Release 5, DFSMShsm will not create multiple file reels, like 3420s. DFSMShsm will continue to read data written in multifile format.

Single-file cartridges are, as the term implies, written in single-file format on cartridges from type 3480 up to the current cartridges available. It provides performance advantages in I/O and system serialization reduction, as well as recovery advantages through AUDIT MEDIACONTROLS and TAPECOPY functions as they are only functional against single-file format cartridges.

Multifile tape reels are associated with devices before type 3480. They require a standard label data set for each user data set that is written to them, which must also have a file sequence number for each data set ranging from 1 to 9999. Once 9999 is reached, DFSMShsm marks the tape full and the remaining capacity on the tape is unusable.

6.2 Global and Specific Scratch Pools

When referring to a global scratch pool, you are referring to a pool of tapes that are not defined to DFSMShsm. When DFSMShsm requests a scratch tape to be mounted, it creates a record in the OCDS. The record is stored by DFSMShsm in the OCDS for the duration of its use by DFSMShsm, after which it is deleted, and the tape returns to the global scratch pool. When a scratch tape from the global scratch pool is used by DFSMShsm, its status is changed to private. Global scratch pools work well with tape management systems like DFSMSrmm.

A specific scratch pool consists of tapes that are predefined to DFSMShsm and can be used only by DFSMShsm. These tapes are manually added to DFSMShsm with the ADDVOL command. For example, to add tape POK001 for

use by DFSMShsm started procedure HSM54 as a backup tape in an automatic tape library (ATL), issue the following command:

```
F HSM54,ADDVOL POK001 BACKUP UNIT(3590-1)
```

We recommend that you use global scratch pools, as discussed in 1.4, “DFSMShsm Tape Processing” on page 5. Output processing in a global scratch pool environment calls for any scratch cartridge that significantly reduces the mount wait time, especially when using devices with automatic cartridge loaders or in an automated tape library.

6.3 DFSMShsm Tape Management System Interaction

DFSMShsm communicates with other components of your MVS system through exits, including your tape management system. ARCTVEXT is the installationwide exit that allows communication between DFSMShsm and your tape management system, with the exception of DFSMSrmm. ARCTVEXT is specified in the ARCCMDxx member of PARMLIB and is enabled with the SETSYS EXITON parameter.

With DFSMS/MVS Version 1 Release 4, the EDGTVEXT general-use programming interface is called by DFSMShsm to communicate with DFSMSrmm. DFSMShsm automatically calls DFSMSrmm to process tapes that are to be returned to the DFSMShsm tape pool or deleted from DFSMShsm.

1. **How do I tell DFSMShsm to enable the ARCTVEXT installationwide exit?**

   To tell DFSMShsm to enable the ARCTVEXT installationwide, add the following SETSYS command in the ARCCMDxx PARMLIB member:

```
SETSYS EXITON(ARCTVEXT)
```

2. **How do I tell DFSMShsm to use the EDGTVEXT programming interface to communicate with DFSMSrmm?**

   The interface between DFSMShsm and DFSMSrmm is automatically in place when you install DFSMS/MVS Version 1 Release 4.

6.3.1 Duplex Tape

For disaster recovery purposes DFSMShsm provides two methods of creating disaster backup copies of tape volumes:

- The DFSMShsm TAPECOPY command
- The duplex tape option

The tapecopy function of DFSMShsm is discussed in 6.3.2, “TAPECOPY” on page 194.

The duplex tape option is an alternative to TAPECOPY processing and is applicable to backup and migration volumes.

With duplex tape, DFSMShsm creates two tapes simultaneously. You can make a backup copy or migrate data to tape on two different tape drives at the same

time. If you have an adequate number of tape drives, the duplex tape option is recommended over TAPECOPY.

1. **How do I invoke the duplex tape option?**

   The option to duplex either backup or migration or both is determined by the following SETSYS parameter:

```
SETSYS DUPLEX(BACKUP(Y) MIGRATION(Y))
```

   In this case duplexing is turned on for both backup and migration and means that two tapes are created concurrently. One is designated the original, with the intention of being kept onsite, and the other the alternate, with the intention of being taken offsite or created in a remote location.

2. **What are the tape data set names of the tape copies that are created?**

   The pair of tapes maintain the original versus alternate distinction. The original tape volume will have one of the following data set names, depending on under which function it was created:

```
prefix.HMIGTAPE.DATASET
prefix.BACKTAPE.DATASET
```

   The alternate tape volumes will have names of the following format:

```
prefix.COPY.HMIGTAPE.DATASET
prefix.COPY.BACKTAPE.DATASET
```

3. **Are these data set name formats compatible with tapes created by the TAPECOPY command?**

   Yes. This format allows tapes created by the duplex tape option to be compatible with those created by TAPECOPY.

4. **Must the original and alternate copies be the same tape geometry?**

   Yes. Although it would be possible in an SMS-managed tape library to direct the alternate tape volume to a different tape library with a different geometry, this is not recommended or supported. The intention of the duplex tape option is to make sure that you have an alternate copy that is exactly the same in all respects as the original.

5. **Are there any special considerations that I need to take into account before switching on duplexing?**

   As two copies are going to be created concurrently, you will need at least two output devices for each backup or migration task that is going to run. This includes the RECYCLE function of DFSMShsm as well. If duplex is in use for a function, each recycle task needs three tape drives: one for input and two for output.

6.3.2 TAPECOPY

1. **What does the TAPECOPY command allow me to do?**

   As its name suggests, the DFSMShsm TAPECOPY command allows you to make copies of your DFSMShsm-owned cartridge tapes. The copies of the original tape are known as alternates and the terminology is similar to that used in tape duplexing.

2. **Do I need to ADDVOL the tape volumes before I make a copy?**

   No. The alternate volumes start as scratch volumes. This prevents backup, dump, or ML2 tapes from being used as alternate volumes.

3. **Which commands can I use to make tape copies?**

   TAPECOPY allows you to copy categories of tapes:

```
TAPECOPY BACKUP

TAPECOPY MIGRATIONLEVEL2

TAPECOPY ALL
```

   These commands cause copies to be taken of full tapes that do not already have an alternate copy.

4. **If my tapes are not already full, what do I need to do?**

   If you already have the following specified in your ARCCMDxx PARMLIB member:

```
SETSYS
  PARTIALTAPE(MARKFULL)
```

   there is nothing else that you need to do as you are asking DFSMShsm to mark partially filled tapes full at the end of each task. If, instead of MARKFULL, you have coded REUSE in your ARCCMDxx PARMLIB member, you have to issue the SETSYS command as above. After a few backup and migration cycles, the volumes will be marked as full.

   Alternatively you could issue one of the following commands to mark the volumes as full:

```
DELVOL volser BACKUP(MARKFULL)

DELVOL volser MIGRATION(MARKFULL)
```

5. **Will the DELVOL cause the volume to be deleted from DFSMShsm′s inventory?**

   No. When the MARKFULL parameter is specified, the only processing that takes place is to mark the volume as full. A delete of the volume does not take place.

6. **I thought that ML2 tape volumes were not marked as full at the end of data set migration. What will happen then?**

ML2 tape volumes are not marked as full at the end of data set migration. There is an exception for ML2 tapes, even in a MARKFULL environment. ML2 tapes are not marked as full after data set migration for a simple reason: If, for example, command migration caused one data set to migrate to tape, that tape would be an ML2 tape for just one data set. This is not smart tape utilization, so ML2 tapes remain as partially filled until a TAPECOPY command is issued for that volume. TAPECOPY, in a MARKFULL environment, marks ML2 volumes as full and makes the copy.

7. **Can I make copies of individual volumes?**

   Yes. Use a version of the following command:

```
TAPECOPY ORIGINALVOLUMES(ovol1,ovol2,ovoln)
```

   This command causes a copy of the volume or volumes that you specify with the ovol parameters.

8. **What are the tape data set names of the tape copies that are created?**

   The original tape volume will have one of the following data set names depending on the function under which it was created:

```
prefix.HMIGTAPE.DATASET
prefix.BACKTAPE.DATASET
```

   The alternate tape volumes will have names of the following format:

```
prefix.COPY.HMIGTAPE.DATASET
prefix.COPY.BACKTAPE.DATASET
```

9. **How does DFSMShsm keep track of the alternate volume?**

   The alternate volume serial is recorded in the TTOC record of the original tape.

10. **Over time if I explicitly copy an original volume to a number of alternate volumes, which volume will DFSMShsm use as the alternate?**

    DFSMShsm records only the most recent copy of the volume in the TTOC record.

    An ARC0436I message is issued each time you make a copy when one already exists.

11. **How can I obtain a list of tape volumes that are marked full and do not have alternate tape volumes?**

    Issue the following command to get a list of those volumes:

```
LIST TTOC SELECT(NOALTERNATEVOLUME)
```

12. **Is it possible to use the volume list that is returned as input to a TAPECOPY command?**

Yes. Use a modification of the following command:

```
TAPECOPY INDATASET(volcopy.list.dsname)
```

to build the original volume list by using an edited version of the output from the LIST command. For more information about specifying this parameter, refer to the *DFSMShsm Storage Administration Reference*, SH21-1078.

6.3.3 TAPEREPL

1. **How do I replace an original tape volume that is lost or damaged?**

   Use the TAPEREPL command to replace an original volume.

2. **How do I use the TAPEREPL command to replace an original volume?**

   If your original volume with a volser of 666666 has become lost or damaged, and assuming that an alternate exists, you would issue the following DFSMShsm command:

```
TAPEREPL ORIGINALVOLUMES(666666)
```

   Once this command has successfully completed, all references to the original tape have been replaced by references to what was its alternate tape, and the original tape is disposed of as specified in the TAPEDELETION option.

3. **Do I still have an alternate tape volume?**

   No. You must issue the TAPECOPY command to make a new copy of the volume. If, for example, the volume that replaced volser 666666 had a volser of 111111 (that is to say, 111111 was the alternate), you would have to issue the following command:

```
TAPECOPY ORIGINALVOLUMES(111111)
```

   Once this command completes, you have a new alternate should it be required.

6.4 DFSMSrmm

DFSMSrmm manages all tape media, such as cartridge system tapes and 3420 reels, as well as other removable media you define to it. For example, DFSMSrmm can record the shelf location for optical disks and track their vital record status; it does not manage the objects on optical disks.

DFSMSrmm can manage removable media libraries, which incorporate all libraries, including:

- System-managed tape libraries, that is, the automated IBM 3494 Tape Library Dataserver, the IBM 3495 Tape Library Dataserver, and the manual IBM 3495 M10 Tape Library Dataserver
- Non-system-managed tape libraries, or traditional tape libraries

- Offsite storage media locations

6.4.1 Interaction between DFSMShsm and DFSMSrmm

In a DFSMSrmm global pool environment, the only tapes that DFSMShsm knows about are the tapes that it has used and have not yet expired. DFSMShsm does not maintain its own pool of scratch tapes and knows nothing of any tapes outside its control.

Note: There are exceptions to this, as DFSMShsm can manage its own specific tapes in coexistence with a global scratch pool evironment. This is not recommended, however.

When DFSMShsm calls for a scratch tape, DFSMSrmm honors the request, and DFSMShsm uses the tape to write to. DFSMSrmm creates a master control record, and DFSMShsm creates a record in the DFSMShsm OCDS.

1. How does DFSMSrmm retain a tape?

The tape can be retained in two ways. It can be retained by DFSMShsm passing an expiration date parameter to DFSMSrmm with a value of 99365 in the ARCCMDxx member of PARMLIB. For example:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(4) -
       BACKUPDEVICECATEGORY(TAPE(PARALLEL UNITNAME(3490) -
       EXPDT(99365)))
```

DFSMSrmm acknowledges that DFSMShsm has control of the tape and does not manage it until DFSMShsm releases it, or until the tape is deleted with the DELVOL command. For this to work, you must specify MAXRETPD(NOLIMIT) in the DFSMSrmm PARMLIB member EDGRMMxx:

```
/*************************************************************/
/* DFSMS/MVS 1.4 DFSMSrmm                                    */
/*Proprietary V2 Statement                                   */
/*Licensed Materials - Property Of IBM                       */
/*"Restricted Materials of IBM"                              */
/*5695-DF1 (C) COPYRIGHT 1979,1993  IBM CORP.                */
/*End Proprietary V2 Statement                               */
/*************************************************************/
/*  RMM (JDZ1150) IVP.  IVP EDGRMM00 PARMLIB MEMBER          */
/*  COPY THIS MEMBER TO 'SYS1.PARMLIB' AS MEMBER EDGRMM54.   */
/*  CHANGE THE MASTER AND JOURNAL FILE NAMES IF NECESSARY    */
/*-----------------------------------------------------------*/
 OPTION DSNAME(RMM54.CONTROL.DSET)    /* CHANGE MASTER FILE NAME */  -
        JRNLNAME(RMM54.JOURNAL.DSET)  /* CHANGE JOURNAL NAME     */  -
        RETPD(0)                      /* SET DEFAULT RETPD       */  -
        CATRETPD(0)                   /* SET CATALOG RETPD       */  -
        UNCATALOG(Y)                                              -
        SATUPD(N)                                                 -
        CDSID(SC54TS)                                             -
        MAXRETPD(NOLIMIT)                                         -
        IPLDATE(Y)                                                -
        NOTIFY(Y)                                                 -
        DATEFORM(E)                                               -
        LINECOUNT(60)                                             -
        SCRATCHPROC(EDGXPROC)                                     -
        SMFAUD(253)                                               -
        SMFSEC(254)                                               -
        SEARCHLIMIT(100)                                          -
        SYSID(SC54)                                               -
        TLCSV1(N)                                                 -
        TPRACF(A)                                                 -
        OPMODE(P)
 VLPOOL PREFIX(POK*) TYPE(S) MEDIANAME(3490) RACF(Y) EXPDTCHECK(N) -
        DESCRIPTION('HSM54 TAPES')
```

The tape can also be retained by using DFSMSrmm VRSs.

DFSMSrmm provides policy management for movement and retention at the data set level. Every tape data set can have a policy, and each policy can specify movement as well as retention. The retention and movement policies you define to DFSMSrmm are known as VRSs. You use them to indicate how long and where you want to keep data sets or volumes. You also use them to define how volumes are to be moved among the libraries that DFSMSrmm supports and the storage locations defined for vital records and disaster recovery purposes.

6.4.2 Vital Record Specifications

VRSs, are used for managing your site's vaulting requirements for either onsite or offsite storage. This is not limited to DFSMShsm as many application batch jobs also create backups that require movement, either to your onsite vault or to an offsite location. VRSs allow you to specify data set name masks, required locations, and how to retain your tapes, in easy-to-follow panels that are delivered with DFSMSrmm. Here we look at some of the DFSMSrmm panels and discuss DFSMShsm considerations.

1. **How do I get into the DFSMSrmm VRS panels?**

You can access the Removable Media Manager panels through ISMF in your TSO/E session as shown in Figure 75 on page 199.

