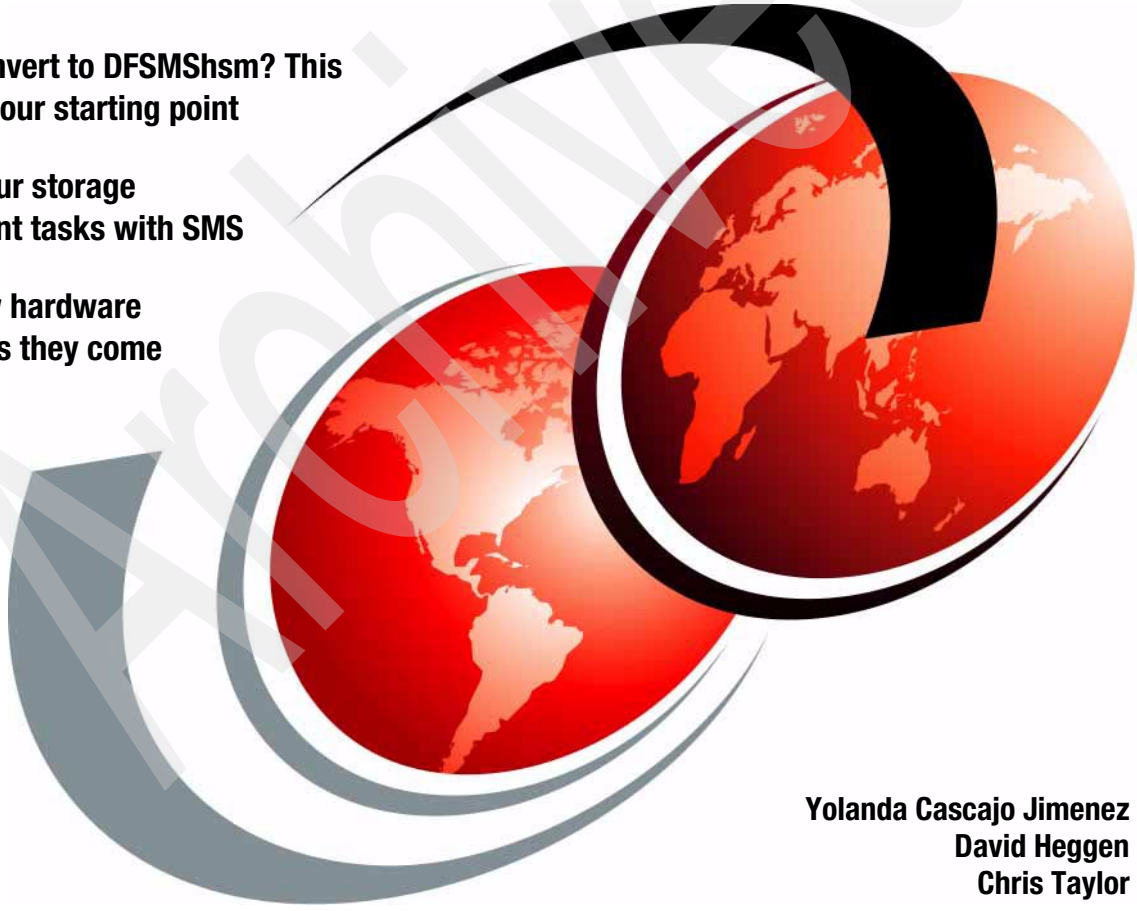
IBM

Converting to DFSMShsm:

A Practical Approach

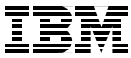
- Want to convert to DFSMShsm? This should be your starting point
- Simplify your storage management tasks with SMS
- Exploit new hardware functions as they come



Yolanda Cascajo Jimenez
David Heggen
Chris Taylor

Redbooks

ibm.com/redbooks



International Technical Support Organization

Converting to DFSMShsm: A Practical Approach

February 2002

SG24-6524-00

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page xv.

First Edition (February 2002)

This edition applies to Version 1, Release 2 of z/OS DFSMS, Program Number 5694-A01.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 471F Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Figures

Tables

Examples

Special notices

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

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)®
Redbooks (logo)™ 
Approach
CICS
Current
DB2
DFS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
Early
IBM
IMS
MVS
OS/390
Parallel Sysplex
RACF
RAMAC
RMF
S/390
Sequent
SP
Tivoli
z/OS

Other company trademarks

The following terms are trademarks of other companies:

FDR, ABR, IAM, are trademarks of Innovation Data Processing in the United States and/or other countries.

CA-1, CA-ACF2, CA-FAVER, CA-MIM, are trademarks of Computer Associates International, Inc. in the United States and/or other countries.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

The intended audience for this IBM Redbook are customers that have achieved partial implementation of the IBM DFSMS product suite and are looking to further their implementation through the use of the DFSMShsm product. It should be reassuring to know that many other customers have converted from products such as FDR/ABR, CA-ASM2 or CA-Disk to DFSMShsm. It is also important to acknowledge that such conversions are not completed overnight, and that the customer's personnel will need to be intensively involved in their conversion project from its conception.

Most conversions require a period of product coexistence to allow the migration to proceed. This provides time for the aging of the prior product's archive/backup data and reduces the data migration requirements. In some circumstances, it may be possible to hasten the conversion by installing enough disk to allow pre-restoration of data to subsequently backup/migrate with DFSMShsm.

This book will take you through everything we have learned about performing these conversions. We cover the elements of preparation, activation, conversion, and exploitation, using examples to illustrate these activities from projects that we have performed.

Storage administrators will find in this book much valuable information to use in their conversion planning activities.

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.

Yolanda Cascajo Jimenez is an Enterprise Storage Software Project Leader at the International Technical Support Organization, San Jose Center. She has 8 years of experience in storage systems, including DASD, tapes, and storage management software. Before joining the ITSO in year 2000, Yolanda worked as I/T Specialist in Spain, acting as pre-sales technical support and consultant in storage systems. She has worked at IBM for 13 years.

David Heggen is a Certified IT Specialist working for IBM SSD Advanced Technical Support in San Jose, CA USA. He has 24 years of experience in Data Processing, starting as an Application Programmer. He has worked at IBM for 19 years, starting as an Systems Programmer and accepting assignments as a Systems Engineer, IT Specialist in Global Services, and is currently working in Pre/Post Storage Sales Support. His areas of expertise include z/OS Storage Management (including involvement in 9 different conversion projects), data migration, device/media conversion and disaster/recovery solutions.

Chris Taylor is a Senior IT Specialist working for the IBM S/390 Tivoli Migration Team (TMT) specializing in Storage solutions migrations in the Americas. He has 20 years of experience in enterprise systems, including 12 years in Germany and the United Kingdom. His duties have included operations, system programming, pre-sales and product development support for disk and tape storage management offerings from several vendors. His current assignment involves converting customers from competitive products to IBM storage software solutions, including DFSMShsm, DFSMSdss, and DFSMS Optimizer.

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

Tony Pearson
IBM Storage Systems Software Architecture and Technical Strategy

Max Kawano
IBM Tivoli Migration Team

Thomas Conley
Pinnacle Consulting Group, Inc.

We also want to thank many customers for their support and help in providing the information we have tried to assemble in this redbook. We would like to especially thank:

Michael Friske
Alan Richardson
Patti Trant
Lynda Tilley

Notice

This publication is intended to help storage administrators plan a conversion from a competitive storage management product to IBM DFSMShsm/DFSMSdss, with or without the SMS implementation. The information in this publication is not intended as the specification of any programming interfaces that are provided by DFSMShsm or DFSMSdss. See the PUBLICATIONS section of the IBM Programming Announcement for DFSMShsm/DFSMSdss for more information about what publications are considered to be product documentation.

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

  `ibm.com/redbooks`

- Send your comments in an Internet note to:

  `redbook@us.ibm.com`

- Mail your comments to the address on page ii.

1

Introduction

This chapter provides basic DFSMShsm information to acquaint you with the available functions and the basic terminology. It also introduces the DFSMS design philosophy to help you understand the way DFSMS works.

At the end of this chapter you should have an overall view of the IBM storage management philosophy and the role of DFSMShsm in this environment.

1.1 DFSMS functional components

The DFSMS product has been undergoing evolution almost since its introduction. The constituent components have been developed over the years and include utilities, binder, access methods, device support, and catalog. DFSMS is an integral part of the z/OS operating system. In its present definition, it consists of:

- DFSMSdfp, including:
  - NaviQuest
  - Catalog Search Interface (CSI)
  - Distributed FileManager (DFM) support
- DFSMSdss
- DFSMShsm
- DFSMSrmm

Additional DFSMS features and supporting products are:

- DFSMS Optimizer
- Network File System

It is the design intent that the DFSMS product suite provide support for storage administration and management without inhibiting the growth of the data center. Customers that implement full function DFSMS receive the following benefits:

- Consistent/automated enforcement of storage management policies
- Policies made public via viewing DFSMS constructs
- Elimination of requirement for esoteric names to support allocation
- Control of “out-of-space” related abends:
  - Using storage group allocation thresholds
  - Using spill volumes
  - Separation of small/large allocations
  - IDCAMS **ALTER ... ADDVOLUMES(*)** command for additional space allocation
  - DFSMShsm extent reduction
  - DFSMShsm interval migration
  - Selection of secondary space amount
  - Use of DFSMSdss defragmentation
- Use of the extended format data sets, allowing:
  - Sequential data striping
  - Compression
  - PDSE
  - VSAM KSDS > 4 GB
  - VSAM Record Level Sharing

1.2 DFSMShsm functions

DFSMShsm provides functions for the following capabilities

- **Storage management:** DFSMShsm uses a hierarchy of storage devices in its automatic management of data, relieving end-users from manual storage management tasks.
- **Space management:** DFSMShsm improves disk space utilization by keeping only active data on fast-access storage devices. It automatically frees space on user volumes by deleting eligible data sets, releasing over-allocated space, and moving low-activity data to lower-cost-per-byte devices.
- **Availability management:** DFSMShsm backs up your data — automatically or by command -- to ensure availability in the event of accidental loss of data sets or physical loss of volumes. DFSMShsm also allows a storage administrator to copy backup and migration tapes. These copies can be stored on site as protection from media damage, or off site as protection from site damage. Disaster backup and recovery is also provided for user-defined groups of data sets (aggregates), so that critical applications can be restored at the same location or at an offsite location.
- **Tape mount management:** DFSMShsm can write multiple output data sets to a single tape, making it a useful tool for implementing tape mount management (TMM) methodology under SMS. When you redirect tape data set allocations to disk, DFSMShsm can move those data sets to tape, as a group, during interval migration. This greatly reduces the number of tape mounts initiated on the system. DFSMShsm uses single file format, which also improves your tape usage and search capabilities.