```
 Panel  Help
------------------------------------------------------------------------
                  ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.4
Enter Selection or Command ===>

Select one of the following options and press Enter:

0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Cri
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availabi
6  Storage Group             - Specify Volume Names and Free Space Thres
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Manag
C  Data Collection           - Process Data Collection Function
L  List                      - Perform Functions Against Saved ISMF List
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                      - Terminate ISMF
```

Figure 75. The ISMF Primary Option Menu

From the ISMF Primary Option Menu, select option R to access the Removable Media Manager (see Figure 76).

```
 Panel  Help
------------------------------------------------------------------------
                      REMOVABLE MEDIA MANAGER (DFSMSrmm)
Option ===>

0  OPTIONS       - Specify dialog options and defaults
1  USER          - General user facilities
2  LIBRARIAN     - Librarian functions
3  ADMINISTRATOR - Administrator functions
4  SUPPORT       - System support facilities
5  COMMANDS      - Full DFSMSrmm structured dialog
6  LOCAL         - Installation defined dialog
X  EXIT          - Exit DFSMSrmm Dialog

Enter selected option or END command.  For more info., enter HELP or PF




5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 76. Removable Media Manager Main Selection Panel

Enter option 3, Administrator, which takes you to the DFSMSrmm Administrator Menu shown in Figure 77 on page 200.

```
  Panel  Help
------------------------------------------------------------------------
                          DFSMSrmm Administrator Menu
Option ===>

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Display or Change volume information
2  OWNER     - Display or Change owner information
3  VRS       - Vital Record Specifications


Enter selected option or END command.  For more info., enter HELP or PF









5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 77. DFSMSrmm Administrator Menu

Select option 3, VRS, to enter the DFSMSrmm Vital Record Specification Menu as shown in Figure 78.

```
  Panel  Help
------------------------------------------------------------------------
                   DFSMSrmm Vital Record Specification Menu
Option ===>

0  OPTIONS   - Specify dialog options and defaults
1  DISPLAY   - Display a Vital Record Specification
2  ADD       - Add a Vital Record Specification
3  CHANGE    - Change a Vital Record Specification
4  DELETE    - Delete a Vital Record Specification
5  SEARCH    - Search for Vital Record Specification


Enter selected option or END command.  For more info., enter HELP or PF









5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 78. DFSMSrmm Vital Record Specification Menu

From here you can display, create, alter, delete, and perform searches for all of your vital records relating to DFSMShsm.

2. How do I create a VRS entry to have my DFSMShsm dump tapes sent offsite for 2 weeks?

To add a vital record specification, from the DFSMSrmm Vital Record Specification Menu, select option 2, Add, which takes you to the DFSMSrmm Add Vital Record Specification panel (Figure 79 on page 201). Enter the DFSMShsm volume dump data set name that your installation is using. We are using HSM as the high-level qualifier for our DFSMShsm environment, so HSM.DMP.** is our dump data set name profile.

```
  Panel  Help
------------------------------------------------------------------------
                 DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . . 'HSM.DMP.**'
       Job name mask . .

  Volume serial . .         ( May be generic )

  VRS name  . . . .
```

Figure 79. DFSMSrmm Add Vital Record Specification Panel

You are presented with the DFSMSrmm Add Data Set VRS panel (Figure 80) where you specify your requirements for onsite or offsite storage (for example, how to retain the tapes and the amount of time to retain the tapes).

```
  Panel   Help
-------------------------------------------------------------------------------
                          DFSMSrmm Add Data Set VRS
Command  ===>

Data set mask  : 'HSM.DMP.**'                                    GDG . . . NO
     Job name mask  :                                            ( Yes or No )
Description  . . DFSMShsm DUMP TAPES

Owner  . . . . . HSMTASK        Retention type . . DAYS
                                                  ( Cycles, Days or Refdays )
                                While cataloged . . . . . NO   ( Yes or No )
Priority . . . . 0              Until expired . . . . . . . NO   ( Yes or No )
Location . . . . LOCAL          ( HOME, SMS library name or storage location )
Delete date  . . 1999/365       ( YYYY/DDD )

Vital record count  . . . 99999   ( Total number of cycles/days )
Delay . . . . . . . . . . 0       ( Days before being moved to location )
Store number in location  14      ( Number of cycles/days for this location )

Next VRS name . . . . . .           ( Specify if multiple moves required )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 80. DFSMSrmm Add Data Set VRS Panel

Table 4 provides information about the fields in the DFSMSrmm Add Data Set VRS panel. For detailed information about the panel fields, refer to the *DFSMSrmm Implementation and Customization Guide*.

Table 4 (Page 1 of 2). Vital Record Specification Description Table

Field	Field Description
Data set mask	Specify the fully or partially qualified data set name.
Job name mask	This can be left blank because DFSMShsm creates the tapes.
GDG	DFSMShsm does not create GDGs, so enter NO.

Table 4 (Page 2 of 2). Vital Record Specification Description Table	
Field	**Field Description**
Description	Any valid description that is meaningful for the VRS
Owner	As DFSMShsm will be a large user of your tape resources, it should have its own DFSMSrmm owner created. In this case we created HSMTASK as the owner for all DFSMShsm tapes. *The DFSMSrmm Implementation and Customization Guide* describes how to create an owner.
Retention type	If you want to send your tapes offsite or to the vault for a specific period, for example, 14 days for your dump tapes, specify DAYS. This value will relate to the *Store number in location* field.
While cataloged	Specify NO for DFSMShsm data sets.
Until expired	Specify NO for DFSMShsm data sets.
Location	Valid entries are HOME, LOCAL, DISTANT, and REMOTE, but refer to the *DFSMSrmm Implementation and Customization Guide*.
Delete date	This field specifies the date when the VRS is to be deleted. DFSMSrmm uses a default value of 1999/365, which means never delete.
Vital record count	99999 indicates that all vital records are kept.
Delay	You can delay the movement of your offsite data. A zero indicates to move the tapes immediately; 1 indicates to move the tapes the following day.
Store number in location	This value indicates the number of days that any particular tape is to stay offsite. We used DAYS in the *Retention type* field so our tapes will stay offsite for 14 days in location LOCAL. (LOCAL can be any storage facility in the nearby area.)
Next VRS name	Use this field only if you are planning on staging your tape movements; for example, DISTANT, then REMOTE, then LOCAL.

3. How do I check my current VRS definitions to ensure I do not duplicate any?

You can search for all of your currently defined VRSs, using option 5, Search, on the DFSMSrmm Vital Records Specification Menu as shown in Figure 78 on page 200. This selection will display the DFSMSrmm Search VRSs panel (Figure 81 on page 203).

```
  Panel  Help
------------------------------------------------------------------------------
                            DFSMSrmm Search VRSs
Command ===>

Optionally specify one of:
Data set mask   **                                          GDG  . .
     Job name mask                                          ( Yes or No )
Volume serial
VRS name  . . .

Owner . . . . . *             Retention type . .
                                          ( Cycles, Days or Refdays )
                              While cataloged         ( Yes or No )
                              Until expired  . .      ( Yes or No )
Location . . . .              ( HOME, SMS Library Name or Storage location )
Next VRS name
Limit  . . . . . 99           ( Limit search to first nnnn VRSs.
                              Default = 10 )

The following line commands will be available:
C - Change VRS                    D - Delete VRS
O - Display VRS Owner             S - Display VRS
```

Figure 81. DFSMSrmm Search VRSs Panel

To display all current VRS definitions, enter asterisks in the *Data set mask* and *Owner* fields. Set *Limit* to a high number as the default is only 10.

4. **What kind of VRS should I create for daily backup tapes that are to remain onsite?**

   Use the example shown in Figure 82 to add your VRS for your daily DFSMShsm backup tapes. The way in which DFSMShsm dump tapes and DFSMShsm backup tapes are processed has changed somewhat.

```
  Panel  Help
------------------------------------------------------------------------------
                        DFSMSrmm Add Data Set VRS
Command ===>

Data set mask  : 'HSM.BACKTAPE.DATASET'                      GDG . . . NO
     Job name mask  :                                        ( Yes or No )
Description  . . HSM BACKUPS
Owner  . . . . . HSMTASK        Retention type . . CYCLES
                                          ( Cycles, Days or Refdays )
                                While cataloged . . . . . NO   ( Yes or No )
Priority . . . . 0              Until expired . . . . . . . NO   ( Yes or No )
Location . . . . HOME           ( HOME, SMS library name or storage location )
Delete date  . . 1999/365       ( YYYY/DDD )

Vital record count  . . . 99999   ( Total number of cycles/days )
Delay . . . . . . . . . . 0       ( Days before being moved to location )
Store number in location  99999   ( Number of cycles/days for this location )

Next VRS name . . . . . .           ( Specify if multiple moves required )

Press ENTER to CHANGE the VRS, or END command to CANCEL.
```

Figure 82. Defining a VRS for DFSMShsm Backup Tapes

In this example, we have changed three fields:

- Retention type
- Location
- Store number in location

Retention type has been changed to CYCLES because we want any tape using this VRS to continue to use it until it is either recycled, expires naturally, or is deleted.

Location can be changed to HOME if you do not want daily backup tapes sent offsite.

Store number in location is changed to 99999, which is the DFSMSrmm default. This works with the CYCLES selection to ensure that all tapes are retained in location HOME.

5. **Do I create a VRS for my migrated data and have it defined the same as my VRS for onsite backup?**

   Yes. Your site policy may vary in terms of how your backup and migration data is treated, but for onsite migrated data, it would be the same except for the data set mask.

6. **If I created a more generic VRS, would it create a problem?**

   No it would not. If you create a VRS with a data set name mask of HSM.DMP.** (which we have) and then later create another VRS with a data set name mask of HSM.**, DFSMSrmm will use the more specific VRS definition when possible.

6.4.3 RACF Considerations

We briefly review RACF considerations for DFSMShsm and DFSMSrmm interaction. For details on DFSMSrmm implementation, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-4932.

1. **Does DFSMShsm have to be authorized to any DFSMSrmm profiles?**

   Yes. DFSMShsm has its own user ID (see 2.3, “Providing Security for DFSMShsm Resources” on page 37), which must be authorized to the DFSMSrmm RACF Facility class profiles. Use this method:

   - Create a Facility class profile for the DFSMShsm user ID, which in our case is HSM54:

```
RDEFINE FACILITY STGADMIN.EDG.OWNER.HSM54
```

   - Next, permit HSM54 to this profile and to additional DFSMSrmm profiles, which should already be defined on a DFSMSrmm system:

```
PERMIT STGADMIN.EDG.OWNER.HSM54 CLASS(FACILITY) ID(HSM54) ACCESS(UPDATE)
PERMIT STGADMIN.EDG.RELEASE     CLASS(FACILITY) ID(HSM54) ACCESS(READ)
PERMIT STGADMIN.EDG.MASTER      CLASS(FACILITY) ID(HSM54) ACCESS(READ)
```

   Note: If the STGADMIN.EDG.RELEASE and STGADMIN.EDG.MASTER Facility class profiles are not defined, contact your security administrator before continuing. If you do not define these Facility

class profiles properly, severe cartridge RELEASE problems can occur in DFSMSrmm.

After you permit the DFSMShsm user ID to the Facility class profiles, refresh your RACF facility class profiles, using this command:

```
SETROPTS GENERIC(FACILITY) REFRESH
```

6.5 SMS-Managed Tape Libraries

An SMS-managed tape library is a named collection of tape storage groups, and tape subsystems and their associated tape cartridges. A tape library can be automated or manual.

An automated tape library (ATL) allows you to use system-managed storage for mountable tape cartridges. Mount requests for these cartridges are handled by a robotic tape cartridge handler.

A manual tape library (MTL) also allows you to use system-managed storage for mountable tape cartridges. Mount requests are handled by a human operator.

6.5.1 SMS-Managed Tape Library Definition

The implementation of SMS-managed tape libraries involves several tasks. These tasks are fully described in the *DFSMS/MVS V1R4 OAM PISA for Tape Libraries*. For a detailed description of the steps you must follow to implement an SMS-managed tape library, see the *IBM Magstar 3494 Tape Libraries: A Practical Guide*. You can access this redbook through the Internet at http://www.redbooks.ibm.com.

To implement DFSMShsm functions in an SMS-managed tape library, follow these steps:

- Determine which tape functions you want to process in a tape library.
- Set up a global scratch pool.
- Define or update a data class to compact tape library data.
- Define or update a storage class to enable a storage group.
- Define or update a storage group to associate tape devices with the library.
- Set up or update ACS routines to filter data sets to the library.
- Define or update the DFSMShsm tape environment in the ARCCMDxx PARMLIB member.

6.5.2 Determine Which Functions to Process in a Tape Library

Tape libraries can process any DFSMShsm tape functions. You must decide which DFSMShsm functions to process in a tape library. Each DFSMShsm function uses a unique tape data set name. An ACS routine can recognize the functions you want to process in an SMS-managed tape library by the data set names. Table 5 on page 206 shows the data set name patterns generated by DFSMShsm according to the function it is performing. You may use the data set name patterns as filters for ACS selection.

Table 5. DFSMShsm Tape Data Set Names and Unit Types Passed to the ACS Routines

DFSMShsm Function	Tape Data Set Names	Commands with unittype Restrictions
Backup to original	prefix.BACKTAPE.DATASET	SETSYS BACKUP(TAPE(unittype))
Backup to alternate	prefix.COPY.BACKTAPE.DATASET	
Recycle of backup tapes to original	prefix.BACKTAPE.DATASET	SETSYS RECYCLEOUTPUT(BACKUP(unittype))
Recycle of backup tapes to alternate	prefix.COPY.BACKTAPE.DATASET	
Migration to original	prefix.HMIGTAPE.DATASET	SETSYS TAPEMIGRATION(- DIRECT(TAPE(unitype)) | ML2TAPE(TAPE(unittype)) | NONE(ROUTETOTAPE(unittype))
Migration to alternate	prefix.COPY.HMIGTAPE.DATASET	
Recycle of migration tapes to original	prefix.HMIGTAPE.DATASET	SETSYS RECYCLEOUTPUT(MIGRATION(unittype))
Recycle of migration tapes to alternate	prefix.COPY.HMIGTAPE.DATASET	
Dump	prefix.DMP.dclass.Vvolser.Dyyddd.Tssmmhh	DEFINE DUMPCLASS(class UNIT(unitype))
Spill	prefix.BACKTAPE.DATASET	SETSYS SPILL(TAPE(unittype))
Tape copy of backup tapes	prefix.COPY.BACKTAPE.DATASET	TAPECOPY ALTERNATEUNITNAME(- unitype1,unittype2)
Tape copy of migration tapes	prefix.COPY.HMIGTAPE.DATASET	
CDS backup Datamover=HSM	uid.BCDS.BACKUP.Vnnnnnnn uid.MCDS.BACKUP.Vnnnnnnn uid.OCDS.BACKUP.Vnnnnnnn uid.JRNL.BACKUP.Vnnnnnnn	SETSYS CDSVERSIONBACKUP (BACKUPDEVICECATEGORY - (TAPE UNITNAME(unitype)))
CDS backup Datamover=DSS	uid.BCDS.BACKUP.Dnnnnnnn uid.MCDS.BACKUP.Dnnnnnnn uid.OCDS.BACKUP.Dnnnnnnn uid.JRNL.BACKUP.Dnnnnnnn	
ABARS processing	outputdatasetprefix.C.CccVnnnn outputdatasetprefix.D.CccVnnnn outputdatasetprefix.I.CccVnnnn outputdatasetprefix.O.CccVnnnn	ABACKUP agname UNIT(unittype)

6.5.3 Set Up a Global Scratch Pool

A global scratch pool is a repository of empty tapes for use by anyone. The tape volumes are not individually known by DFSMShsm while they are members of the scratch pool. When a scratch tape is mounted and written to by DFSMShsm, it becomes a private tape and is removed from the scratch pool. When tapes used by DFSMShsm no longer contain valid data, they are returned to the global scratch pool for use by anyone, and DFSMShsm removes all knowledge of their existence.

Global scratch pools are recommended because mount requests can be responded to more quickly and more easily than when tapes reside in a specific scratch pool. Using a global scratch pool enables easy exploitation of cartridge loaders (including cartridge loaders in tape-library-resident devices) and works well with tape management systems such as DFSMSrmm. Ensure that tapes

entered into an SMS-managed tape library global scratch pool are assigned a scratch status.

If you plan to use a global scratch pool, ensure that the following commands are present in your ARCCMDxx PARMLIB member.