Figure 1-1 shows an overview of the functions that are included in space and availability management.

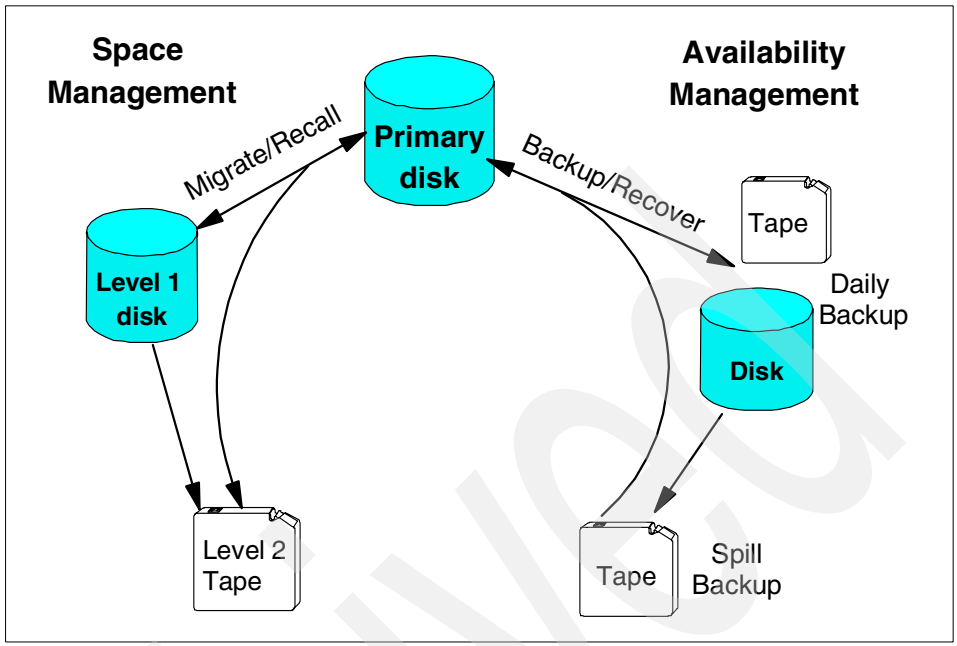


Figure 1-1 DFSMShsm overview

Full exploitation of DFSMShsm functions in a system-managed storage environment requires the use of DFSMSdss for certain functions. You can also use the DFSMS Optimizer feature to monitor and tune DFSMShsm functions.

1.2.1 Product terminology

In this section we introduce some of the DFSMShsm common used terms, for a better understanding of the DFSMShsm environment.

Some terms common to most of the DFSMShsm activities are:

Duplex Creation of a tape copy (either ML2 or backup) simultaneous with the original tape.

Level 0 Also referred to as primary disk. These are the volumes defined to the SMS storage groups that DFSMShsm manages. From these volumes we migrate/recall, and backup/recover data sets. The only allowed target of a recall or recover will be a level 0 volume.

Recycle A DFSMShsm command to improve tape utilization as data sets expire, get deleted, backup versions roll off, and so on. The expected benefit of recycle is that DFSMShsm will use fewer tapes and more scratch tapes will be available.

SMS System Managed Storage. This is the allocation manager component of DFSMSdfp that is available on all z/OS platforms. The user interface for SMS is provided by the ISMF application.

Figure 1-2 shows terms used when referring to availability management activities.

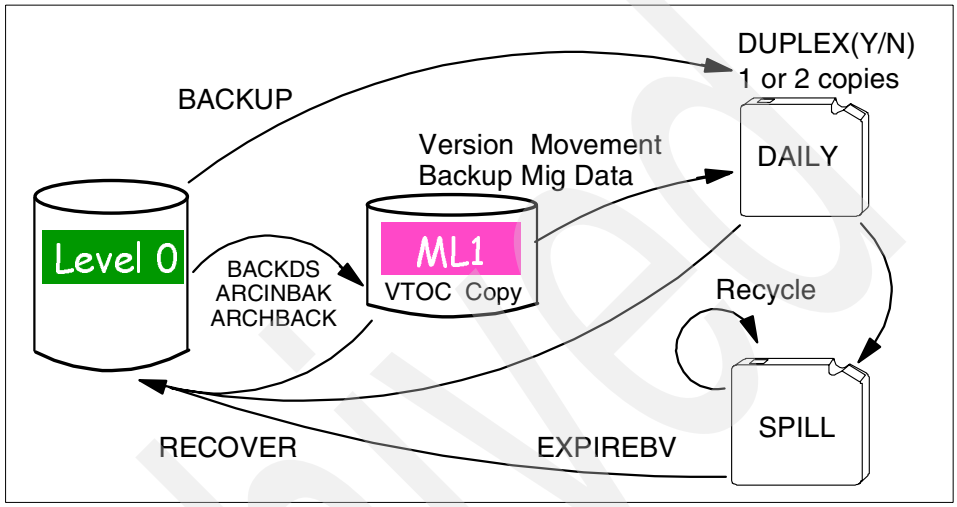


Figure 1-2 Availability management activities

Backup The creation of a copy of a data set either automatically, by TSO command or by the supplied program ARCINBAK. Multiple copies (versions) of a data set can exist at the same time. These copies represent the data set at various stages over time.

Daily backup tape Tapes that are used to contain a backup for a specific “day” as defined by the backup cycle.

EXPIREBV Command provided to remove DFSMShsm backups of data sets that have been deleted.

Spill backup tape Backup tape that are not assigned to a specific day. Can be used to supplement daily backup tapes or be the target of recycle output.

Recover The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

Figure 1-3 explains the role of ABARS with respect to availability management.

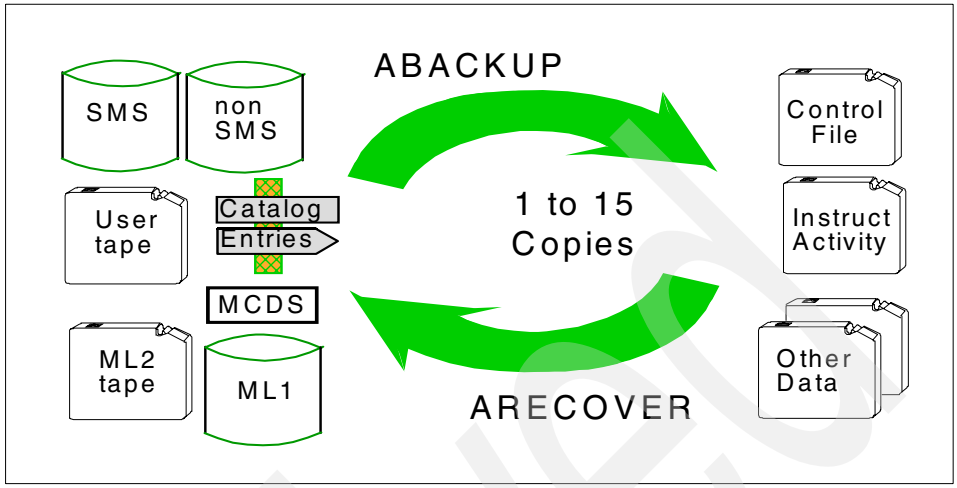


Figure 1-3 ABARS Activity

ABARS	Aggregate Backup and Recovery Support
Aggregate	Collection of data sets, typically a group of data sets comprising an application.
ABACKUP	Creation of an aggregate backup (self describing)
ARECOVER	Restoration of the data contained in the aggregate
Instruct/activity logs	Information considered necessary to assist with the recovery process at the recovery site. These are optional files.

Figure 1-4 explains terms used when referring to space management activities.

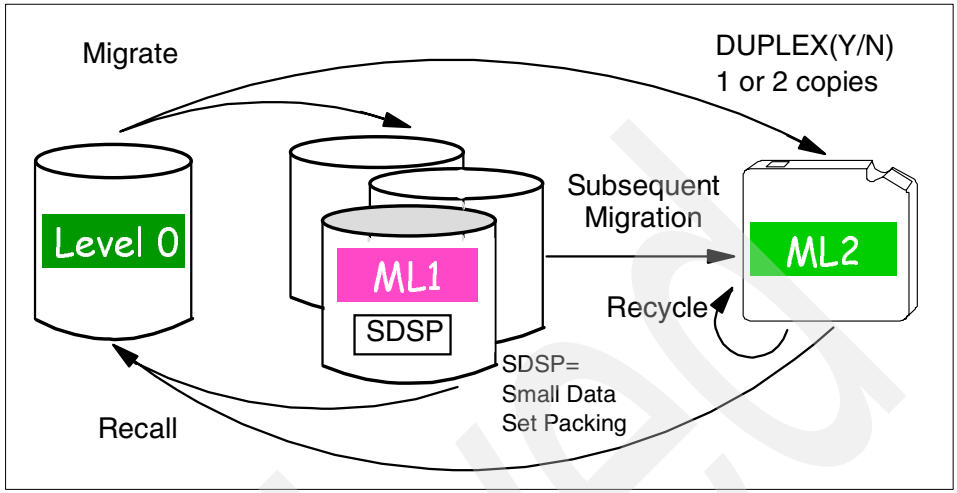


Figure 1-4 Space management activities

Migrate This refers to the movement of a data set in the storage hierarchy to a lower cost storage medium. There is only one copy of a data set that is migrated and there is a catalog entry for this data set on volume MIGRAT. Data sets can migrate manually via TSO command or automatically when data set age and volume occupancy thresholds are met.

ML1 Migration Level 1. A level on the storage hierarchy defined to typically compacted disk. Data sets can get migrated to/from or recalled from this level. Data sets cannot get recalled to this level.

ML2 Migration Level 2. The last level of the storage hierarchy defined to use typically high capacity tape. Data can only be migrated to or recalled from this level. It is also possible to migrate directly to ML2 from level 0, bypassing ML1.

Recall Restoration of a migrated data set from ML1 or ML2 to level 0 disk.

SDSP Small Data Set Packing. An optional element of ML1 to accommodate customers that have many small data sets to migrate. This provides a mechanism to place more than one data set on a disk track. Physically, SDSPs are supported by a VSAM KSDS allocated on ML1 volumes.

Subsequent migration The activity performed in DFSMShsm by secondary space management. This specifically refers to the movement of a data set from ML1 to ML2.

We will introduce more terms as we go into more detail on the product.

1.2.2 Performing tasks

In DFSMShsm you have various ways of performing requests:

- ISMF panels
- TSO commands
- Macros
- Inline backup facility

ISMF panels

Interactive Storage Management Facility is an ISPF application that simplifies storage management tasks. Though ISMF panels and line operators you are able to define and manage your storage management policies. The input to ISMF can be used to generate the necessary commands to perform a requested task.

TSO commands

You may also request DFSMShsm services issuing TSO commands. There are commands that any user is able to issue (known as user commands), as well as other commands which are restricted to authorized storage administrators. These authorized users can be system programmers, storage administrators, disaster/recover administrators, or anyone with the need to restore or move data in the hierarchy beyond the scope of an ID or application.

For more information about user commands, refer to *DFSMShsm Managing Your Own Data*. For more information about storage administrator commands, refer to *DFSMShsm Storage Administration Guide*.

Macros

DFSMShsm provides macros for allowing you to request DFSMShsm services from your application programs. The following macros are supported:

- ARCFMWE frees up storage in the common storage area (CSA)
- ARCHBACK backs up a specific data set
- ARCHBDEL deletes backup version(s) of a data set
- ARCHDEL deletes a migrated data set
- ARCHMIG migrates a specific data set
- ARCHRCAL recalls a data set
- ARCHRCOV recovers a data set
- ARCHSEND sends a command to DFSMShsm
- ARCXTRCT extracts data from DFSMShsm

For more information about these macros, refer to *DFSMShsm Managing Your Own Data*.

1.3 DFSMS design philosophies

There are philosophical differences between the design of the IBM products and other storage management products. Discussions with customers either considering or actively converting to DFSMShsm provide us these observations. In all cases we will assume that the desired allocation manager to implement is provided by DFSMSdfp. This will require some customers considering conversion to re-think their present environment, as this expands the role that DFSMS performs on their behalf. For other customers, this represents no change at all.

These are some of the philosophical differences:

- All fields in the SMS class definitions are active:
  - This is where storage management policies are defined.
  - With implementation of DFSMShsm, this is especially true of MGMTCLAS, and will involve revisiting the present MGMTCLAS specifications, particularly in the areas of *GDG early migration* and *backup parameters*. We want to avoid a situation in which these parameters will take effect when DFSMShsm processes the data. Instead, we intend to make the behavior of DFSMShsm operation planned and intentional.
- All SMS managed data sets are to be cataloged and located via standard catalog search.

  This means that you are willing to treat an uncataloged data set as an error. The behavior of jobs in your system will change. Rather than getting a **`not catlg 2`** error at the *end* of a step, you will now get a duplicate data set name, and the job will get a JCL error at the *beginning* of the step. Most customers like this behavior and have little problem adjusting to it. This is a DFSMS requirement in general, as well as being an absolute requirement for DFSMShsm to manage the same data as the other storage management product (co-existence).
- If the user codes a parameter for new data set allocation, with few exceptions, it is not overridden.

  Allocation management products, other than SMS, tout their ability to allow installations to enforce standards and actively change JCL on behalf of the users. SMS allows override of six parameters in JCL: DATACLAS, MGMTCLAS, STORCLAS, UNIT, VOLSER, and EXPDT.

Since DFSMS 1.4 (around 1998), the ability to perform *space constraint relief* has been available for VSAM extended format data sets. This allows the change of the requested space to prevent allocation failure, rather than to enforce standards. Philosophically, DFSMS considers user requested parameters as being more correct than installation directed overrides. This does *not* imply that DFSMS cannot provide parameters when they are not coded by the user. The best example of this case is VOLCOUNT supplied by DATACLAS. When VOLCOUNT is supplied in DATACLAS and is not coded by the user, this effectively allows the data set to extend to multi-volume without changing any JCL.

- DFSMShsm has no concept of backup with permanent retention:
  - Backups in this sense are the successful result of ABSTART, ARCINBAK, or HBACKDS commands.
  - Other methods exist to create a permanent backup.

  In some customer installations, the storage management product is selected to perform permanent or archival backups. In some cases, this is a requirement identified after the backup is taken and comprises a re-designation of the retention date for the backup.

  Backup data in DFSMShsm is managed by the fact that the data set exists, as well as data set changes, number of versions desired, backup frequency, and optionally guaranteed backup frequency.

  If, for example, the number of versions is two, there is no mechanism available to prevent the oldest generation from being deleted if there are two backup versions in existence and the original data set has the data-set-changed flag set on.

  Strategically, customers desiring a permanent backup would be best served by using a system supplied utility like IEBGENER, IEBCOPY, REPRO, or EXPORT to create the backup, in order to have the best opportunity to read and recover the data set at an unspecified future date. Who knows what products will be available and supported in 20 years?

  However, to have the ease of use or performance characteristics of a storage management product in performing this backup, we would advise either using DFSMSdss logical dump, or copying the data set to another data set such as **`MY.DATASET.BACKUP.J2001324`** and letting the backup be a permanent data set that DFSMShsm can now manage. The advantage of copying the data set rather than backing it up is that the backup is still available via the HLIST command. You can see it by issuing the command:

  ```
  HLIST LEVEL(MY.DATASET) BOTH
  ```

  The only retention specified for DFSMShsm backups concerns what happens to backups after the original data set gets deleted.

Some customers will find it a difficult task to translate their storage management product's use of backup expiration dates to the backup versions used by DFSMShsm. Backup versions to DFSMShsm could exist indefinitely (if the original data set never changes and never gets deleted), or they might last no longer than 24 hours. If it is not possible to determine how many backup versions you will need, we would suggest having two versions and addressing the exceptions to this as needed.

- All eligible data sets for migration do not necessarily migrate with DFSMShsm.

  Storage group *low threshold* defines when to stop. The specification of the low threshold on a storage group tells DFSMShsm when it is allowed to stop migrating data from a volume. If allowed to stop before all eligible data sets are migrated, DFSMShsm has provided the free space needed for tomorrow's processing, and has accomplished this using the least amount of system resources. If you want to always have DFSMShsm move all eligible data sets from the managed volumes, you can code a low threshold of zero, but this defeats one of DFSMShsm's design benefits.

- With DFSMShsm, automatic functions provide the most efficient execution and highest throughput.

  By using other storage management products, customers generally see DFSMShsm's command interface as the easiest way to translate their existing procedures. This may be true, but it is not the best implementation when many thousand data sets are needed to be processed. DFSMShsm's automated functional design is the intended interface for the majority of daily storage management functions. Most DFSMShsm installations perform more than 99% of their storage management functions through the use of the PSMSTART, ABSTART, ADSTART, and SSMSTART commands. The command interface, although highly evolved, is more suited to (and intended for) ad hoc requests.

It is our hope and intent that this documentation will provide insightful help and guidance on your conversion project.

2

Preparing your environment

In this chapter we discuss how to prepare your environment in order to activate DFSMShsm.

The first stage in a successful conversion to DFSMShsm is to understand the complete current storage management environment. This assessment is mandatory and *must* be performed in order to avoid pitfalls later in the conversion process. We consider the areas that require a thorough understanding prior to conversion.

Next we explain SMS constructs and their relationship with DFSMShsm, as well as the initial SMS preparation recommended prior to activation, which is detailed in Chapter 3, “DFSMShsm activation” on page 69. We also discuss some of the features of DFSMS, such as NaviQuest, and explain how it can be used to test either a new SMS implementation or the ongoing ACS and configuration maintenance.

2.1 Installation assessment

In the previous chapter we discussed some of the product philosophies that exist in DFSMShsm. With that in mind, you should evaluate the functions being used in your current storage management product and see if these are possible or even necessary when using DFSMShsm.

These are some of the areas to be examined:

- Current environments:
  - Disk
  - Tape
  - SMS
- Backup requirements and schedules
- Migration (archive) requirements
- Number of tapes and/or data sets containing archive or backup data in the current storage management environment

This information will allow you to develop a detailed conversion plan and ensure a smooth transition to DFSMShsm. Using this information, you will be able to:

- Understand the current archive and backup policies and ensure that these are achieved using DFSMShsm
- Define thresholds in order to remove unreferenced data from primary disk volumes to less-expensive media
- Prepare your current SMS and DFSMShsm environments for transition to the new system
- Ensure that current service levels are maintained
- Define storage groups to be managed by DFSMShsm

2.2 Reviewing the disk storage environment

The disk environment is the first topic for examination. This examines hardware as well as other aspects of the environment, such as pooling.