```
SETSYS SELECTVOLUME(SCRATCH)
SETSYS TAPEDELETION(SCRATCHTAPE)
SETSYS PARTIALTAPE(REUSE)         /* MARKFULL if using VTS */
```

6.5.4 Setting Up the ACS Routines

When DFSMShsm requests a tape device allocation from the MVS operating system, ACS routines are invoked, and the data set name, unit type, and normal ACS variables are passed to the routines. The ACS routines use this information to provide a storage class and a tape storage group that are associated with a tape library.

You can tell SMS which DFSMShsm functions are being directed to SMS-managed tape by building filter lists that include the data set names generated by the specifics functions that you want to go to an SMS-managed tape library (see Table 5 on page 206).

If you are using duplex tape, you can tell SMS to route your DFSMShsm original tape allocations to one tape library and the alternate (duplex) tape allocations to a different tape library, which could be an offsite tape library that you could use in a disaster recovery situation. Figure 83 on page 208 shows how you can filter the original and the duplexed tape data set names. HSM is the prefix for our DFSMShsm tape data set names.

```
/*******************************************************************/

       FILTLIST &SMS_MANAGED_TAPE_DSN INCLUDE
       ('HSM.BACKTAPE.DATASET','HSM.HMIGTAPE.DATASET',
        HSM.DMP.*.V*.D*.T*)

/*******************************************************************/

       FILTLIST &HSM_OFFSITE_TAPE_DSN INCLUDE
       ('HSM.COPY.BACKTAPE.DATASET','HSM.COPY.HMIGTAPE.DATASET')

/*******************************************************************/

        FILTLIST &VALID_DASD_UNIT INCLUDE
        ('3380','3390','SYSDA','SYSALLDA','')

/*******************************************************************/

        FILTLIST &VALID_TAPE_UNIT INCLUDE
        ('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')

/*******************************************************************/
```

Figure 83. Filter Lists of Original and Alternate Tapes

This example shows two separate data set name filter lists for DFSMShsm tape data:

SMS_MANAGED_TAPE_DSN where we include the data set names that our DFSMShsm generates for our original backup, migration, and dump tapes

HSM_OFFSITE_TAPE_DSN where we include the duplex tapes

Figure 84 on page 209, Figure 85 on page 210, and Figure 86 on page 211 show examples of ACS routines that direct the DFSMShsm backup and migration tape data to SMS-managed tape libraries. We used the filter lists shown in Figure 83 in these four ACS routine samples.

```
PROC &DATACLAS

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
                      .
   (PUT FILTLISTS HERE)  .
                      .
/** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **/

    SELECT /* MAIN LOGIC BLOCKS */

        WHEN (&UNIT EQ &VALID_TAPE_UNIT)

            SELECT /* TAPE ALLOCATIONS */

 /*********************************************************************/
 /* TAILOR AND SET DATACLAS FOR SMS-MANAGED TAPE SUBTYPES IF THEY     */
 /* EXIST                                                             */
 /*********************************************************************/
 /* REPEAT THIS BLOCK FOR EACH SMS-MANAGED TAPE SUBTYPE THAT HAS A    */
 /* DIFFERENT DATA CLASS                                              */
 /*********************************************************************/

               WHEN (&DSN EQ &SMS_MANAGED_TAPE_DSN OR
                     &DSN EQ &HSM_OFFSITE_TAPE_DSN)
                   DO
                       SET &DATACLAS ='DC54236'
                       EXIT
                   END /* DO */

 /*********************************************************************/
 /* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT  */
 /* OR SMS-MANAGED TAPE SUBTYPES EXIST                                */
 /*********************************************************************/

            END /* TAPE ALLOCATION SELECT */

 /*********************************************************************/
 /* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR       */
 /* OPTICAL                                                           */
 /*********************************************************************/

        OTHERWISE /* NON-MANAGED DEVICES */

            DO
                SET &DATACLAS = ''
                EXIT
            END /* DO */

    END /* SELECT */

END /* PROC */
```

Figure 84. Sample Data Class ACS Routines for Migration, Backup, and Dump Tapes

Figure 84 shows that this ACS routine sets the DC54236 data class to our DFSMShsm tape data. This data class was defined to use MEDIA2 Enhanced capacity cartridge system tape (ECCST) (see 6.5.6, “Defining a Data Class” on page 213).

Figure 85 on page 210 shows sample code in the storage class ACS routine, which assigns a storage class SC54TAPE to DFSMShsm tape data sets.

```
PROC &STORCLAS

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
                        .
    (PUT FILTLIST HERE) .
                        .
/** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **/

    SELECT /* MAIN LOGIC BLOCKS */

 /*********************************************************************/
 /* TAILOR AND SET STORCLAS FOR SMS-MANAGED TAPE SUBTYPES IF THEY     */
 /* EXIST                                                             */
 /*********************************************************************/
 /* REPEAT THIS BLOCK FOR EACH SMS-MANAGED TAPE SUBTYPE THAT HAS A    */
 /* DIFFERENT STORAGE CLASS                                           */
 /*********************************************************************/

       WHEN (&UNIT EQ &VALID_TAPE_UNIT)

           SELECT /* TAPE ALLOCATIONS */

               WHEN (&DSN EQ &SMS_MANAGED_TAPE_DSN OR
                     &DSN EQ &HSM_OFFSITE_TAPE_DSN)

                   DO
                       SET &STORCLAS ='SC54TAPE'
                       EXIT
                   END /* DO */

 /*********************************************************************/
 /* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT  */
 /* OR SMS-MANAGED TAPE EXIST                                         */
 /*********************************************************************/

           END /* TAPE ALLOCATION SELECT */

 /*********************************************************************/
 /* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR       */
 /* OPTICAL                                                           */
 /*********************************************************************/

       OTHERWISE /* NON-MANAGED DEVICES */

           DO
               SET &STORCLAS = ''
               EXIT
           END /* DO */

   END /* SELECT */

END /* PROC */
```

Figure 85. Sample Storage Class ACS Routine for Migration, Backup, and Dump Tapes

Figure 86 on page 211 shows sample code for a storage group ACS routine. Dump, backup, and migration original tapes are allocated in the SG54TAPE storage group, where a local ATL is defined. Backup and migration alternate tapes are allocated in the SG54OFFS storage group, where an offsite tape library is defined.

SMS-managed tape data does not have to have a management class associated with it, so we do not assign any management class to DFSMShsm migration and backup tape data sets.

```
PROC &STORGRP

/*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*/
                    .
     (PUT FILTLIST HERE) .
                    .
/**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**/

      SELECT /* MAIN LOGIC BLOCKS */

 /*********************************************************************/
 /* TAILOR AND SET STORGRP FOR SMS-MANAGED TAPE SUBTYPES              */
 /*********************************************************************/
 /* REPEAT THIS BLOCK FOR EACH SMS-MANAGED TAPE SUBTYPES THAT HAS A   */
 /* DIFFERENT STORAGE GROUP                                           */
 /*********************************************************************/

        WHEN (&UNIT EQ &VALID_TAPE_UNIT)

            SELECT /* TAPE ALLOCATIONS */

                WHEN (&DSN EQ &SMS_MANAGED_TAPE_DSN)

                    DO
                        SET &STORGRP =’SG54TAPE’
                        EXIT
                    END /* DO */

                WHEN (&DSN EQ &HSM_OFFSITE_TAPE_DSN)

                    DO
                        SET &STORGRP =’SG54OFFS’
                        EXIT
                    END /* DO */

            END /* TAPE ALLOCATION SELECT */

               OTHERWISE /* UNEXPECTED DEVICES */

                    DO
                        SET &STORGRP = ’SG54HSM’
                        EXIT
                    END /* DO */

    END /* SELECT */

END /* PROC */
```

Figure 86. Sample Storage Group ACS Routine for Backup, Migration, and Dump Tapes

6.5.5 Defining a Storage Class

A data set can be SMS-managed only when a storage class is assigned to it. For tape data sets, specialized performance and availability services are not required.

You can define the storage class for tapes with defaults, because none of the attributes applies to SMS-managed tape. See Figure 87 for an example of a storage class definition for SMS-managed tape.

```
                              STORAGE CLASS DEFINE              Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS3
Storage Class Name  : SC54TAPE
To DEFINE Storage Class, Specify:
  Description ==> Storage Class for Tape Data Sets
              ==>
  Performance Objectives
   Direct Millisecond Response . . . .            (1 to 999 or blank)
   Direct Bias . . . . . .                        (R, W or blank)
   Sequential Millisecond Response . .            (1 to 999 or blank)
   Sequential Bias . . . . . .                    (R, W or blank)
   Initial Access Response Seconds . .            (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .            (0 to 999 or blank)
  Availability . . . . . . . . . . . N             (C, P ,S or N)
  Accessibility  . . . . . . . . . . N             (C, P ,S or N)
   Backup  . . . . . .                            (Y, N or Blank)
   Versioning  . . . . . .                        (Y, N or Blank)
```

```
                              STORAGE CLASS DEFINE              Page 2 of 2
 Command ===>

 SCDS Name . . . . . : SYS1.SC54.SCDS3
 Storage Class Name  : SC54TAPE

 To DEFINE Storage Class, Specify:

   Guaranteed Space . . . . . . . . . N          (Y or N)
   Guaranteed Synchronous Write . . . N          (Y or N)
   CF Cache Set Name  . . . . . .                (up to 8 chars or blank
   CF Direct Weight . . . . . .                  (1 to 11 or blank)
   CF Sequential Weight . . . . . .              (1 to 11 or blank)
```

Figure 87. Storage Class Define Panel

6.5.6 Defining a Data Class

The data class construct can provide the tape devices and media type information for tape data sets. The attributes you can specify in a data class construct are:

- The type of media to use
- Whether the data is to be compacted
- Recording technology (18 track, 36 track, or 128 track)

You may have different tape devices and different media types at your installation, so you may have to assign specific data classes for the selection of specific devices and tape media.

To define a data class, go to ISMF Primary Option Menu and select option 4, Data Class, as shown in Figure 88.

```
                    ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.4
Enter Selection or Command ===> 4

Select one of the following options and press Enter:

0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Criteria
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availability
6  Storage Group             - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
C  Data Collection           - Process Data Collection Function
L  List                      - Perform Functions Against Saved ISMF Lists
R  Removable Media Manager   - Perform Functions Against Removable Media
```

Figure 88. Accessing the Data Class Panels from the ISMF Primary Option Menu

On the Data Class Application Selection panel, select option 3, Define, to define the data classes you need. On Figure 89 on page 214 we are going to define a data class named DC54236.

```
                    DATA CLASS APPLICATION SELECTION
Command ===>

To perform Data Class Operations, Specify:
  CDS Name . . . . . 'SYS1.SC54.SCDS3'
                              (1 to 44 character data set name or 'Active' )
  Data Class Name  . . DC54236   (For Data Class List, fully or partially
                                  specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
```

Figure 89. Data Class Application Selection Panel

On page 2 of the Data Class Define panel (Figure 90) we define the data class attributes that are applicable to tape data sets.