2.2.1 Pooling

The current disk environment needs to be examined in order to see whether the current policies are effective. These policies may be enforced by using an allocation product like DFSMSdfp, or other products such as CA-Allocate. If an allocation product is not installed, then pooling is achieved using traditional pooling, for example, UNIT=WORK.

Regardless of the pooling currently in use, you should review whether these policies are working as originally intended. Over the period of time, data sets may have been moved to a pool in error and part of the preparation work should be to correct these errors.

If SMS is to be implemented, then certain data set types are not supported and will need to be moved before a volume in a storage group can be converted to SMS. Examples of these data set types include ISAM, unmovable, and data sets protected using OS passwords. For a more complete list of unsupported data set types, please review the manual *DFSMS Implementing System Managed Storage.*

2.2.2 Disk architecture

In addition, this is also a good time to review the disk architecture that is installed. In the case where disks with compression capability are available, we suggest reviewing whether migration to ML1 disk will be beneficial. These devices include RAMAC Virtual Array (RVA), or StorageTek Shared Virtual Array (SVA) devices, for example. In this case, migrating a data set from the primary volume straight to ML2 tape may prove more efficient.

2.2.3 Disk space

At this point you need to review the amount of available disk space which is required for conversion. This includes space for the DFSMShsm control data sets and journal, problem determination data sets (PDA), and ML1 volumes. In addition, additional disk space is required to perform the conversion of the data set versions from the current storage management product. We recommend allowing extra space for these disk volumes, since space problems will slow down the conversion process.

If you have enough room in your current production pools to process the data set versions, the need for a large amount of disk volumes is reduced. There may still be a requirement for some disk volumes, since data sets in your current storage management product may have been archived from a volume which no longer exists. In this case, the data set must be restored to a different volume.

2.2.4 Sharing requirements

Another important disk aspect you will need to review is sharing requirements between systems. The requirements are governed by the number of systems that access data sets managed by DFSMShsm. A data set that resides in a shared catalog should be available on all attached systems and a DFSMShsm started task must be started on these systems in order to allow recalls to take place. For more details about serialization, refer to 2.6, “DFSMShsm in a multiple-image environment” on page 55.

2.3 Reviewing the tape environment

The tape environment is an integral part of the storage management environment and a full understanding of current processing is paramount. DFSMShsm uses tapes for functions such as:

- Backup
- Migration (archive)
- Dump (full volume backup)
- Tape copy
- Recycle
- ABARS processing

All of these functions play a major part in the ability of an organization to have access to data when required, either at the home location or at another site. This is crucial when planning or executing a disaster recovery scenario.

The initial preparation should concentrate on the types of tapes and devices involved and the ability to access the correct drives and capacities. Components of the tape environment needing review include:

- Library environment (tape library or non-library)
- Tape management policy
- Device management policy
- Performance management policy
- Virtual tape

2.3.1 Tape scratch pool selection

When requesting mounts for output processing, DFSMShsm allows the use of *global* scratch pools or *specific* scratch pools. A global scratch pool is a group of scratch tape volumes available for use by any application. In contrast, a specific scratch pool can only be used by DFSMShsm and each volser needs to be added and deleted individually.

Figure 2-1 shows the difference between global and specific scratch pools.

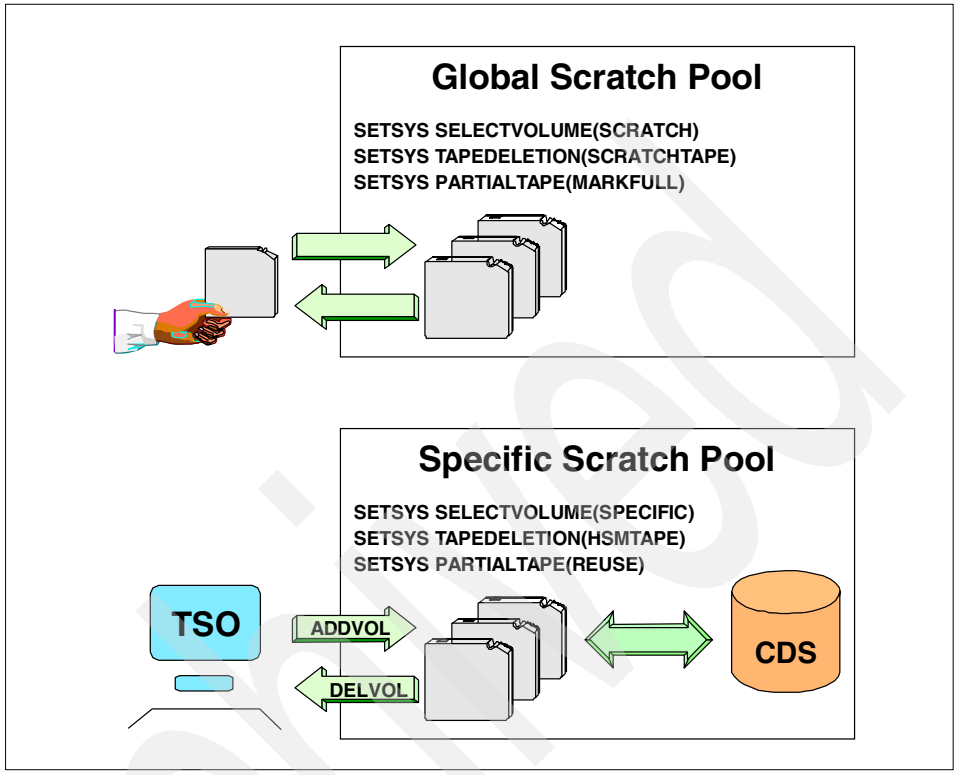


Figure 2-1 Overview of tape scratch pools

We recommend using global scratch pools for all DFSMShsm functions, since this will allow you to utilize the full benefits of a tape management system. It will also allow mount requests to be fulfilled faster through the use of automatic cartridge loaders. The commands coded in the ARCCMDxx member to select from the global scratch pool are:

```
SETSYS SELECTVOLUME(MIGRATION(SCRATCH))
SETSYS SELECTVOLUME(BACKUP(SCRATCH))
SETSYS SELECTVOLUME(DUMP(SCRATCH))
```

To ensure that volumes no longer containing valid data are returned to the global scratch, you should code the following commands:

```
SETSYS TAPEDELETION(MIGRATION(SCRATCHTAPE))
SETSYS TAPEDELETION(BACKUP(SCRATCHTAPE))
SETSYS TAPEDELETION(DUMP(SCRATCHTAPE))
```

2.3.2 Scratch tapes

During the time that DFSMShsm and the other storage management product run alongside each other, there will be a period of time when additional tape scratch volumes are required. The number of tapes will depend on how soon versions are expired from the other product. Refer to Chapter 4, “Converting to DFSMShsm” on page 89, for detailed information on the methods that can be used to convert from one product to another, as well as expiration criteria that can be considered for archive and backup versions.

2.3.3 Tape management system considerations

During the period that a tape contains valid migration, backup or ABARS data, it can be considered to be *managed* by DFSMShsm. Most recognized tape management systems allow tapes to be externally managed, which means that the management of the tape is left entirely to DFSMShsm.

The ARCTVEXT exit provides the method of communicating to the tape management system that DFSMShsm has finished with the tape. The vendors of the various tape management systems provide the version of this exit to be used with their product. The exit needs to be installed into a library in the LINKLIST concatenation and is invoked using the following command:

```
SETSYS EXITON(ARCTVEXT)
```

Note: This exit is not required when using DFSMSrmm as the tape management system. In this case the EDGTVEXT interface is automatically invoked during return processing and no SETSYS parameter is required.

Depending on the tape management product involved, the entry in the options member of the product will look similar to this example. This statement is used to define an External Data Manager (EDM) to CA-1 or CA-Dynam/TLMS:

```
EDM=HSM,DSN=HSM.-,PGM=ARCCTL
```

You may wish to define a separate EDM for the migration, backup and dump functions, as well as the duplex tapes:

```
EDM=HSM1,DSN=-.HMIGTAPE.-,PGM=ARCCTL
EDM=HSM2,DSN=-.BACKTAPE.-,PGM=ARCCTL
EDM=HSM3,DSN=-.DUMP.-,PGM=ARCCTL
```

It is important to remember that EDM interfaces are only called during the process of releasing the tape back to the tape management system. They are not called during tape open processing. If the tape is not flagged as externally managed after DFSMShsm has written to it, the problem is likely a definition problem.

Using DFSMShsm with DFSMSrmm

For retention of DFSMShsm tapes, there are two methods that can be used. In “Expiration date protection” on page 20, we demonstrate how expiration date processing is used to control expiration of a tape. In this case DFSMSrmm will not process the tape until DFSMShsm releases it or is deleted using the DELVOL command.

This retention method requires that MAXRETPD(NOLIMIT) is coded in the DFSMSrmm PARMLIB member EDGRMMxx. This method also requires a way to manage the tapes protected in this way, if you are using a global scratch pool. The VLPOOL command in the DFSMSrmm PARMLIB member EDGRMMxx can be used to automate expiration date protection responses. This allows volumes to be reused without operator intervention and without creating data integrity exposures.

The alternative method is to use DFSMSrmm *vital record specifications* (VRSs). In addition to providing retention policies, these records allow you to specify movement policies and will allow you to define a vaulting policy for your DFSMShsm tapes, as well as for other applications.

For tapes that require no movement, the following VRS examples can be used:

```
RMM ADDVRS DSNAME('mprefix.HMIGTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('bprefix.BACKTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('mprefix.COPY.HMIGTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('bprefix.COPY.BACKTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('authid.**') COUNT(99999) CYCLES
```

In these examples, the following are the values indicated:

mprefix Value specified in SETSYS MIGRATEPREFIX parameter
bprefix Value specified in SETSYS BACKUPPREFIX parameter
authid Value specified in UID in the started task procedure

In addition, we recommend that you specify VRSs to cover data sets that might have been left open by a system failure or may have been open during DFSMSrmm inventory management:

```
RMM ADDVRS DSNAME('ABEND') JOBNAME(hsm_proc) COUNT(99999) CYCLES
RMM ADDVRS DSNAME('OPEN') JOBNAME(hsm_proc) COUNT(99999) CYCLES
```

In this case, **hsm_proc** refers to the DFSMShsm started task name and ensures that volumes are retained by DFSMSrmm until DFSMShsm releases them.

DFSMSrmm Implementation and Customization Guide contains detailed information regarding the interaction between DFSMShsm and DFSMSrmm. Please refer to this manual for information on setting up FACILITY classes or resources in your security product.

2.3.4 Protecting tapes

Protecting DFSMShsm tapes helps ensure that only DFSMShsm is allowed to access its tapes. The implementation method used depends on the security product installed.

RACF protection

If you are using RACF as your security system, then DFSMShsm can add protection to tapes selected for output and remove the protection when the tape is released back to the scratch pool. The commands used to implement this are:

```
SETSYS TAPESECURITY(RACF)
SETSYS TAPESECURITY(RACFINCLUDE)
```

The difference between the two commands is that RACFINCLUDE also allows data sets protected by an OS-password to be processed. The tape volume will not be password protected.

Expiration date protection

Expiration date protection is an option that can be used in a RACF environment as well as with other security products. This allows tapes to be protected by adding an expiration date of 99365 in the IBM Standard Data Set Label 1 (HDR1, EOV1 and EOF1). In an environment with an active tape management system, it is better to use policies defined there, but the expiration date can be used as added protection in case the tape management system is not active. This is activated using the commands:

```
SETSYS TAPESECURITY(EXPIRATION)
SETSYS TAPESECURITY(EXPIRATIONINCLUDE)
```

As in the previous commands, EXPIRATIONINCLUDE allows data sets protected using an OS password to be processed.

Note: Tapes created using EXPIRATIONINCLUDE parameter may require re-initialization before returning to the global scratch pool. This is usually not required when using a tape management system.

Password protection

Password protection adds a password indicator to the IBM Standard Data Set Label 1 (HDR1, EOV1 and EOF1). This method is not normally recommended for a number of reasons. These include:

- Aggregate backup tapes (ABARS) cannot be password protected.
- Tapes may need re-initialization before returning to the global scratch pool.

2.3.5 Review tape duplexing requirements

The DFSMShsm duplex function allows a primary and copy tape to be created concurrently. The primary tape can be kept onsite for recovery purposes while the copy tape can be kept at a secure offsite location. This is the preferred method of creating copy tapes and should be used in preference to the TAPECOPY command.

Duplex tape supports the following functions:

- Volume backup (including auto-backup)
- Volume migration (including primary space management)
- Recycle
- Backup of migrated data sets
- Backup copy moves from ML1 volumes
- Secondary space management
- Data set migration
- FREEVOL
  - Migration volume
  - Backup volume
  - ML1BACKUPVERSIONS
- SPILL processing
- ARECOVER ML2 tape
- Data set backup

The duplex function is activated using the following commands:

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

The most common reason for not creating copy tapes concurrently is physical tape drive availability. In the case of automatic functions, 2 drives could be required for each task active in the system or complex. For example, if DFSMShsm is running automatic backup on 2 systems, each with 3 backup tasks allowed, then the total number of drives equals:

```
3 backup tasks X 2 tape drives X 2 systems = 12 drives
```

This reflects the maximum number of drives that could be assigned to the backup function if the backup times were overlapping on the two systems.

The DUPLEX command differentiates between backups and migrations. If migration tasks are started, then additional units may also be required. For this reason it may be necessary to stagger backup times on different systems in order to satisfy tape unit requests. Alternatively, you can disable concurrent duplexing and use TAPECOPY at a later time to create the copy tapes.

The current duplex status can be found in the output from the QUERY SETSYS command:

```
ARC0442I   TAPE OUTPUT PROMPT FOR TAPECOPY=NO, DUPLEX
ARC0442I   (CONT.) BACKUP TAPES=YES, DUPLEX MIGRATION TAPES=YES
```

In a library environment, allocation methods need to be considered. In other storage management products, such as CA-Disk, FDR/ABR, and CA-ASM2, device separation is performed either using JCL allocation or dynamic allocation via system parameter. Within an SMS environment, the ACS routines can be used to direct the duplex allocation to a different library (ATL). This can include directing to an offsite location. In a non-SMS environment, the unit specified is used for both primary and duplex (for example, SETSYS UNIT(3590)). This specification requires that allocation separation is performed outside of DFSMShsm.

For example, if your site is using silo devices from StorageTek, then the Host Software Component (HSC) must be configured to perform the device separation. Depending on the release of HSC that you are running, this can be done using definitions in the TAPEREQ member of the HSC PARMLIB. If your release of HSC does not support use of this member, then HSC exits can be used to direct allocations. For more information, please refer to the HSC documentation.

IBM APAR OW45271 describes a new function within DFSMS allowing ACS routines to route mount requests to manual tape libraries. APAR OW45357 adds the support to DFSMShsm to support this function. Using this functionality, customers can realize the benefits of using SMS ACS routines to allocate to tape drives outside of an ATL.

Table 2-1 shows the data set name patterns used by DFSMShsm for the different functions. It can be used to set up filter lists for routing to specific libraries or devices:

Table 2-1 DFSMShsm tape data set name and unit type passed to allocation

DFSMShsm function	Tape data set name	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	

DFSMShsm function	Tape data set name	Commands with unittype restrictions
Migration to original	prefix.HMIGTAPE.DATASET	SETSYS TAPEMIGRATION(- DIRECT(TAPE(unittype)) | 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(unittype))
Spill	prefix.BACKTAPE.DATASET	SETSYS SPILL(TAPE(unittype))
Tapecopy of backup tapes	prefix.COPY.BACKTAPE.DATASET	TAPECOPY ALTERNATEUNITNAME (unittype1, unittype2) TAPECOPY ALTERNATE3590UNITNAME (unittype1, unittype2)
Tapecopy of migration tapes	prefix.COPY.HMIGTAPE.DATASET	
CDS backup Datamover=HSM	uid.BCDS.BACKUP.Vnnnnnn uid.MCDS.BACKUP.Vnnnnnn uid.OCDS.BACKUP.Vnnnnnn uid.JRNL.BACKUP.Vnnnnnn	SETSYS CDSVERSIONBACKUP (BACKUPDEVICECATEGORY - TAPE UNITNAME(unittype)))
CDS backup Datamover=DSS	uid.BCDS.BACKUP.Dnnnnnn uid.MCDS.BACKUP.Dnnnnnn uid.OCDS.BACKUP.Dnnnnnn uid.JRNL.BACKUP.Dnnnnnn	
ABARS processing	outputdatasetprefix.C.CccVnnnn outputdatasetprefix.D.CccVnnnn outputdatasetprefix.I.CccVnnnn outputdatasetprefix.O.CccVnnnn	ABACKUP agname UNIT(unittype)

2.3.6 Tape mount management

Output tape processing performance can be significantly improved by optimizing tape usage using tape mount management. The performance enhancements that can be realized include:

- **Reduction in tape mounts:** New data set allocations can be directed by the ACS routines to a disk storage group and will then migrate to tape when the amount of data merits movement. A similar implementation is achievable using another allocation manager.

- **Reduction of tapes and maximized utilization:** As multiple data sets are being moved to tape by DFSMShsm, they are written to a single tape file. A new tape is requested after the previous has been filled to capacity.
- **Batch run time improvements:** Batch jobs are now writing data sets at disk I/O speeds and do not encounter the wait times associated with tape mounts.

Your installation may already be using a similar philosophy today, using methods provided by your current product. This method can be continued using SMS and DFSMShsm, exploiting interval migration capabilities. For additional information, refer to *DFSMS Using the Volume Mount Analyzer*.

2.3.7 Virtual tape

Another optimization method for the tape environment is the use of virtual tape solutions, such as the IBM 3494 Virtual Tape Server (VTS) subsystem. The benefits of virtual tape allow an implementation without changes to allocation routines or other software changes, as well as reducing device contention against physical tape drives by adding additional virtual devices.

If you are currently using virtual tape technology, it is recommended that you review whether DFSMShsm data should be stored on this medium. The following recommendations apply:

- **Review tape utilization:** Since DFSMShsm re-uses tape it has already written to, this may reduce the need for the stacking capabilities of the virtual tape system.
- **Review number of tape devices:** One of the major benefits of virtual tape is to increase the number of tape units available to the system. More devices will allow you to run a greater number of task concurrently.
- **Review backup requirements:** Virtual tape is a good output media for backup data sets which are less likely to be read after they have been written.

  We recommend setting the DFSMShsm parameter SETSYS PARTIALTAPE(MARKFULL). This will prevent the volume being selected in order to extend a current backup tape, which may initiate a recall of the logical volume in the virtual tape system. Additionally this parameter may reduce the reclaim process used by the virtual tape system, since the physical tape copy of the virtual volume is not invalidated by the extend process.
- **Review migration requirements:** Migration data is less likely to be recalled immediately and will usually trigger a recall of the logical volume in the virtual tape system. This may cause delays in the DFSMShsm recall process, especially since multiple recalls could take place causing additional physical tape mounts in the virtual tape system.

2.4 Preparing the SMS environment

In a previous section we have briefly discussed the overall disk environment, as far as it relates to basic pooling and architecture. At this point we discuss SMS (System Managed Storage), review the benefits of implementation, and elaborate on its interaction with DFSMShsm.