```
                              DATA CLASS DEFINE                 Page 2 of 3
Command ===>

SCDS Name . . . : SYS1.SC54.SCDS3
Data Class Name : DC54236
To DEFINE Data Class, Specify:
  Retpd or Expdt  . . . . . .           (0 to 9999, YYYY/MM/DD or blank)
  Volume Count  . . . . . . . 1         (1 to 59 or blank)
    Add'l Volume Amount . . .           (P=Primary, S=Secondary or blank)
  Imbed . . . . . . . . . . .           (Y, N or blank)
  Replicate . . . . . . . . .           (Y, N or blank)
  CIsize Data . . . . . . . .           (1 to 32768 or blank)
  % Freespace CI  . . . . . .           (0 to 100 or blank)
              CA  . . . . . .           (0 to 100 or blank)
  Shareoptions Xregion  . . .           (1 to 4 or blank)
               Xsystem  . . .           (3, 4 or blank)
  Compaction  . . . . . . . . y         (Y, N or blank)
  Media Interchange
    Media Type  . . . . . . . 2         (1, 2, 3, 4 or blank)
    Recording Technology  . . 36        (18, 36, 128 or blank)
```

Figure 90. Data Class Define Panel (Page 2 of 3)

The Data Class Define panel fields that affect tape device and media selection are explained below.

Compaction specifies whether or not data compaction should be selected for data sets assigned to this data class. We recommend that you always set the compaction to YES, even if you plan to direct DFSMShsm output tapes to a virtual tape server (VTS).

Media Type specify the tape cartridge media used for data sets associated with this data class. The possible Media Type values are:

1 For MEDIA1 (cartridge system tape; CST)

2 For MEDIA2 (ECCST)

3 For MEDIA3 (high performance cartridge tape)

4 For MEDIA4 (extended high performance cartridge tape)

blank Cartridge type not specified

Recording Technology specifies the number of tracks on tape cartridges used for data sets associated with this data class. For MEDIA1 and MEDIA2 tapes, this value must be 36; for MEDIA3 and MEDIA4 tapes, it must be 128.

DFSMShsm is a good candidate to use large capacity tapes. If you plan to direct your DFSMShsm tape output to an SMS-managed library, we recommend using MEDIA2 or larger tape cartridges.

If you plan to direct your DFSMShsm tapes to a VTS, we recommend that you define the logical volumes as MEDIA2 tapes (ECCST) and set the TAPEUTILIZATION as:

```
SETSYS TAPEUTILIZATION(LIBRARYMIGRATION (PERCENTFULL(50))
SETSYS TAPEUTILIZATION(LIBRARYBACKUP (PERCENTFULL(100))
```

The 50% value for the PERCENTFULL subparameter reduces the recall time for the migrated data sets. You can leave the PERCENTFULL value of 100% to the backup, as it contains data that is not going to be read again, unless a RECOVER is needed.

6.5.7 Defining a Storage Group

Storage group type TAPE is provided to classify tape cartridges in SMS. A tape storage group consists of tape libraries and the tape cartridges associated with them.

To define a new storage group, go to the ISMF Primary Option Menu (see Figure 88 on page 213) and select option 6, Storage Group. This brings you to the Storage Group Application Selection panel (Figure 91 on page 216). In this example, we are going to define a tape storage group called SG54OFFS.

```
                    STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . . 'SYS1.SC54.SCDS3'
                            (1 to 44 character data set name or 'Active' )
  Storage Group Name  . . SG540FFS           (For Storage Group List, fully or
                                             partially specified or * for all)
  Storage Group Type  . . TAPE               (VIO, POOL, DUMMY, OBJECT, OBJECT
                                             BACKUP, or TAPE)

Select one of the following options  :
  2  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
```

Figure 91. Storage Group Application Selection Panel

Figure 92 shows the Tape Storage Group Define panel. On this panel we define the tape library names that are associated with this storage group. We specify the name of our offsite tape library, SG54OFFS.

```
                          TAPE STORAGE GROUP DEFINE
Command ===>

SCDS Name . . . . . : SYS1.SC54.SCDS3
Storage Group Name  : SG540FFS

To DEFINE Storage Group, Specify:

  Description ==> OFFSITE TAPE LIBRARY
              ==>

  Library Names  (1 to 8 characters each):
    ===> ATL0FFS   ===>           ===>           ===>
    ===>           ===>           ===>           ===>


  DEFINE SMS Storage Group Status . .... Y  (Y or N)
```

Figure 92. Tape Storage Group Define Panel

Figure 93 on page 217 shows the panel that results from specifying Y for the DEFINE SMS Storage Group Status field in Figure 92. This panel enables you to specify the storage group status for each system in the same SMSplex. In our example, systems SC54 and SC66 are enabled to this storage group. SC54 is our primary DFSMShsm system; SC66 is our secondary processor.

```
                      SMS STORAGE GROUP STATUS DEFINE          Page 1 of 2
Command ===>

SCDS Name  . . . . : SYS1.SC54.SCDS3
Storage Group Name : SG54OFFS
Storage Group Type : TAPE
To DEFINE Storage Group System/                              ( Possible SMS SG
Sys Group Status, Specify:                                 Status for each:
                                                           - Pool SG Type
 System/Sys      SMS SG    System/Sys      SMS SG            NOTCON, ENABLE
 Group Name      Status    Group Name      Status            DISALL, DISNEW
 ----------      ------    ----------      ------            QUIALL, QUINEW
 SC54       ===> ENABLE    SC66       ===> ENABLE       - Tape SG Type
            ===>                      ===>                NOTCON, ENABLE,
            ===>                      ===>                DISALL, DISNEW )
            ===>                      ===>
            ===>                      ===>              * SYS GROUP = sysplex
            ===>                      ===>                minus Systems in the
            ===>                      ===>                Sysplex explicitly
            ===>                      ===>                defined in the SCDS
```

Figure 93. SMS Storage Group Status Define Panel

6.5.8 Define the Tape Library in ARCCMDxx

Figure 94 shows the sample SETSYS commands that define a typical SMS-managed tape library environment.

```
/*********************************************************************/
/*  SETSYS COMMANDS IN THE ARCCMDXX PARMLIB MEMBER THAT DEFINE THE    */
/*  DFSMSHSM ENVIRONMENT FOR AN SMS-MANAGED TAPE LIBRARY.             */
/*********************************************************************/
/*
SETSYS DUPLEX(BACKUP MIGRATION)
SETSYS SELECTVOLUME(SCRATCH)
SETSYS PARTIALTAPE(REUSE)
SETSYS TAPEDELETION(SCRATCHTAPE)
SETSYS BACKUP(TAPE)
SETSYS TAPEMIGRATION(ML2TAPE)
SETSYS TAPESECURITY(RACFINCLUDE)
SETSYS EXITON(TV)
DEFINE DUMPCLASS(ATLHSM NODATASETRESTORE NORESET AUTOREUSE)
/*
```

Figure 94. Sample SETSYS ATL Definition Commands in ARCCMDxx

SETSYS DUPLEX causes DFSMShsm to create two tapes concurrently during migration or backup processing: the original and its alternate.

SETSYS SELECTVOLUME(SCRATCH) specifies that DFSMShsm select scratch tapes from the global scratch pool.

SETSYS PARTIALTAPE(REUSE) specifies that DFSMShsm try to reuse ML2 and backup tapes that are partially filled with DFSMShsm data.

SETSYS TAPEDELETION(SCRATCHTAPE) specifies that recycled migration and backup tapes, along with expired dump tapes, are going to be returned to a global scratch pool.

SETSYS BACKUP(TAPE) specifies that backup processing is to tape devices.

SETSYS TAPEMIGRATION(ML2TAPE) specifies that ML2 media is tape.

SETSYS TAPESECURITY(RACFINCLUDE) specifies that DFSMShsm protects its tape volumes by adding them to a RACF tape volume set of DFSMShsm.

SETSYS EXITON(TV) activates the DFSMShsm tape volume exit ARCTVEXT. You must activate this exit if you are not running DFSMSrmm Version 1 Release 4. ARCTVEXT is used to tell the tape management system that DFSMShsm has released ownership of a DFSMShsm tape.

DEFINE DUMPCLASS defines the parameters for dump processing. The AUTOREUSE parameter is especially useful for tape libraries; it causes the dump tapes, when expired, to return to the global scratch pool for reuse.

6.5.9 Using RECYCLE to Move Data into the ATL

DFSMShsm starts writing data to the ATL according to how you have set up the ACS environment, but you are left with many volumes containing valid data sets on your old media, 3480 cartridges, for example. You can use the DFSMShsm RECYCLE function to move your backup or migration data to the ATL.

Update your ARCCMDxx PARMLIB member with the SETSYS RECYCLEOUTPUT command to specify to which media type your backup and migration data sets are recycled.

```
SETSYS RECYCLEOUTPUT(BACKUP(unittype),MIGRATION(unittype))
```

You can selectively move date into the ATL, depending on time constraints, rack limitations and savings, or any number of factors. For example, if you want to slowly move your data into the ATL, you can start with your DFSMShsm backup data only and allow natural attrition to take its course.

```
SETSYS RECYCLEOUTPUT(BACKUP(3590-1))
```

Alternatively, for fast migration to the ATL, you can specify both BACKUP and MIGRATION and then command RECYCLE your old volumes.

```
SETSYS RECYCLEOUTPUT(BACKUP(3590-1),MIGRATION(3590-1))

then

RECYCLE EXECUTE VOLUME(volser)
RECYCLE EXECUTE VOLUME(volser) FORCE
```

You can also use the RECYCLE command to selectively include or exclude tape ranges. For example, if you choose to recycle all tapes, regardless of whether

the are backup or migration tapes, into the ATL and are not concerned with how full they are, you can use the following command:

```
HSEND RECYCLE EXECUTE ALL SELECT(INCLUDE(RANGE(volserF:volserQ)))  -
      PERCENTVALID(100)
```

Or you can recycle only backup tapes less than 30% utilized.

```
HSEND RECYCLE EXECUTE BACKUP SELECT(INCLUDE(RANGE(volserF: volserQ))) -
      PERCENTVALID(30)
```

where volserF and volserQ are the first and last volsers, respectively, in the range that you chose to recycle.

6.6 Emulating 3490 Cartridges on a 3590 Device

You can use the IBM 3590 Magstar tape drives as a 3490E emulated tape subsystem. Use the IBM 3591-A01 ESCON tape control unit, which can be connected to up to four IBM 3590 tape drives.

1. **Why would I emulate using a 3490 cartridge?**

   You would have to use the 3490E emulation of 3590 for one of these reasons:

   - You do not have the minimum software installed to support the 3590 Magstar tapes as real 3590 devices, but you have the minimum support for 3490E.
   - You have installed a StorageTek (STK) ATL and want to install the IBM 3590 Magstar in it. In this case, an IBM 3590-C12 is attached to the STK ATL. This frame contains four IBM 3590-B1A drives. It is connected to an IBM 3591-A01 controller that emulates a 3490E subsystem to the host. This kind of emulation is no longer necessary for STK tape library customers who want to install IBM 3590-1 in the STK ATL, as IBM now supports native 3590-1 in the STK ATLs.

   Using the 3590 as emulated 3490E you can:

   - **Solve capacity and performance problems**. The capacity of an IBM Magstar cartridge is approximately 12 times that of an IBM 3490E tape drive using ECCST. Also, the performance of the IBM Magstar tape drives may substantially improve throughput for certain tape applications.
   - **Maximize investment protection**. Later, you can move the IBM Magstar cartridges and install the tape drives in an IBM ATL, such as the IBM 3494.
   - **Invest in extendable technology**. You are investing in a technology that is going to be extended in the future.
   - **Prepare for the future**. You are preparing to take full advantage of all functions and features of the IBM Magstar technology.

2. **What updates do I make to ARCCMDxx?**

The setup presented below is required when you install IBM 3591-attached drives and start using them with DFSMShsm.

a. **Direct DFSMShsm output to 3591-attached devices.**

   Specify which DFSMShsm functions should use 3591 devices. In this example, all DFSMShsm functions are directed to 3591-attached devices. EMUL3490 is an example of an esoteric unit name for the new 3591 drives, and ESO3480 refers to existing IBM 3480 tape drives. The esoteric that you use must be defined to MVS and then identified to DFSMShsm with SETSYS USERUNITTABLE before it can be used on other DFSMShsm commands.

```
SETSYS USERUNITTABLE(ESO3480:ESO3480, EMUL3490:EMUL3490)

SETSYS BACKUP(TAPE(EMUL3490)) -
   TAPEMIGRATION(ML2TAPE(TAPE(EMUL3490))) -
   RECYCLEOUTPUT(BACKUP(EMUL3490) MIGRATION(EMUL3490))

DEFINE DUMPCLASS(classname UNIT(EMUL3490))
```

   Note that you have to specify your existing esoteric ESO3480 on the SETSYS USERUNITTABLE command, even if it is no longer used for output, to allow your 3480 volumes associated with that esoteric name to be used for input.

b. **Avoid attempts to extend the native 3490E cartridges on 3591 devices.**

   Existing cartridges previously used for the same DFSMShsm functions for which 3591 cartridges will be used must be excluded from selection by DFSMShsm as they could be directed to a 3591-attached drive for output.

   DFSMShsm can select any of the following for output on a 3490E drive:

   - An empty or a partially filled ECCST cartridge
   - An empty CST cartridge
   - A partially filled CST cartridge in 36-track format

   You have to mark full all partially filled 36-track backup and migration volumes and delete all empty CST and ECCST cartridges to prevent their selection by DFSMShsm. Note that dump volumes are always automatically marked full by DFSMShsm.

   DFSMShsm can select a partially filled cartridge for output on a 3591-attached drive even if the cartridge was previously written to another device type. The esoteric name on which a tape was first written does not have to be the same when the tape is later used. The unit name associated with a volume does not prevent the volume from being selected for output if it is different from the unit name specified for the DFSMShsm function. DFSMShsm extracts the common technology such as from the esoteric name set of drives that is used for the output function, and, if a writeable tape fits that technology, it can be selected for output.

   Use the LIST command to create a list of partial or available tapes, and, for each volser on the list, submit a DELVOL command to mark it full or delete it:

```
LIST TTOC SELECT(BACKUP NOTFULL) ODS(dsn)
LIST TTOC SELECT(ML2 NOTFULL) ODS(dsn)
LIST DUMPVOLUME SELECT(AVAILABLE) ODS(dsn)

DELVOL volser BACKUP(MARKFULL)
DELVOL volser MIGRATION(MARKFULL)

DELVOL volser BACKUP(PURGE)
DELVOL volser MIGRATION(PURGE)
DELVOL volser DUMP(PURGE)
```

c. **Retain read access to existing DFSMShsm cartridges.**

For each volume that DFSMShsm uses it records a unit name that will be used for subsequent allocations of the tape for input. Even if an esoteric name is used for selecting the output device was specified when the volume was defined using the ADDVOL command, a generic unit name may be recorded for the volume once it has been used. If this generic unit name is 3480, 3480x, or 3490, DFSMShsm may ask them to be mounted in a 3591-attached device.

The LIST command allows you to identify those volumes that show device type 3480, 3480X, or 3490. If you have redefined an existing esoteric to point to the new 3591-attached drives, you have to change those as well. Then enter an ADDVOL command to make each change. The esoteric that you use must be defined to MVS and then identified to DFSMShsm before it can be used in an ADDVOL command. In this example, ESO3480 is used as an esoteric for existing 3480 drives, and EMUL3490 is used for the 3591-attached drives:

```
LIST ML2(TAPE) ODS(dsn)
LIST BVOL ODS(dsn)
LIST DVOL ODS(dsn)

SETSYS USERUNITTABLE(ESO3480:ESO3480, EMUL3490:EMUL3490)

ADDVOL volser UNIT(ESO3480) MIG(ML2)
ADDVOL volser UNIT(ESO3480) BACKUP(DAILY)
ADDVOL volser UNIT(ESO3480) BACKUP(SPILL)
ADDVOL volser UNIT(ESO3480) DUMP
```

d. **Set the PERCENTFULL parameter.**

The capacity of a 3590 cartridge is approximately 12 times that of an ECCST cartridge. We recommend setting the PERCENTFULL value to 1100% to ensure that a tape can be duplexed or copied to another tape, even if the length of the alternate tape is shorter than the original.

```
SETSYS TAPEUTILIZATION(UNIT(EMUL3490) PERCENTFULL(1100))
```

e. **Recycle old tapes.**

Once your DFSMShsm environment is up and running using 3591-attached drives, you may want to run RECYCLE to move migration and backup tapes to 3590 cartridges. Statement (a) specifies that all tapes within the CST000 — CST099 range will be recycled, and their valid data will be moved to the new 3590 cartridges, as we have specified in the SETSYS RECYCLEOUTPUT command in the ARCMDxx member of PARMLIB. In this example, we are assuming that the old tapes we want to recycle are in the range specified. Statement (b) is an example of recycling, all tapes that have 30% or less of valid data in them.

```
(a) RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(CST000:CST099)))
            PERCENTVALID(100)

(b) RECYCLE ALL EXECUTE PERCENTVALID(30)
```

f. **Setup ABARS processing.**

The aggregate backup function of ABARS creates the following files on tape volumes, which can then be transported to the recovery site:

- Data file
- Control file
- Instruction/activity log file (optional)

The output files are physical sequential data sets. Each of these files must be written to a separate physical tape volume (or volumes, as the files may extend several volumes). The size of the data file can be very large, making it a good candidate for 3590 cartridges. The control and instruction files tend to be much smaller, so you may want to write them on smaller capacity media. However, ABARS does not allow you to specify the output unit individually for these files. The unit name specified in the ABACKUP command applies to all three files.

To allocate the aggregate backup and ABARS files to 3591-attached drives, issue this command:

```
ABACKUP aggregatename EXECUTE UNIT(EMUL3490)
```

EMUL3490 is the esoteric unit name for our IBM 3591 tape subsystem.

If you are using an IBM 3494 or 3495 Automated Tape Library Dataserver with system-managed tape, you can use ACS routines to override the ABACKUP unit specification and direct the output files to different media.

Figure 95 on page 223 shows an ACS storage class code fragment selecting ABARS output data set names, as per the FILTLIST, and directing the control and instruction files to system-managed tape volumes in the IBM Automated Tape Library Dataserver by assigning them storage class SC54TAPE. The data file is assigned a null storage class, that is, it is not to be system-managed and will be allocated to the nonlibrary tape unit specified on the ABACKUP command.

```
PROC STORCLAS

/*********************************************************************/
FILTLIST AB_DATA  INCLUDE(ABARS.PAY1.D.C%%V%%%%,   /* LO Data file     */
                          ABARS.PAY1.O.C%%V%%%%)   /* Other Data file  */
FILTLIST AB_CNTL  INCLUDE(ABARS.PAY1.C.C%%V%%%%,   /* Control file     */
                          ABARS.PAY1.I.C%%V%%%%)   /* Instruction file */
FILTLIST AB_UNIT  INCLUDE('EMUL3490')
/*********************************************************************/
  .
  .
  .
SELECT
  WHEN (&DSN = &AB_CNTL && &UNIT = &AB_UNIT)       /* Direct control   */
    DO                                             /* and instruction  */
      SET &STORCLAS = 'SC54TAPE'                   /* files to tape    */
      EXIT                                         /* storage class    */
    END

  WHEN (&DSN = &AB_DATA && &UNIT = &AB_UNIT)       /* Direct data file */
    DO                                             /* to drives        */
      SET &STORCLAS = ''                           /* outside ATL      */
      EXIT
    END
END
  .
  .
  .
END /* OF STORAGE CLASS ROUTINE PROC */
```

Figure 95. Sample ACS for Directing ABARS Data Sets to an ATL

ABARS.PAY1 is the prefix for the ABARS output data sets that we specified in our aggregate group definition.

3. What updates do I make to DFSMSrmm?

There are no additional definitions to include in DFSMSrmm. If you do not have already defined volume pools to DFSMSrmm, you have to define a new volume pool by adding the following definition in EGDRMMxx member of PARMLIB:

```
VLPOOL PREFIX(M*) TYPE(SCRATCH) MEDIANAME(3590) -
       DESCRIPTION('3590 SCRATCH POOL') RACF(N) EXPDTCHECK(N)
```

Then, use DFSMSrmm subcommands ADDRACK and ADDVOLUME to define the new volumes in your system:

```
RMM ADDRACK M00000 COUNT(100)

RMM ADDVOLUME M00000 COUNT(100) POOL(M*)
    STATUS(SCRATCH) INIT(Y) MEDIANAME(3590)
```

6.7 Reducing Recall Processing Time

In a multihost environment DFSMShsm can experience delays during recall processing when the SETSYS TAPERECALLLIMITS command (which is provided in APAR OW28621) is not soecified. This optional parameter specifies how long a single tape can remain mounted before DFSMShsm checks to see whether there are higher priority recall requests waiting to process from the same system or another host.

1. **What is the syntax of the TAPERECALLIMITS parameter to add to ARCCMDxx?**

```
SETSYS TAPERCALLLIMITS( TASK(time) TAPE(time) )
```

2. **What happens if I do not specify the TAPERECALLLIMITS parameter?**

   DFSMShsm uses the default values for this parameter:

```
SETSYS TAPERCALLLIMITS( TASK(15) TAPE(20) )
```

3. **What do the optional subparameters of TAPERECALLLIMITS specify?**

   TASK specifies the number of minutes that a recall task has on a single tape mount before DFSMShsm checks for any higher priority recalls on that host, provided the maximum number of tape recall tasks are active. If a higher priority is task is found, DFSMShsm will complete its current recall task, demount the tape, and process the new higher priority task.

   TAPE specifies the number of minutes that a recall task has on a single tape mount before additional multihost criteria are considered. If a higher priority task is found on another DFSMShsm host, DFSMShsm will complete the current recall task and demount the tape for use on the other host.

   Once this action has been performed, DFSMShsm excludes any recall request from that tape to allow the host system(s) time to retry their delayed requests and catch up.

   So in the default setting of TASK(15), DFSMShsm will process recalls from a single tape mount for 15 minutes before checking for higher priority tasks on that system, which it will honor if it finds any.

   In the default setting of TAPE(20), DFSMShsm will process recalls from a single tape mount for 20 minutes before checking for higher priority tasks on another host, which it will honor if it finds any.

4. **How would I know whether a tape was demounted for use on another system?**

   You will see the following message:

```
ARC0312I RECALLS FROM TAPE volser TERMINATED FOR USAGE BY HOST h
```

   where *h* is the identifier of the host with the highest priority recall request.

5. **What if I have many tape drives but I never have tape recall requests queued up behind each other?**

   In this instance the TAPERECALLLIMITS would most likely not interrupt recall processing for a higher priority recall request; it would have already been satisfied on another tape drive.

6.8 Tape Mount Management

Depending on your installation's current tape practices, you can maximize the use of your tape resources and even improve batch job throughput by implementing tape mount management (TMM).

TMM is a method of managing tape data sets within the SMS storage hierarchy. Here is how TMM works:

1. Tape data sets are categorized according to size, pattern of usage, and other criteria, so that appropriate SMS policies can be assigned to TMM candidates.
2. Data sets written to tape are intercepted at allocation and, if eligible for tape mount management, redirected to a system-managed DASD pool. The pool serves as a staging area for these data sets until they are migrated to tape by DFSMShsm. The location of the data is transparent to the application program.
3. During interval migration, DFSMShsm checks the occupancy of the DASD buffer storage group to ensure that space is available when needed, migrating data sets to a lower level of the storage hierarchy when they are no longer required on primary DASD volumes.
4. DFSMShsm eventually moves the data to tape, using single-file format and data compaction to create full tape cartridges.
5. If an application later requests a data set, DFSMShsm automatically recalls it from where it resides in the storage hierarchy and allocates it on primary DASD for access.

This process can significantly reduce tape mounts and the number of cartridges required to store the data.

6.8.1 TMM Implementation Plan

Implementing TMM requires careful planning. Following are the major tasks you should include in your implementation plan:

1. Analyze your current tape environment
2. Simulate the proposed TMM environment
3. Select DASD volumes to satisfy buffer requirements
4. Define SMS constructs for the candidate data sets
5. Create the ACS routines, or update the current ones
6. Tune DFSMShsm operations

6.8.2 Volume Mount Analyzer

Volume mount analyzer (VMA) is a tool that can help you analyze your installation's current tape environment. It uses system management facility (SMF) data to analyze tape mount activity of a certain period and produces reports that help you:

- Get information about tape mount events, such as data set name, job name, program, and data set size
- Evaluate the tape hardware configuration
- Determine which data sets are good candidates for TMM
- Determine data class and management class requirements for TMM candidates
- Determine patterns that could be used as ACS filters for inclusion in TMM or exclusion from TMM
- Determine the resources that are needed, and the benefits of the TMM implementation at your installation

VMA can create reports that tell you what are the best management class options for the TMM candidates, and the amount of primary DASD and ML1 DASD. These reports also show you the additional tape mount activity that should occur with DFSMShsm as a result of TMM implementation.

For detailed procedures on performing a volume mount analysis and interpreting the results, see *DFSMS/MVS Using the Volume Mount Analyzer*.

6.8.3 DFSMShsm Parameters and TMM

Figure 96 is an example of the SETSYS commands that are applicable to a TMM implementation.

```
/*********************************************************************/
/*  SETSYS COMMANDS IN THE ARCCMDXX PARMLIB MEMBER THAT DEFINE THE     */
/*  DFSMSHSM ENVIRONMENT FOR TMM IMPLEMENTATION                        */
/*********************************************************************/
/*
SETSYS COMPACT(DASDMIGRATE)
SETSYS SELECTVOLUME(SCRATCH)
SETSYS PARTIALTAPE(REUSE)
SETSYS TAPEDELETION(SCRATCHTAPE)
SETSYS TAPEMIGRATION(ML2TAPE)
SETSYS TAPESECURITY(RACFINCLUDE)
ADDVOL HSM14D UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -
        NOSMALLDATASETPACKING)                     -
        THRESHOLD(90)
/*
```

Figure 96. SETSYS Parameters for TMM

SETSYS COMPACT(DASDMIGRATE) causes DFSMShsm to compress the data sets when migrating to DASD. This option should be used only if some of the TMM data goes to ML1. It should not be used if RVA DASD is used for ML1.

SETSYS SELECTVOLUME(SCRATCH) specifies that DFSMShsm select scratch tapes from a global scratch pool.

SETSYS PARTIALTAPE(REUSE) specifies that DFSMShsm try to reuse tapes that are partially filled with DFSMShsm data. This makes better use of the tape media.

SETSYS TAPEDELETION(SCRATCHTAPE) specifies that recycled migration and backup tapes, along with expired dump tapes, are going to be returned to a global scratch pool.

SETSYS TAPEMIGRATION(ML2TAPE) specifies that ML2 media is tape.

SETSYS TAPESECURITY(RACFINCLUDE) specifies that DFSMShsm protect its tape volumes by adding them to a RACF tape volume set of DFSMShsm.

ADDVOL can be used to add the ML1 volumes that are needed. VMA tells you how much additional ML1 DASD space is needed for TMM implementation.

Chapter 7. Hints and Tips for Tuning DFSMShsm

DFSMShsm has many commands and parameters that allow a great deal of flexibility and options. However, many installations have unique requirements that are not supported by these commands and parameters available. To support these unique requirements help may be at hand with DFSMShsm tuning patches.

The patches that we cover in this chapter are supported DFSMShsm patches that will remain supported from release to release without any further modification on your part.

Before applying any patches to your system you must take the following into account:

- You must be familiar with the DFSMShsm PATCH command.
- To check for any errors when using the PATCH command, you must specify the VERIFY parameter.
- To see the output from a PATCH command, specify the ODS parameter.

1. **What does the PATCH command do?**

   The PATCH command changes storage within the DFSMShsm address space. You can identify the storage location to be changed with an absolute address or a qualified address.

2. **Where can I find comprehensive information about tuning patches?**

   Refer to the *DFSMShsm Implementation and Customization Guide*.

3. **Where can I find the full documentation for the DFSMShsm PATCH command?**

   This command is fully documented in the *DFSMShsm Diagnosis Guide*.

4. **Can I put the PATCH command into my ARCCMDxx PARMLIB member to save having to enter the command each time?**

   Yes. As long as you have the same requirement day after day, this is not a problem. We advise you to add comments that indicate the intention of the patch so that others are aware of its purpose.

5. **How do I allow password-protected generation data sets to be migrated and then scratched without checking the password at roll-off time?**

   Issue the following command:

```
PATCH .MCVT.+53 BITS(.1......)
```

6. **How do I scratch generation data sets at roll-off time, regardless of their expiration dates?**

   First, the SCRATCH option has to have been issued when the GDG was defined.

   To have DFSMShsm ignore the expiration date of a migrated GDS at roll-off time, issue the following command:

```
PATCH .MCVT.+53 BITS(1.......)
```

You can use this command in conjunction with the previous patch command under step 5.

7. **How can I determine why my SMS-managed data sets were not processing during automatic migration or backup?**

   This information is recorded in DFSMShsm message ARC0734I. If you want to display this message for every SMS-managed data set during migration, issue the following command:

```
PATCH .MGCB.+26 X'FF'
```

   For backup, issue the following command:

```
PATCH .BGCB.+24 X'FF'
```

8. **In JES3 how do I disable the delay in issuing PARTREL for generation data sets?**

   Issue the following command:

```
PATCH .MGCB.+27 BITS(.......1)
```

9. **How do I use DFSMShsm in a JES3 environment that performs main device scheduling only for tapes?**

   Issue the following command:

```
PATCH .MCVT.+29A BITS(1.......)
```

10. **How can I prevent or alter the prevent-migration activity for JES3 setups?**

    To prevent the JES3 migration activity, issue the following command:

```
PATCH .MCVT.+14A X'00' VERIFY(.MCVT.+14A X'03')
```

    To shorten migration issue the following command where *nn* is the number of days by which you will shorten it: the number of days you wish to shorten it:

```
PATCH .MCVT.+14A X'nn' VERIFY(.MCVT.+14A X'03')
```

11. **How do I replace HSMACT as the high-level qualifier for activity logs?**

    Issue the following command:

```
PATCH .MCVT.+321 'SHORT ' VERIFY(.MCVT.+321 'HSMACT')
```

12. **How can I change the parameters passed to DFSMSdss?**

    - To specify that volume dumps be performed with the COMPRESS function, issue the following command:

```
PATCH .MCVT.+3C3 BITS(..1.....)
```

    - To specify that volume dumps be performed with the COMPRESS function for dump class IPLVOLS only, issue the following command:

```
FIXCDS W IPLVOLS PATCH(X'01' BITS(1.......))
```

    - To remove the ALLEXCP function from the volume-dump function, issue the following PATCH command:

```
PATCH .MCVT.+3C3 BITS(1.......)
```

    - To remove the ALLDATA function from the volume-dump function, issue the following command:

```
PATCH .MCVT.+3C3 BITS(.1......)
```

    - To change the default block size of 64 KB used for volume dumps to 32 KB, issue the following command:

```
PATCH .MCVT.+3D4 BITS(.......1)
```

13. **Is it possible to enable ABARS ABACKUP and ARECOVER to wait for a tape unit allocation?**

    For ABACKUP, issue the following command:

```
PATCH .ABRCB.+81 BITS(...1....)
```

    For ARECOVER, issue the following command:

```
PATCH .ABRCB.+81 BITS(....1...)
```

14. **How can I change the RACF Facility class user ID for the console operator's terminal?**

To change the user ID to something other than the default of OPER issue the following command, where userid is the user ID you want to change it to:

```
PATCH .ABRCB.+4BC 'userid'
```

15. **How can I allow DFSMShsm automatic functions to process volumes more than once per day?**

    DFSMShsm will not allow automatic functions to run on a volume if any processor has completed processing that volume within the past 14 hours.

16. **How can I restart automatic space management?**

    If automatic primary space management has run to completion and you want to start automatic primary space management again, use one or both of the following commands:

    For SMS-managed volumes:

```
PATCH .MCVT.+414 X'00000000'
```

    For non-SMS-managed volumes:

```
PATCH .MCVT.+488 X'00000000'
```

    In both cases you have to issue a SETSYS PRIMARYSPMGMSTART command to define a new start window.

17. **How can I restart automatic backup?**

    Automatic backup can allowed to run multiple times per day. You have to consider the phases of backup that you want to run:

    a. Phase 1 - Backing up the CDSs
    b. Phase 2 - Moving backup versions
    c. Phase 3 - Backing up migrated data sets
    d. Phase 4 - Backing up DFSMShsm-managed volumes with the automatic backup attribute

    If automatic backup has run to completion and you want to start automatic backup functions pertaining to any phase, you must issue the following command:

```
PATCH .BCR.+50 X'00000000'
```

    If automatic backup has run to completion and you want to start automatic backup functions pertaining to phases 2 and 3, you must issue the following command:

```
PATCH .BCR.+5C X'00000000'
```

If automatic backup has run to completion and you want to start automatic backup functions pertaining to phase 4, you must use the patch for phase 1. The phase 1 patch enables automatic backup if the volumes have not been processed in the last 14 hours. If the volumes have been processed in the last 14 hours and you want to process them now, you must additionally issue one or both of the following commands:

For an SMS-managed volume:

```
PATCH .MCVT.+418 X'00000000'
```

For a non-SMS-managed volume:

```
PATCH .MCVT.+48C X'00000000'
```

Issue the SETSYS AUTOBACKUPSTART command to define a new window.

18. **How can I restart automatic dump?**

    To allow automatic dump to be run several times, you need to issue one or both of the following commands:

    For an SMS-managed volume:

```
PATCH .MCVT.+41C X'00000000'
```

    For a non-SMS-managed volume:

```
PATCH .MCVT.+490 X'00000000'
```

    Issue the SETSYS AUTODUMPSTART command to define a new window.

19. **Is it possible to change the default tape data set names that DFSMShsm uses for tape copy and full volume dump?**

    In releases of DFHSM before Version 2 Release 6.0, the default names for tape copy and full volume dump tape data sets were different from the default names used in DFHSM Version 2 Release 6.0 and in subsequent DFSMShsm releases. Some sites have used and continue to use naming conventions established before DFHSM Version 2 Release 6.0. As a convenience, this patch is provided to eliminate changing naming conventions that have been established with previous releases of DFHSM. If you want DFSMShsm to use a naming convention that is consistent with releases of DFHSM, issue the following commands:

    For default tape copy names before DFHSM 2.6:

```
PATCH .MCVT.+284 BITS(.0......)
```

    For default dump tape names before to DFHSM 2.6:

```
PATCH .MCVT.+284 BITS(0.......)
```

20. **How do I prevent interactive TSO users from being placed in a wait state during a data set recall?**

    Issue one or both of the following commands:

    For tape data set recalls:

```
PATCH .MCVT.+52 bits(..1.....)
```

    For DASD data set recalls:

```
PATCH .MCVT.+52 bits(...1....)
```

21. **How can I prevent ABARS ABACKUP processing from creating an extra tape volume for the instruction data set and activity log files?**

    To dump the activity log and instruction data set to a separate tape volume, issue the following command:

```
PATCH .ABRCB.+1F X'00'
```

    To never dump the activity log and instruction data set to a separate tape volume, issue the following command:

```
PATCH .ABRCB.+1F X'03'
```

22. **How do I stop ABARS ABACKUP processing from including multivolume BDAM data sets?**

    To stop this issue the following command:

```
PATCH .ABRCB.+A8 BITS(.....1..)
```

23. **How can I prevent ABARS ABACKUP processing from deleting existing ABACKUP output files?**

    To prevent this issue the following command:

```
PATCH .ABRCB.+82 BITS(......1.)
```

24. **How do I specify the amount of time to wait for an ABARS secondary address space to initialize?**

    To specify this issue the following command:

```
PATCH .ABRCB.+494 X'nnnnnnnn' VERIFY(.ABRCB.+494 X'0000012C')
```

where X'nnnnnnnn' is the hexadecimal representation for the number of seconds to wait for an ABARS secondary address space to start.

25. **How do I patch ABARS to use NOVALIDATE when invoking DFSMSdss?**

    Issue the following command:

```
PATCH .ABRCB.+82 BITS(1.......)
```

26. **How can I patch ABARS to provide a dump whenever a specific error occurs during DFSMSdss processing during ABACKUP and ARECOVER?**

    To obtain a dump on DFSMSdss message ADR454I, issue the following command:

```
PATCH .ABRCB.+1C X'F4F5F4'
```

27. **Can I route the ABARS ARC6030I message to the operator console?**

    Yes. Issue the following command:

```
PATCH .ABRCB.+81 BITS(.....1..)
```

28. **How do I patch ABARS to prevent it from automatically cleaning up residual versions of ABACKUP output files?**

    Issue the following command:

```
PATCH .ABRCB.+82 BITS(.1......)
```

29. **How can I change the default number of recall requests for a data set residing on a volume that is in use by recycle or TAPECOPY processing?**

    To fail recall if the tape is in use by recycle or tapecopy, issue the following command:

```
PATCH .MCVT.+315 X'00'
```

    The default number of times that a recall is tried is 15. If you want to change the number, for example, to 30 times, issue the following command:

```
PATCH .MCVT.+315 X'1E'
```

30. **Is it possible to change the default number of buffers DFSMShsm uses to back up data sets?**

To change this number issue the following command:

```
PATCH .MCVT.+391 X'nn'
```

where nn can be 01 through to 05.

31. **How do I disable delete-if-backed-up (DBU) processing for SMS data sets?**

    Issue the following command:

```
PATCH .MCVT.+431 BITS(......1.)
```

32. **How do I request that the message issued for SETSYS TAPEOUTPUTPROMPT processing be a WTOR instead of the default WTO?**

    To request a WTOR, issue the following command:

```
PATCH .MCVT.+4C3 BITS(.......1)
```

    To request a WTO, issue the following command:

```
PATCH .MCVT.+4C3 BITS(.......0)
```

33. **How do I remove ACL as a condition for D/T3480 esoteric unit name translation?**

    Issue the following command:

```
PATCH .MCVT.+4C0 BITS(...1....)
```

    If you use this patch in your ARCCMDxx PARMLIB member, it must be before your SETSYS USERUNITTABLE command.

34. **Is it possible to change the amount of time that ABACKUP processing waits for an ML2 volume to become available?**

    Yes. Issue the following command:

```
PATCH .ABRCB+4A2 X'nnnn'
```

    where X'nnnn' is the hexadecimal representation for the number of 20-second intervals to be tried. The default decimal value is 90, which produces a 30-minute wait.

35. **How do I prevent deadlocks during volume dumps?**

    Issue the following command:

```
PATCH .MCVT.+3C3 BITS(.....1..)
```

36. **How can I modify the number of elapsed days for a checkpointed data set to be eligible for migration?**

    Issue the following command:

```
PATCH .MGCB.+70 X'nn'
```

    where X'nn' is the hexadecimal representation of the number of days. The default is 5.

Appendix A. Sample ACS Routines

This appendix contains sample ACS routines used to manage DFSMShsm data.

A.1 Sample Storage Class ACS Routine

In our system we have choosen a high-level qualifier of HSM for the DFSMShsm data sets, which is reflected in the ACS routines. Figure 97 on page 240 shows the following:

- A storage class for high performance and with guaranteed space (SC54GRT) is assigned to the DFSMShsm control data sets and journal. This storage class is explicitly specified in the allocation of these data sets. Only the storage administrators (MHLRES3, MHLRES4, and MHLRES5) can use the SC54GRT storage class.
- Data allocated on DFSMShsm-owned volumes (migration and backup) is excluded from DFSMSdfp management.
- Other DFSMShsm data sets, such as logs and CDS backups, are recognized and allocated in the DFSMSdfp volume pools.
- When a data set is recalled or recovered by DFSMShsm, the storage class is retained if one already exists.

Examples of the parameters used in the DFSMSdfp classes referenced in these routines are presented in 2.1.2, “Storage Class” on page 9.

```
PROC &STORCLAS

 /*******************************************************************/
 /* START OF STORAGE CLASS PROC                                     */
 /*******************************************************************/
 /*  CHANGE HISTORY                                                 */
 /*  DATE     PERSON  DESCRIPTION OF CHANGE                         */
 /*******************************************************************/
 /* 95/05/06   DEF    INITIAL IMPLEMENTATION FOR STAGE I            */
 /*                   DATA                                          */
 /* 95/02/05   DEF    MINIMAL CONFIGURATION                         */
 /*******************************************************************/
 /* YOU NEED TO MOVE ALL THE FILTLISTS TO THE                       */
 /* LOCATION BETWEEN THE TWO LINES OF SPECIAL CHARACTERS SHOWN      */
 /* BELOW THIS COMMENT BEFORE USING THE CODE FRAGMENTS              */
 /*******************************************************************/


/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


        FILTLIST &OMVS_DATA_DSN INCLUDE
        (OMVS*.**)

/*******************************************************************/

        FILTLIST &HSM_NONSMS_DATA_DSN INCLUDE
        (HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,HSM.MDB.**,
         HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)

/*******************************************************************/

        FILTLIST &HSM_CONTRDS_DATA_DSN INCLUDE
        (HSM.%CDS.*,HSM.JRNL.*)

/*******************************************************************/

        FILTLIST &HSM_LOGS_DATA_DSN INCLUDE
        ('HSM.EDITLOG',HSM.HSMLOG*,HSM.HSMPDO*)

/*******************************************************************/

        FILTLIST &HSM_ACT_DATA_DSN INCLUDE
        (HSMACT.H%.*.D*.T*)

/*******************************************************************/

        FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
        (HSM.%CDS.BACKUP.*,
         HSM.JRNL.BACKUP.*)

/*******************************************************************/

        FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
        (ABARS.*.INSTRUCT)
```

Figure 97 (Part 1 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/

        FILTLIST &SMS_MANAGED_OMVS_DATA_HLQ INCLUDE
        (OMVS*)

        FILTLIST &SPECIAL_SC_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_SCNOSMS_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_STORCLAS INCLUDE
        ('SC54GRT','SC54STD','SC54LOW')

        FILTLIST &SPECIAL_MC_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_MGMTCLAS INCLUDE
        ('MC54SPEC')

/*********************************************************************/


        FILTLIST &VALID_DASD_UNIT INCLUDE
        ('3380','3390','SYSDA','SYSALLDA','')

        FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
        ('STORE','CHANGE','CTRANS')

        FILTLIST &VALID_TAPE_UNIT INCLUDE
        ('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')

/*********************************************************************/


/** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **/

    SELECT /* MAIN LOGIC BLOCKS */

 /*********************************************************************/
 /* INCUDE ALL STANDARDS ENFORCEMENT LOGIC IN THIS BLOCK, BUT ONLY    */
 /* IN EITHER THE DATA CLASS OR STORAGE CLASS ACS ROUTINE             */
 /*********************************************************************/



 /*********************************************************************/
 /* TAILOR AND ADD DATA SET NAMING STANDARD HERE IF USED              */
 /*********************************************************************/



 /*********************************************************************/
 /* TAILOR AND ADD VOLUME FENCING STANDARD HERE IF USED               */
 /*********************************************************************/
```

Figure 97 (Part 2 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD DATA SET SIZE STANDARD HERE IF USED                */
/*********************************************************************/



/*********************************************************************/
/* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK, INCLUDE BOTH DASD     */
/* ALLOCATIONS FOR SMS AND NON-SMS                                   */
/*********************************************************************/

      WHEN ((&UNIT EQ &VALID_DASD_UNIT)
      AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))

          SELECT /* DASD DATA ALLOCATION */

/*********************************************************************/
/* EXCLUDE DFSMSHSM RECALLS AND RECOVERIES FROM ACS PROCESSING IF    */
/* A STORAGE CLASS IS ALREADY ASSIGNED TO THE DATA SET               */
/*********************************************************************/

              WHEN ((&ACSENVIR = 'RECALL' OR &ACSENVIR ='RECOVER')
              AND (&STORCLAS NE ''))