The major benefit of a successful SMS implementation is to eliminate, simplify, and automate tasks that would normally need to be performed by the end-user or storage administrators. The features should be viewed as more than just an allocation product, but also as an enforcement agent and a vehicle to defining the complete life cycle of a data set. SMS manages:

- New data set allocation
- Disk utilization
- Performance
- Availability
- Backup
- Migration
- Data set life cycle
- Data set management
- Storage pools
- Tapes

In general, most sites have some form of SMS activated. This can range from the SMS address space being active with a “null” configuration to the majority of the disk data being under SMS control. A minimal configuration is required in order to use some of the features that DFSMSdfp offers, such as using the *system determined blocksize* for optimal disk allocations. For example, a site may have elected to activate a storage group (volume pool) in order to exploit the benefits of extended format data sets, such as hardware compression or system-managed buffering. Another reason for activation is to allow SMS to direct tape allocations to ATLs.

2.4.1 SMS terminology

Before going into more details about the SMS implementation, we the basic terms needed for a good understanding of the current and following topics.

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 availability, and performance attributes, such as response time and cache

requirements, for data sets. Only system-managed data sets and objects can be assigned a storage class. Storage classes free users from having to know about the physical characteristics of storage devices and manually placing their data on appropriate devices.

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

Storage group A list of storage volumes and attributes that you define. The collection can be a group of:

- System paging volumes
- Disk volumes
- Tape volumes
- Optical volumes
- Combination of disk and optical volumes that look alike
- Disk, tape and optical volumes treated as a single object storage hierarchy

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

SMS classes and groups are used to set service requirements, performance goals, and data definition models for your installation. You may use the ISMF application to create the appropriate classes and groups, and ACS routines to assign them to data according to your installation's policies.

Figure 2-2 shows the order in which the ACS routines are processed.

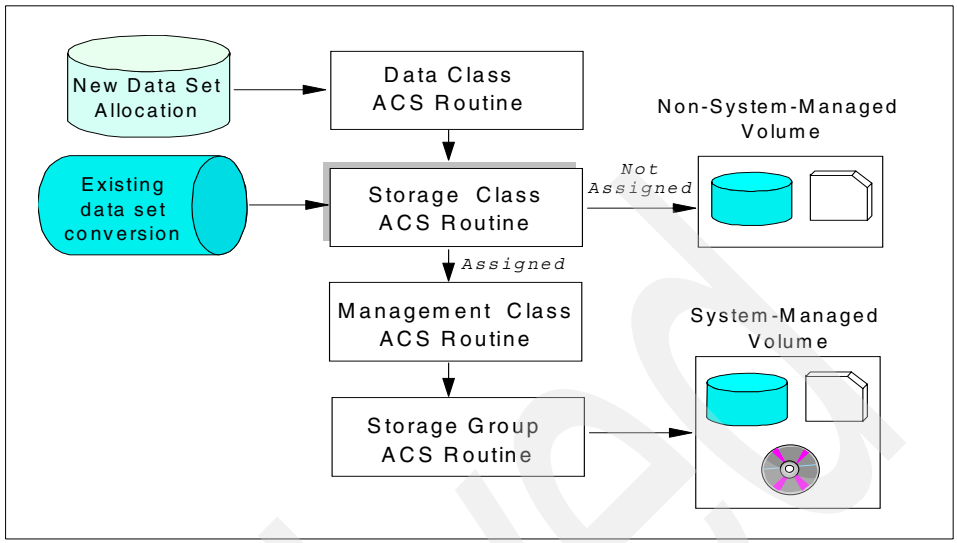


Figure 2-2 ACS routines processing order

2.4.2 NaviQuest

NaviQuest is a data and storage management tool for implementing, testing and verifying the SMS environment. It is installed under the ISMF primary option menu and uses standard ISPF panels.

NaviQuest allows you to do the following:

- Create DFSMS implementation test cases:
  - This can be used during the design and testing phase of your initial ACS routines and SMS configuration.
- Run SMS testing functions in batch.
- Update SMS configuration in batch:
  - This allows you to define or alter constructs attributes in batch, including management class space and backup values.
  - Volumes can also be added and deleted from storage groups.
- Create batch or interactive reports:
  - This can be used during the SMS planning and design phase.
- Perform ongoing storage administration functions:
  - This simplifies testing and verification of SMS configuration changes.

One of the interesting functions is the ability to generate disk reports from DCOLLECT data. For example, a data set report can be customized to report on the fields such as:

- Data set name
- DSORG
- Last reference date
- SMS attributes
- Volume serial
- Percent used
- Change indicator

For more information on NaviQuest, please refer to the publication *DFSMS/MVS NaviQuest User's Guide,* or the redbook *Maintaining Your SMS Environment*, SG24-5484.

2.5 Implementing a basic SMS environment

SMS implementation is a multi-staged project. If you are just starting out, we would like to provide some aspects of SMS implementation to aid in your understanding and deliver improved service to your users. Figure 2-3 shows the recommended stages for a full SMS implementation.

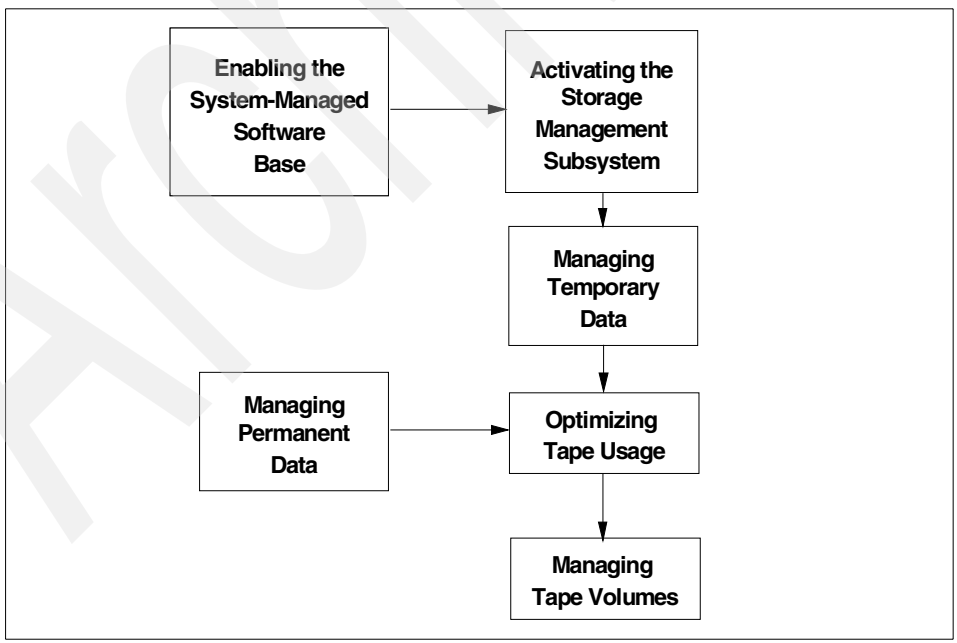


Figure 2-3 SMS implementation flow

You can simply activate SMS by assigning all allocations a null storage class ("**SET STORCLAS=''**"). This activation is sufficient to allow additional implementation. No management class or storage group will be assigned as long as the storage class is null.

In the following sections we review the main steps to get the basic SMS environment ready to work.

Supported data set formats

In the publication *DFSMShsm Implementation and Customization Guide* you can find a list of data set formats that are supported by DFSMShsm. The list differentiates between the dump and space management and backup functions. This allows some data sets which are not supported using the logical processing of DFSMSdss to be accessed using physical dump.

These are some of the data set types which need review:

- Multi-volume data sets
  - Data sets that are multi-volume must be SMS-managed so they can be processed by volume space management or availability management. ABARS is able to process non-SMS-managed multivolume data sets.
  - In order for SMS multi-volume data sets to be processed SETSYS USERDATASETSERIALIZATION must be specified. The differences between the various serialization methods will be discussed later.
- ISAM
  - Not supported in DFSMShsm functions.
- Migrating data sets beginning with SYS1 or HSM.
  - The restriction can be removed using a SETMIG LEVEL ( ) command.
- Data sets cataloged in a VSAM (non-ICF) catalog.

2.5.1 Removing undefined data set organizations

Once you have implemented a null configuration, you are at a point where you can use the SMS data class to remove undefined DSORGs from new allocations. An undefined DSORG is typically created when an allocation of a data set is performed without specifying a DSORG and without OPENing the data set after it is created.

Data sets of this type pose a problem to storage management software because it is not possible to know how to migrate or back up this data set. In some customer environments there are many data sets with an unknown DSORG taking up valuable space on disk for no good reason.