                  DO
                      SET &STORCLAS = &STORCLAS
                      EXIT
                  END /* DO */


/*********************************************************************/
/* ALLOW SPECIAL USERS TO PLACE SELECTED DATA SETS OUTSIDE OF SMS    */
/* BY USING STORCLAS=SCNOSMS AT ALLOCATION                           */
/*********************************************************************/

              WHEN ((&USER EQ &SPECIAL_SCNOSMS_USER)
              AND (&STORCLAS EQ 'SCNOSMS'))

                  DO
                      SET &STORCLAS = ''
                      EXIT
                  END /* DO */


/*********************************************************************/
/* ALLOW HSM NON-SMS SELECTED DATA SETS OUTSIDE OF SMS               */
/*********************************************************************/

              WHEN (&DSN EQ &HSM_NONSMS_DATA_DSN)
                  DO
                      SET &STORCLAS = ''
                      EXIT
                  END /* DO */


/*********************************************************************/
/* TAILOR AND ADD JCL EXCEPTIONS HERE IF THEY ARE USED               */
/*********************************************************************/
```

Figure 97 (Part 3 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND SET STORCLAS FOR ALLOCATION EXCEPTIONS                 */
/*********************************************************************/
/* REPEAT THIS BLOCK FOR EACH SET OF SPECIAL USERS THAT ARE          */
/* ALLOWED TO REQUEST SMS SERVICE THROUGH STORCLAS AT ALLOCATION     */
/*********************************************************************/

              WHEN ((&USER EQ &SPECIAL_SC_USER)
              AND (&STORCLAS EQ &SPECIAL_STORCLAS))

                  DO
                      SET &STORCLAS = &STORCLAS
                      EXIT
                  END /* DO */



/*********************************************************************/
/* TAILOR AND ADD DATA SET LIST EXCEPTIONS HERE IF THEY ARE USED     */
/* INCLUDE SEQUENTIAL STRIPING AND CONCURRENT COPY                   */
/*********************************************************************/




/*********************************************************************/
/* TAILOR AND ADD EITHER TEMP AND VIO OR TEMP AND NO VIO HERE IF     */
/* TEMP DATA IS SMS-MANAGED                                          */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD TEMP SHORT DURATION HERE IF TEMP SHORT DURATION    */
/* DATA AND IF IT IS SMS-MANAGED                                     */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD TSO LOGON AND LIST DATA SUBTYPES IF THEY HAVE      */
/* STORAGE CLASSES DIFFERENT THAN OTHER TSO DATA AND THEY ARE        */
/* SMS-MANAGED                                                       */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD OTHER TSO SUBTYPES HERE IF THEY ARE SMS-MANAGED    */
/*********************************************************************/
```

Figure 97 (Part 4 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD TEST SUBTYPES HERE IF THEY ARE SMS-MANAGED         */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD BATCH PRODUCTION SUBTYPES HERE IF THEY ARE         */
/* SMS-MANAGED                                                       */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD ONLINE PRODUCTION SUBTYPES HERE IF THEY ARE        */
/* SMS-MANAGED                                                       */
/*********************************************************************/
```

Figure 97 (Part 5 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD ANY OTHER DATA SUBTYPES HERE IF THEY ARE           */
/* SMS-MANAGED                                                       */
/*********************************************************************/


              WHEN (&DSN EQ &OMVS_DATA_DSN)

                  DO
                      SET &STORCLAS ='OPENMVS'
                      EXIT
                  END /* DO */



              WHEN (&DSN EQ &HSM_LOGS_DATA_DSN)

                  DO
                      SET &STORCLAS ='SC54STD'
                      EXIT
                  END /* DO */

              WHEN (&DSN EQ &HSM_ACT_DATA_DSN)

                  DO
                      SET &STORCLAS ='SC54LOW'
                      EXIT
                  END /* DO */

              WHEN (&DSN EQ &HSM_BKCONT1_DATA_DSN)

                  DO
                      SET &STORCLAS ='SC54LOW'
                      EXIT
                  END /* DO */

              WHEN (&DSN EQ &HSM_ABARS_DATA_DSN)

                  DO
                      SET &STORCLAS ='SC54LOW'
                      EXIT
                  END /* DO */



/*********************************************************************/
/* MAKE ALL OTHER DASD DATA SETS NON-MANAGED                         */
/*********************************************************************/

              OTHERWISE /* NON-MANAGED DASD DATA */

                  DO
                      SET &STORCLAS = ''
                      EXIT
                  END /* DO */

          END /* DASD ALLOCATION SELECT */
```

Figure 97 (Part 6 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*******************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK                      */
/*******************************************************************/

       WHEN (&UNIT EQ &VALID_TAPE_UNIT)

           SELECT /* TAPE ALLOCATIONS */


/*******************************************************************/
/* TAILOR AND ADD TAPE MOUNT MANAGEMENT SUBTYPES HERE IF THEY EXIST  */
/*******************************************************************/




/*******************************************************************/
/* TAILOR AND ADD SMS-MANAGED TAPE SUBTYPES HERE IF THEY EXIST       */
/*******************************************************************/




/*******************************************************************/
/* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT  */
/* OR SMS-MANAGED TAPE EXIST                                       */
/*******************************************************************/


               WHEN (&UNIT EQ &VALID_TAPE_UNIT)
               SET &STORCLAS = &STORCLAS


           END /* TAPE ALLOCATION SELECT */


/*******************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK, INCLUDE LOGIC FOR  */
/* SUBTYPES FOR IAFC OR IMAGE PLUS HERE IF THEY EXIST                */
/*******************************************************************/

       WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)

           SELECT /* OPTICAL ALLOCATIONS */

               WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
               SET &STORCLAS = &STORCLAS
```

Figure 97 (Part 7 of 8). Storage Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD OPTICAL STORE FUNCTION HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD OPTICAL TRANSITION AND CHANGE FUNCTIONS HERE IF    */
/* OPTICAL DATA EXISTS                                               */
/*********************************************************************/



/*********************************************************************/
/* ADD OTHERWISE FOR OPTICAL DATA HERE IF OPTICAL DATA EXISTS        */
/*********************************************************************/



           END /* OPTICAL ALLOCATION SELECT */

/*********************************************************************/
/* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR       */
/* OPTICAL                                                           */
/*********************************************************************/

       OTHERWISE /* NON-MANAGED DEVICES */

           DO
               SET &STORCLAS = ''
               EXIT
           END /* DO */

   END /* SELECT */

END /* PROC */
```

Figure 97 (Part 8 of 8). Storage Class ACS Routine for DFSMShsm Data

A.2 Sample Management Class ACS Routine

Figure 98 on page 248 shows sample code required in the management class ACS routine, which assigns different management criteria to the DFSMShsm-owned data sets according to their use and importance.

```
PROC &MGMTCLAS

 /*********************************************************************/
 /* START OF MANAGEMENT CLASS PROC                                    */
 /*********************************************************************/
 /*  CHANGE HISTORY                                                   */
 /*  DATE     PERSON  DESCRIPTION OF CHANGE                           */
 /*********************************************************************/
 /* 95/05/06   DEF    INITIAL IMPLEMENTATION FOR STAGE I              */
 /*                   DATA                                            */
 /* 95/02/05   DEF    MINIMAL CONFIGURATION                           */
 /*********************************************************************/
 /* YOU NEED TO MOVE ALL THE FILTLISTS TO THE                         */
 /* LOCATION BETWEEN THE TWO LINES OF SPECIAL CHARACTERS SHOWN        */
 /* BELOW THIS COMMENT BEFORE USING THE CODE FRAGMENTS                */
 /*********************************************************************/


/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */


        FILTLIST &OMVS_DATA_DSN INCLUDE
        (OMVS*.**)

/*********************************************************************/

        FILTLIST &HSM_NONSMS_DATA_DSN INCLUDE
        (HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,
         HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)

/*********************************************************************/

        FILTLIST &HSM_CONTRDS_DATA_DSN INCLUDE
        (HSM.%CDS.*,HSM.JRNL.*)

/*********************************************************************/

        FILTLIST &HSM_LOGS_DATA_DSN INCLUDE
        ('HSM.EDITLOG',HSM.HSMLOG*,HSM.HSMPDO*)

/*********************************************************************/

        FILTLIST &HSM_ACT_DATA_DSN INCLUDE
        (HSMACT.H%.*.D*.T*)

/*********************************************************************/

        FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
        (HSM.%CDS.BACKUP.*,
         HSM.JRNL.BACKUP.*)

/*********************************************************************/

        FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
        (ABARS.*.INSTRUCT)
```

Figure 98 (Part 1 of 7). Management Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/

        FILTLIST &SMS_MANAGED_OMVS_DATA_HLQ INCLUDE
        (OMVS*)

        FILTLIST &SPECIAL_SC_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_SCNOSMS_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_STORCLAS INCLUDE
        ('SC54GRT','SC54STD','SC54LOW')

        FILTLIST &SPECIAL_MC_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_MGMTCLAS INCLUDE
        ('MC54SPEC')

/*********************************************************************/


        FILTLIST &VALID_DASD_UNIT INCLUDE
        ('3380','3390','SYSDA','SYSALLDA','')

        FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
        ('STORE','CHANGE','CTRANS')

        FILTLIST &VALID_TAPE_UNIT INCLUDE
        ('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')

/*********************************************************************/


/** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ **/

    SELECT /* MAIN LOGIC BLOCKS */

 /*********************************************************************/
 /* INCLUDE AN EMPTY STANDARDS ENFORCEMENT BLOCK HERE FOR             */
 /* COMPATIBILITY WITH THE OTHER ACS ROUTINES                         */
 /*********************************************************************/

 /*********************************************************************/
 /* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK                        */
 /*********************************************************************/


        WHEN ((&UNIT EQ &VALID_DASD_UNIT)
        AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))

            SELECT /* DASD DATA ALLOCATION */
```

Figure 98 (Part 2 of 7). Management Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* EXCLUDE DFSMSHSM RECALLS AND RECOVERIES FROM ACS PROCESSING IF    */
/* A MANAGEMENT CLASS IS ALREADY ASSIGNED TO THE DATA SET            */
/*********************************************************************/

              WHEN ((&ACSENVIR = 'RECALL' OR &ACSENVIR ='RECOVER')
              AND (&MGMTCLAS NE ''))

                  DO
                      SET &MGMTCLAS = &MGMTCLAS
                      EXIT
                  END /* DO */


/*********************************************************************/
/* TAILOR AND ADD JCL EXCEPTIONS HERE IF THEY ARE USED               */
/*********************************************************************/


/*********************************************************************/
/* TAILOR AND SET MGMTCLAS FOR ALLOCATION EXCEPTIONS                 */
/*********************************************************************/
/* REPEAT THIS BLOCK FOR EACH SET OF SPECIAL USERS THAT ARE          */
/* ALLOWED TO REQUEST SMS SERVICE THROUGH MGMTCLAS AT ALLOCATION     */
/*********************************************************************/

              WHEN ((&USER EQ &SPECIAL_MC_USER)
              AND (&MGMTCLAS EQ &SPECIAL_MGMTCLAS))

                  DO
                      SET &MGMTCLAS = &MGMTCLAS
                      EXIT
                  END /* DO */



/*********************************************************************/
/* TAILOR AND ADD DATA SET LIST EXCEPTIONS HERE IF THEY ARE USED     */
/* INCLUDE CONCURRENT COPY                                           */
/*********************************************************************/




/*********************************************************************/
/* SYSTEM TEMPORARY DATA SETS DO NOT HAVE MANAGEMENT CLASSES         */
/*********************************************************************/
```

Figure 98 (Part 3 of 7). Management Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD TEMP SHORT DURATION HERE IF IT EXISTS              */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD TSO LOGON AND LIST SUBTYPES IF THEY HAVE           */
/* MANAGEMENT CLASSES DIFFERENT THAN OTHER TSO DATA                  */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD OTHER TSO SUBTYPES HERE IF THEY EXIST              */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD TEST SUBTYPES HERE IF THEY EXIST
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD BATCH PRODUCTION SUBTYPES HERE IF THEY EXIST       */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND ADD ONLINE PRODUCTION SUBTYPES HERE IF THEY EXIST      */
/*********************************************************************/
```

Figure 98 (Part 4 of 7). Management Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND SET MGMTCLAS FOR OTHER SUBTYPES                        */
/*********************************************************************/
/* REPEAT THIS BLOCK FOR EACH OTHER SUBTYPE THAT HAS A DIFFERENT     */
/* MANAGEMENT CLASS                                                  */
/*********************************************************************/

              WHEN (&DSN EQ &HSM_BKCONT1_DATA_DSN)

                  DO
                      SET &MGMTCLAS ='MC54NMIG'
                      EXIT
                  END /* DO */


              WHEN (&DSN EQ &HSM_ACT_DATA_DSN)

                  DO
                      SET &MGMTCLAS ='MC54WORK'
                      EXIT
                  END /* DO */

              WHEN (&DSN EQ &HSM_LOGS_DATA_DSN)

                  DO
                      SET &MGMTCLAS ='MC54SPEC'
                      EXIT
                  END /* DO */

              WHEN (&DSN EQ &HSM_ABARS_DATA_DSN)

                  DO
                      SET &MGMTCLAS ='MC54PRIM'
                      EXIT
                  END /* DO */



/*********************************************************************/
/* TAILOR AND ADD OTHER SUBTYPES HERE IF THEY EXIST                  */
/*********************************************************************/



/*********************************************************************/
/* TAILOR AND SET MGMTCLAS FOR ALL OTHER SMS-MANAGED SUBTYPES        */
/*********************************************************************/

              OTHERWISE /* UNEXPECTED DASD DATA */

                  DO
                      SET &MGMTCLAS = 'MC54NMIG'
                      EXIT
                  END /* DO */

          END /* DASD ALLOCATION SELECT */
```

Figure 98 (Part 5 of 7). Management Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK                        */
/*********************************************************************/

       WHEN (&UNIT EQ &VALID_TAPE_UNIT)

           SELECT /* TAPE ALLOCATIONS */


/*********************************************************************/
/* TAILOR AND ADD TAPE MOUNT MANAGEMENT SUBTYPES HERE IF THEY EXIST  */
/*********************************************************************/




/*********************************************************************/
/* TAILOR AND ADD SMS-MANAGED TAPE SUBTYPES HERE IF THEY EXIST       */
/*********************************************************************/




/*********************************************************************/
/* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT  */
/* OR SMS-MANAGED TAPE EXIST                                         */
/*********************************************************************/