A way to prevent unknown DSORG data sets from being created is by using the data class ACS routine and assigning all new disk allocations a default data class. If no DSORG is specified at allocation or overridden at open, then a DSORG=PS will be assigned. Here are the instructions, assuming that the SMS address space is active and the ISMF application is available:

1. Create a data class.
   a. Start the ISMF application from ISPF/PDF panels using an TSO userid that is authorized as a storage administrator. You get the panel shown in Figure 2-4 (ISMF Primary Option Menu).

```
 Panel  Help
------------------------------------------------------------------------------
                    ISMF PRIMARY OPTION MENU - DFSMS V2R10
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 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 2-4 ISMF Primary Option Menu panel

   b. Select Option 4 (Data Class). You get the Data Class Application Selection panel (Figure 2-5).

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

To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SYS1.SMS.SCDS'
                              (1 to 44 character data set name or 'Active' )
  Data Class Name  . . default   (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


Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 2-5 ISMF Data Class Application Selection panel

c. You have to complete the following information:

- **CDS Name**, which is the name of the SMS Saved Configuration Data Set (SCDS), in quotes.
- **Data Class Name**, the name of the data class to define. In this case, DEFAULT.

d. Select Option 3 (Define), and press Enter key. You get the Data Class Define panel (Figure 2-6). In this set of panels:

- Optionally, type a **Description**.
- Press the Enter key, and then the PF3 key, to save the definition. There are no other parameters necessary for this data class.

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              DATA CLASS DEFINE                 Page 1 of 3
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : DEFAULT

To DEFINE Data Class, Specify:
  Description ==>
              ==>
  Recorg  . . . . . .              (KS, ES, RR, LS or blank)
  Recfm . . . . . .                (any valid RECFM combination or blank)
  Lrecl . . . . . .                (1 to 32761 or blank)
  Keylen  . . . . . .              (0 to 255 or blank)
  Keyoff  . . . . . .              (0 to 32760 or blank)
  Space Avgrec  . . . . .          (U, K, M or blank)
        Avg Value . . . .          (0 to 65535 or blank)
        Primary . . . . .          (0 to 999999 or blank)
        Secondary . . . .          (0 to 999999 or blank)
        Directory . . . .          (0 to 999999 or blank)
```

Figure 2-6 ISMF Data Class Define panel

2. Now we also need a data class ACS routine to assign our new **Default** data class. For doing this, create a member 'DC' in the library where the SMS ACS source is kept. Figure 2-7 shows a sample data class ACS routine, assuming that you have no data class ACS routine yet.

```
PROC DATACLAS
FILTLIST  DISK_UNIT  INCLUDE('DISK',
                             'SYSALLDA',
                             'SYSDA',
                             'WORK',
                             '3390','')
SELECT
  WHEN (&UNIT = &DISK_UNIT)
      SET &DATACLAS = 'DEFAULT'
  OTHERWISE
      SET &DATACLAS = ''
  END /* SELECT */
END  /* END OF DATA CLASS ROUTINE PROC */
```

Figure 2-7 Sample ACS routine to assign data class "DEFAULT"

In this example, DISK_UNIT is a filter list with an exhaustive list of all esoteric and generic names that can be used to specify disk allocation in your environment. SMS must have these explicitly assigned to avoid managing unintended allocations. Included in this list is the value ' ', null string, for those cases where the allocation UNIT parameter is not coded. This routine is intended to assign the **Default** data class to all new disk allocations. After adding this to the ACS source, the storage administrator will need to:

1. Translate the source to the SCDS (ISMF Option 7.2).
2. Validate the SCDS (ISMF Option 7.3).
3. Perform the SET SMS to activate the new configuration. This can be done from either ISMF (Option 8.5) or operator command.

Starting from this data class definition, you may expand the scope of your data classes to include the use of VSAM extended addressability, compression, TMM, or system-managed tape.

2.5.2 Storage class and other considerations

In all SMS implementations that we have reviewed for customers, there is always one class structure that designates the intended storage group for the allocation. This is not an exact 1:1 relationship, but it is close enough. This relationship is intended to survive to the data set expiration so that, at any time in the life of the data set, its intended storage group can be determined. This includes when the data set migrated and/or if it allocates to a *spill* storage group. This is a practical solution for installations that want all the benefits of a storage hierarchy and do not intend to allow wholesale use of *guaranteed space*. We have seen any class used for this purpose, but we advise customers to use the storage class assignment to provide the storage group designation. This provides the necessary piece of information and sacrifices the least SMS functionality.

What do we mean by *sacrificing SMS functionality*? If there is an attribute assigned in via any SMS class construct that goes across all storage groups but applies only to certain data sets, then using this SMS class construct defeats the purpose of storage group identification. A good example would be the data class compression attribute. Compression may apply to PS (Physical Sequential) data sets as well as VSAM Extended Function data sets. Compacted data is not typically a unique aspect of a storage group, rather its is an attribute of a data set (like logical record length). These data sets could reside in any SMS storage group with this attribute, and therefore, it makes no logical sense to use data class as our storage group marker. This is not to say that we have never seen all compressed data in a customers environment reside in a single SMS storage group. In the cases where we have seen this done, it was in one of a few storage groups and it was almost the sole purpose of their SMS implementation.

The only SMS attributes in a storage class that would possibly go across more than one storage group would be *data striping* and *guaranteed space*. These are typically special case data sets known to the installation, or can be considered attributes of all data sets allocated in a specific storage group. If either of these cases are true, then no sacrifice of SMS functionality needs to be made.

2.5.3 SMS management class

In this section we review the management class and storage group constructs and examine how DFSMShsm uses them to perform storage management functions. Using sample constructs, we explain the fields and how the function affects DFSMShsm processing.

Attention: Most of the attributes in the management class have a default value. If a value is not specified the default will be used, which means DFSMShsm may process the data differently than expected.

The management class allows you to group data sets logically and specify how they will be managed by DFSMShsm. It will allow you to set up the policies for availability management (backups) and space management (migration and expiration).

Figure 2-8 shows the use of management class by DFSMShsm and DFSMSdss for data management requirements.

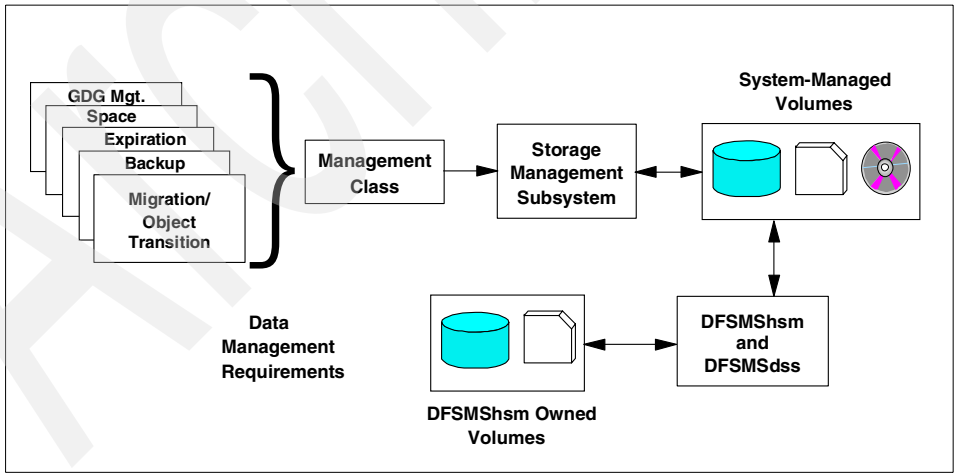


Figure 2-8 Management class use with DFSMShsm and DFSMSdss

Default management class

We strongly recommend that you define a default management class in the SMS base configuration.

If a data set is system-managed (under SMS control), that is, it has a storage class assigned, then DFSMShsm will check for a management class. If the data set does not have an associated management class, then a check is made for a default management class. If one exists, the data set is processed using these attributes. If a default management is not present, then DFSMShsm generates its own default attributes.

Figure 2-9 shows the **Default Management Class** setting, using ISMF option 8.1, as defined in the SMS base configuration:

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                               CDS BASE DISPLAY
Command ===>

CDS Name  . : ACTIVE
CDS Status  : VALID

Description : BASE SMS CONFIG FOR MAIN SYSPLEX
==> . . . . :
Default Management Class : MC365       Default Device Geometry :
Default Unit . . . . . . : 3390          Bytes/track . . . . . : 56664
                                         Tracks/cylinder . . . : 15

System: SC63     SC64     SC65



Sysgrp:
```

Figure 2-9 Default management class specification

Management class parameters

Table 2-2 shows the management class parameters relevant to DFSMShsm. The default values are also supplied.

Table 2-2 Management class parameters with default values

Management Class Attribute	Default Value	Required parameter
Expire after days non-usage	NOLIMIT	Yes
Expire after Date/days	NOLIMIT	Yes
Retention limit	NOLIMIT	Yes
Partial Release	No	Yes
Primary days non-usage	2 days	Yes, if Command not NONE
Level-1 days non-usage	60 days	Yes, if Command not NONE
Command or Auto-migrate	No default, must be either BOTH, COMMAND or NONE	Yes
# GDG elements on primary	<no default>	No
Rolled-off GDS action	<no default>	No
Backup frequency	1 day	Yes, if Admin or user command not set to None, or if Auto backup set to Y
Number of backup versions (Data set exists)	2	Yes, if Admin or user command not set to None, or if Auto backup set to Y
Number of backup versions (Data set deleted)	1	Yes, if Admin or user command not set to None, or if Auto backup set to Y
Retain days only backup version (Data set deleted)	60	Yes, if Admin or user command not set to None, or if Auto backup set to Y
Retain days extra backup versions	30	Yes, if Admin or user command not set to None, or if Auto backup set to Y
Admin or user command backup	BOTH	Yes
Auto Backup	Y	Yes
Backup copy technique	No default, must be either (P)referred, (R)equired or (S)tandard	Yes
Aggregate backup # versions	<no default>	No
Retain only version	<no default>	No
Unit	No default, but cannot be blank if Retain only version is specified or Retain only version is NOLIMIT	No
Retain extra versions	<no default>	No

Management Class Attribute	Default Value	Required parameter
Unit	No default, but cannot be blank if Retain extra versions is specified or Retain extra versions is NOLIMIT	No
Copy serialization	No default. If blank then no aggregate processing is performed	No
Abackup copy technique	No default, must be either (P)referred, (R)equired or (S)tandard	Yes

Management class definition example

In this section we review the different attributes that we can specify in a management class.

Expiration attributes

In Figure 2-10, we are looking at the expiration attributes for a data set associated with management class MC365. The data set does not have an expiration date specified in the disk VTOC.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                          MANAGEMENT CLASS DISPLAY                 Page 1 of 5
Command ===>

CDS Name . . . . . .  : ACTIVE
Management Class Name . : MC365

Description  :



Expiration Attributes

  Expire after Days Non-usage  . : 365
  Expire after Date/Days . . . . : NOLIMIT
  Retention Limit  . . . . . . . : NOLIMIT
```

Figure 2-10 Management class expiration attributes

The following statements apply in this case:

- Since the data set has no VTOC expiration date, the expiration attributes in the management class will be honored.
- The data set will be expired after it has remained unreferenced for 365 days, as specified in **Expire after Days Non-usage**. If the data set is referenced during this time, the reference date will be updated to reflect the usage date and the 365 days period will start again.
- Since **Expire after Date/Days** is set to NOLIMIT, this attribute is not used to expire the data set.
- **Retention Limit** is set to NOLIMIT, which means that if a data set is assigned with a expiration date or retention period, either via JCL or SMS data class, this value would not be overridden. If SETSYS EXPIREDDATASETS(SCRATCH) is specified, the data set is scratched regardless of the management class settings.