               WHEN (&UNIT EQ &VALID_TAPE_UNIT)
               SET &MGMTCLAS = &MGMTCLAS


           END /* TAPE ALLOCATION SELECT */


/*********************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK                     */
/*********************************************************************/

       WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)

           SELECT /* OPTICAL ALLOCATIONS */

               WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
               SET &MGMTCLAS = &MGMTCLAS

/*********************************************************************/
/* TAILOR AND ADD OPTICAL STORE FUNCTION HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/




/*********************************************************************/
/* TAILOR AND ADD OPTICAL TRANSITION AND CHANGE FUNCTIONS HERE IF    */
/* OPTICAL DATA EXISTS                                               */
/*********************************************************************/
```

Figure 98 (Part 6 of 7). Management Class ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* ADD OTHERWISE FOR OPTICAL DATA HERE IF OPTICAL DATA EXISTS        */
/*********************************************************************/



          END /* OPTICAL ALLOCATION SELECT */

/*********************************************************************/
/* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR       */
/* OPTICAL                                                           */
/*********************************************************************/

       OTHERWISE /* UNEXPECTED DEVICES */

           DO
               SET &MGMTCLAS = 'MC54NMIG'
               EXIT
           END /* DO */

   END /* SELECT */

END /* PROC */
```

Figure 98 (Part 7 of 7). Management Class ACS Routine for DFSMShsm Data

A.3 Sample Storage Group ACS Routine

Figure 99 on page 255 shows sample code that we used in our storage group ACS routine. You may need to create another storage group and update your storage group ACS if you want to separate the DFSMShsm data sets from other data.

```
PROC &STORGRP

 /*********************************************************************/
 /* START OF STORAGE GROUP PROC                                       */
 /*********************************************************************/
 /*  CHANGE HISTORY                                                   */
 /*  DATE     PERSON  DESCRIPTION OF CHANGE                           */
 /*********************************************************************/
 /* 95/05/06   DEF    INITIAL IMPLEMENTATION FOR STAGE I              */
 /*                   DATA                                            */
 /* 95/02/05   DEF    MINIMAL CONFIGURATION                           */
 /*********************************************************************/
 /* YOU NEED TO MOVE ALL THE FILTLISTS TO THE                         */
 /* LOCATION BETWEEN THE TWO LINES OF SPECIAL CHARACTERS SHOWN        */
 /* BELOW THIS COMMENT BEFORE USING THE CODE FRAGMENTS                */
 /*********************************************************************/

/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */

        FILTLIST &OMVS_DATA_DSN INCLUDE
        (OMVS*.**)

/*********************************************************************/

        FILTLIST &HSM_NONSMS_DATA_DSN INCLUDE
        (HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,
         HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)

/*********************************************************************/

        FILTLIST &HSM_CONTRDS_DATA_DSN INCLUDE
        (HSM.%CDS.*,HSM.JRNL.*)

/*********************************************************************/

        FILTLIST &HSM_LOGS_DATA_DSN INCLUDE
        ('HSM.EDITLOG',HSM.HSMLOG*,HSM.HSMPDO*)

/*********************************************************************/

        FILTLIST &HSM_ACT_DATA_DSN INCLUDE
        (HSMACT.H%.*.D*.T*)

/*********************************************************************/

        FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
        (HSM.%CDS.BACKUP.*,
         HSM.JRNL.BACKUP.*)

/*********************************************************************/

        FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
        (ABARS.*.INSTRUCT)
```

Figure 99 (Part 1 of 5). Storage Group ACS Routine for DFSMShsm Data

```
/*********************************************************************/

        FILTLIST &SMS_MANAGED_OMVS_DATA_HLQ INCLUDE
        (OMVS*)

        FILTLIST &SPECIAL_SC_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_SCNOSMS_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_STORCLAS INCLUDE
        ('SC54GRT','SC54STD','SC54LOW')

        FILTLIST &SPECIAL_MC_USER INCLUDE
        ('MHLRES3','MHLRES4','MHLRES5')

        FILTLIST &SPECIAL_MGMTCLAS INCLUDE
        ('MC54SPEC')

/*********************************************************************/


        FILTLIST &VALID_DASD_UNIT INCLUDE
        ('3380','3390','SYSDA','SYSALLDA','')

        FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
        ('STORE','CHANGE','CTRANS')

        FILTLIST &VALID_TAPE_UNIT INCLUDE
        ('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')

/*********************************************************************/


/** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **/

    SELECT /* MAIN LOGIC BLOCKS */

 /*******************************************************************/
 /* INCLUDE AN EMPTY STANDARDS ENFORCEMENT BLOCK HERE FOR           */
 /* COMPATIBILITY WITH THE OTHER ACS ROUTINES                       */
 /*******************************************************************/

 /*******************************************************************/
 /* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK                      */
 /*******************************************************************/

        WHEN ((&UNIT EQ &VALID_DASD_UNIT)
        AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))

            SELECT /* DASD DATA ALLOCATION */
```

Figure 99 (Part 2 of 5). Storage Group ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD GUARANTEED SPACE LOGIC HERE IF GUARANTEED SPACE IS */
/* USED                                                              */
/*********************************************************************/
       WHEN ((&USER EQ &SPECIAL_SC_USER)
       AND (&STORCLAS EQ 'SC54GRT'))
         DO
           SET &STORGRP EQ 'SG54HSM'
           EXIT
         END
/*********************************************************************/
/* TAILOR AND ADD DATA SET LIST EXCEPTIONS HERE IF THEY ARE USED     */
/* INCLUDE SEQUENTIAL STRIPING AND CONCURRENT COPY                   */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD EITHER TEMP AND VIO OR TEMP AND NO VIO HERE IF     */
/* TEMPORARY DATA IS SMS-MANAGED                                     */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD TEMP SHORT DURATION HERE IF THERE IS TEMP SHORT    */
/* DURATION DATA AND IF IT IS SMS-MANAGED                            */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD TSO LOGON AND LIST SUBTYPE IF THEY HAVE STORAGE    */
/* GROUPS DIFFERENT THAN OTHER TSO SUBTYPES AND THEY ARE SMS-MANAGED */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD OTHER TSO SUBTYPES HERE IF THEY ARE SMS-MANAGED    */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD TEST SUBTYPES HERE IF THEY ARE SMS-MANAGED         */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD BATCH PRODUCTION SUBTYPES HERE IF THEY ARE         */
/* SMS-MANAGED                                                       */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD ONLINE PRODUCTION SUBTYPES HERE IF THEY ARE        */
/* SMS-MANAGED                                                       */
/*********************************************************************/
```

Figure 99 (Part 3 of 5). Storage Group ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* TAILOR AND ADD OTHER SUBTYPES HERE IF THEY ARE SMS-MANAGED        */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND SET STORGRP FOR ALL OTHER SMS-MANAGED SUBTYPES         */
/*********************************************************************/

              WHEN (&DSN EQ &OMVS_DATA_DSN)

                  DO
                      SET &STORGRP ='OPENMVS'
                  END /* DO */

              OTHERWISE /* UNEXPECTED DASD DATA */

                  DO
                      SET &STORGRP = 'SG54HSM'
                      EXIT
                  END /* DO */

          END /* DASD ALLOCATION SELECT */

/*********************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK                        */
/*********************************************************************/

       WHEN (&UNIT EQ &VALID_TAPE_UNIT)

           SELECT /* TAPE ALLOCATIONS */

/*********************************************************************/
/* TAILOR AND ADD TAPE MOUNT MANAGEMENT SUBTYPES HERE IF THEY EXIST  */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD SMS-MANAGED TAPE SUBTYPES HERE IF THEY EXIST       */
/*********************************************************************/

/*********************************************************************/
/* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT  */
/* OR SMS-MANAGED TAPE EXIST                                         */
/*********************************************************************/

           END /* TAPE ALLOCATION SELECT */
```

Figure 99 (Part 4 of 5). Storage Group ACS Routine for DFSMShsm Data

```
/*********************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK                     */
/*********************************************************************/

       WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)

           SELECT /* OPTICAL ALLOCATIONS */

               WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
               SET &STORGRP = &STORGRP

/*********************************************************************/
/* TAILOR AND ADD OPTICAL STORE FUNCTION HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD OPTICAL TRANSITION FUNCTION HERE IF OPTICAL DATA   */
/* EXISTS                                                            */
/*********************************************************************/

/*********************************************************************/
/* ADD OTHERWISE FOR OPTICAL DATA HERE IF OPTICAL DATA EXISTS        */
/*********************************************************************/

           END /* OPTICAL ALLOCATION SELECT */

/*********************************************************************/
/* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR       */
/* OPTICAL                                                           */
/*********************************************************************/

       OTHERWISE /* UNEXPECTED DEVICES */

           DO
               SET &STORGRP = 'SG54HSM'
               EXIT
           END /* DO */

   END /* SELECT */

END /* PROC */
```

Figure 99 (Part 5 of 5). Storage Group ACS Routine for DFSMShsm Data

Appendix B. Special Notices

This publication is intended to help storage administrators and system programmers implement DFSMShsm. It also provides an update to the latest level of DFSMShsm and details how to implement new and enhanced functions. The information in this publication is not intended as the specification of any programming interfaces that are provided by DFSMShsm. See the PUBLICATIONS section of the IBM Programming Announcement for DFSMS/MVS Version 1 Release 4 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

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, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of

including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
Enterprise Systems Architecture/390
ESCON
IBM
MVS/DFP
MVS/ESA
NetView
OpenEdition
OS/390
Parallel Sysplex
RACF
RAMAC
S/390
SNAP/SHOT
System/390
VTAM
ACF/VTAM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 265.

- *DFSMS Optimizer: The New HSM Monitor/Tuner*, SG24-5248
- *DFSMShsm ABARS and Mainstar Solutions*, SG24-5089
- *IBM Magstar 3494 Tape Libraries: A Practical Guide*, SG24-4632
- *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268
- *Implementing Concurrent Copy*, GG24-3990

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

C.3 Other Publications

These publications are also relevant as further information sources:

- *DFSMShsm Storage Administration Guide*, SH21-1076
- *DFSMShsm Storage Administration Reference*, SH21-1075
- *DFSMShsm Implementation and Customization Guide*, SH21-1078
- *DFSMShsm Diagnosis Guide*, LY27-9607
- *DFSMS/MVS V1R4 Access Method Services for ICF*, SC26-4906
- *DFSMS/MVS V1R4 OAM PISA*, SC26-3051
- *DFSMSrmm Implementation and Customization Guide*, SC26-4932
- *DFSMSdfp Storage Administration Reference*, SC26-4920
- *DFSMS/MVS Installation Exits*, SC26-4908
- *DFSMSdss Storage Administration Reference*, SC26-4929
- *DFSORT R14 Application Programming Guide*, SC33-4035

- *DFSMS/MVS Using the Volume Analyzer*, SC26-4925
- *dpAM R2 Collect Feature Customization and Reference*, SH19-6758

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download or order hardcopy/CD-ROMs redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders via e-mail including information from the redbook order form to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755	
Canada (toll free)	1-800-IBM-4YOU	
Outside North America	(long distance charges apply)	
(+45) 4810-1320 - Danish	(+45) 4810-1220 - French	(+45) 4810-1270 - Norwegian
(+45) 4810-1420 - Dutch	(+45) 4810-1020 - German	(+45) 4810-1120 - Spanish
(+45) 4810-1540 - English	(+45) 4810-1620 - Italian	(+45) 4810-1170 - Swedish
(+45) 4810-1670 - Finnish		

This information was current at the time of publication, but is continually subject to change. The latest information for customers may be found at http://www.redbooks.ibm.com/ and for IBM employees at http://w3.itso.ibm.com/.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may also view redbook, residency and workshop announcements at http://inews.ibm.com/.

IBM Redbook Fax Order Form

Fax your redbook orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

ABARS	aggregate backup and recovery support
ABEND	abnormal end
ACDS	active control data set
ACS	automatic class selection
AMS	access method services
APF	authorized program facility
ATL	automated tape library
BCDS	backup control data set
CA	control area
CDS	control data set
CI	control interval
CLIST	command list
CSA	common storage area
CST	cartridge storage tape
DAE	dump analysis elimination
DASD	direct access storage device
DBU	delete-if-backed-up
DD	data definition
DFSMS	Data Facility Storage Management Subsystem
DSR	daily statistics record
ECCST	Extended Capacity Cartridge Storage Tape
ECSA	extended common service area
EOV	end of volume
FMID	function modification identifier
FSR	function statistics record
FTP	File Transfer Protocol
GDG	generation data group
GDS	generation data set
GRS	global resource sharing
GTF	generalized trace facility
GUI	graphical user interface
HDA	head disc assembly
HSM	Hierarchical Storage Manager
IBM	International Business Machines Corporation
ICF	integrated catalog facility
IDRC	improved data recording capability
IP	internet protocol
IPCS	interactive problem control system
IPL	initial program load
ISMF	interactive storage management facility
ISV	independent software vendor
ITSO	International Technical Support Organization
JCL	job control language
JES	job entry subsystem
KSDS	key sequenced data set
LAN	local area network
MCB	BCDS data set record
MCDS	migration control data set
ML1	migration level 1
ML2	migration level 2
MTL	manual tape library
MWE	management work element
MVS	Multiple Virtual Storage
OCDS	offline control data set
OEM	original equipment manufacturer
PDA	problem determination aid
PDS	partitioned data set
PDSE	partitioned data set extended
PSCB	protected step control block
PTF	program temporary fix
PU	physical unit
RACF	resource access control facility
RAID	redundant array of independent disks
REXX	Restructured Extended Executor
RLS	record level sharing
RVA	RAMAC Virtual Array
SDM	system data mover
SDSP	small data set packing
SMF	system management facility

SMP/E	System Modification Program Extended
SMS	system-managed storage
SSID	subsystem identifier
STK	StorageTek Corporation
TCP/IP	Transmission Control Protocol/Internet Protocol
TMM	tape mount management
TMP	terminal monitor program
TSO	time sharing option
TTOC	tape table of content
UDP	user datagram protocol
VRS	vital record specification
VSAM	virtual storage access method
VSR	volume statistics record
VTOC	volume table of contents
VTS	virtual tape server
WSDS	working space data set
XCF	cross-system coupling facility

Index

A

B

C

D

E

O

P

Q

R

S

T

V

W

ITSO Redbook Evaluation

DFSMShsm Primer
SG24-5272-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and Fax it to: USA International Access Code + 1 914 432 8264 or:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer __Business Partner __Solution Developer __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

_____

SG24-5272-00
Printed in the U.S.A.

SG24-5272-00