Space management attributes

Figure 2-11 shows the attributes related to space release and migration, and also GDG management.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                          MANAGEMENT CLASS DISPLAY               Page 2 of 5
Command ===>

CDS Name  . . . . . . . . . . : ACTIVE
Management Class Name . . . : MC365

Partial Release . . . . . . : YES


Migration Attributes
  Primary Days Non-usage  . : 10
  Level 1 Days Date/Days  . : 30
  Command or Auto Migrate . : BOTH


GDG Management Attributes
  # GDG Elements on Primary :  2
  Rolled-off GDS Action . . :  EXPIRE
```

Figure 2-11 Management class space management attributes

Next we review the meaning of each of the attributes specified:

- **Partial Release . . . YES,** means to release unused space in the data set during space management, assuming that the data set is either physical sequential, partitioned, or extended format VSAM. Other values that could be specified are:
  - **YI - Yes Immediate**. Release unused space in the data set during space management and also on closing the data set if open for output.
  - **C - Conditional**. For physical sequential and partitioned data sets, release unused space during space management if nonzero secondary allocation is specified. For extended format VSAM, the data set is processed as if the value were Yes.
  - **CI - Conditional Immediate**. For physical sequential and partitioned data sets, release unused space during space management if nonzero secondary allocation is specified and also on closing the data set if open for output. For extended format VSAM, the data set is processed as if the value were Yes.
  - **No -** Unused space release is not performed against this data set.
- The data set is eligible for migration from primary disk to Migration Level 1 (ML1) after 10 days, as specified in **Primary Days Non-usage**.

  Note: Even though the data set has been on disk for the number of days specified in **Primary Days Non-usage**, it will not necessarily be migrated during the next space management cycle. DFSMShsm will only migrate data sets in order to achieve the free space specified in the storage group. This is different from most other storage management products, in that they will migrate a data set when this value is reached, regardless of the volume occupancy.

- The data set is eligible for migration from ML1 to ML2 after 30 days. **Level 1 Days Date/Days** attribute serves multiple purposes:
  - This value is the *total* number of days (on primary disk and ML1) that the data set must be unused before being eligible to migrate from level 1 to level 2. It does not refer to the number of days that a data set has resided on ML1. All eligible data sets will move to level 2 during migration processing.
  - It dictates the eligibility of a data set to migrate to level 1.
    - A value of 0 indicates that the data set will migrate straight to level 2.
    - A value of 1 - 9999 means that the data set may migrate from primary disk to migration level 1 and subsequently to level 2.

- NOLIMIT means that the data set is eligible for migration to level 1, but not to level 2.

– If the data set is still on primary disk and the value of **Level 1 Days Date/Days** is met, the data set will migrate directly to level 2.

- The data set can be migrated using either a migrate command or automatically during primary space management, as stated in **Command or Auto Migrate..: BOTH**. Specifying COMMAND allows the data set to migrate using command only. NONE disables migration for the data set.

  Command level migration does not use **Primary Days Non-Usage** to determine if the data set is eligible for migration. The data set will be migrated if either BOTH or COMMAND are specified.

- If the data set is a Generation Data Set (GDS), the two newest versions are processed as defined in **Primary Days Non-Usage**.

  – Older entries in the Generation Data Group (GDG) are eligible for early migration. They are selected ahead of other (non-GDS) data sets on the volume and the value in **Primary Days Non-Usage** is not honored in these cases.

  – The **# GDG Elements on Primary** parameter does not guarantee that two versions of the GDG will remain on primary disk. Normal aging rules will apply as defined in **Primary Days Non-Usage**.

- Generation data sets which are rolled off, meaning they are no longer in the GDG sphere, will be scratched.

  – This applies to GDSs which are part of a GDG which has the NOSCRATCH parameter specified. GDSs with the SCRATCH parameter will be scratched when they are rolled off.

  – If **Rolled-off GDS Action** is set to MIGRATE, then the GDS is assigned the highest migration priority.

Backup attributes

Figure 2-12 shows the backup attributes specified for sample management class MC365.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                          MANAGEMENT CLASS DISPLAY               Page 3 of 5
Command ===>

CDS Name  . . . . . . : ACTIVE
Management Class Name : MC365

Backup Attributes
  Backup frequency  . . . . . . . . . . . . : 1
  Number of backup versions . . . . . . . . : 10
     (Data Set Exists)
  Number of backup versions . . . . . . . . : 2
     (Data Set Deleted)
  Retain days only backup version . . . . : 60
     (Data Set Deleted)
  Retain days extra backup versions . . . : 20

  Admin or User Command Backup  . . . . . : BOTH
  Auto Backup . . . . . . . . . . . . . . . : YES
  Backup copy technique . . . . . . . . . . : STANDARD
```

Figure 2-12 Management class backup attributes

The following applies to the data set managed by this management class:

- One day must elapse before the data set is backed up, if it has been changed.

  Specify 0 in **Backup frequency** if you want the data set backed up every time it changes.

- Ten backup versions of the data set will be kept in the Backup Control Data Set (BCDS), if the data set is still cataloged, as specified in **Number of backup versions**. When this value is exceeded, the oldest version is scratched immediately.

  Note: This is an important difference to other storage management products. Products, such as CA-Disk, will delete the additional versions during its files data set processing (IXMAINT). This is a batch process and could run some time after the new version was added. It is therefore possible to have more copies than the management class allows.

  The maximum number of backup versions that DFSMShsm can store in the BCDS is 100. This number depends on the record size of the BCDS. If this is less than 6544, then the maximum number of versions is 29.

- After the data set is deleted or expired from primary disk or ML1, then two versions will be kept for additional 60 days (**Retain days only backup version**).
  - If more than two versions exist at the time the data set is deleted, the additional versions are discarded when EXPIREBV process is run.
  - The versions already in the BCDS will be used after the data set has been deleted. The maximum number of versions allowed will now check the **Number of backup versions (Data set deleted)** instead of **Number of backup versions (Data set exists)**.
- Extra backup versions of the data set are kept for 20 days (**Retain days extra backup versions**). This value applies whether the data set still exists or has been deleted and also that the maximum number of versions has not been exceeded.

Note: Please note that expired backup versions of data sets are deleted during EXPIREBV processing. In fact, EXPIREBV needs to initially run twice in order to discard a version of a data set that has been deleted. The first run will detect that the data set is deleted and will note the date of the EXPIREBV run in the BCDS. A subsequent run will delete the data set based on **Retain days extra backup versions**. EXPIREBV will be discussed in detail later in this book.

- Both the user and the storage administrator can issue backup commands against this data set (**Admin or User Command Backup**). A value of ADMIN limits command backups to the storage administrator.
- Automatic backups can be taken of the data set (**Auto Backup .....: YES**).

Note: If **Admin or User Command Backup** has a value of NONE, then no automatic backups are taken, regardless of the value in **Auto backup**.

- The data set is backed up using standard access without an attempt to use the concurrent copy technique, as specified in **Backup copy technique . . . STANDARD**. Other values for this field are:
  - **R - Concurrent required**. Backups will fail for data sets that reside on devices not supported by concurrent copy or if concurrent copy is not available.
  - **P - Concurrent preferred**. Backups are taken for data sets using concurrent copy if available, otherwise the standard technique is used.

ABARS attributes

In order to use Aggregate Backup and Recovery Support (ABARS), **AGGREGATE Backup Attributes** are specified in the management class. These can be supplied in an existing or separate management class.

Figure 2-13 shows some sample management class definitions for ABARS attributes.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                          MANAGEMENT CLASS DEFINE            Page 5 of 5
Command ===>

SCDS Name . . . . . . : 'ACTIVE'
Management Class Name : MCAB365

To DEFINE Management Class, Specify:
  AGGREGATE Backup Attributes:
    # Versions . . . . . . . . . 10       (1 to 9999, NOLIMIT or blank)
    Retain Only Version  . . . 12       (1 to 9999, NOLIMIT or blank)
      Unit . . . . . . . . . . M        (D=days, W=weeks, M=months, Y=years or
                                        blank)
    Retain Extra Versions  . . 4        (1 to 9999, NOLIMIT or blank)
      Unit . . . . . . . . . . W        (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 2-13 Management class aggregate backup attributes

Our sample management class is called MCAB366 and it is different from the one used for explaining the expiration and migration attributes. ABARS use a *selection data set* and an optional *instruction data set.*

- The **selection data set** contains the following information:
  - An INCLUDE list of data sets to be processed.
  - An optional EXCLUDE list of data sets.
  - An optional ACCOMPANY list of tape data sets that should be taken with the aggregate file and cataloged at the recovery site.
  - An optional ACCOMPANYEXCLUDE list of data sets that should not be processed.

  - An optional ALLOCATE list of data sets that should be allocated and cataloged at the recovery site.
  - An optional ALLOCATEEXCLUDE list showing data sets in the ALLOCATE list that should not be processed.
- The **instruction data set**, if created, contains helpful information or instructions that can be used at the recovery site.

Using Figure 2-13 on page 43, the aggregate is processed with the following attributes:

- Ten versions of the aggregate will be kept, as specified in **# Versions . . . 10**.
  - Up to 9999 versions can be kept and automatically rolled off.
  - NOLIMIT means that no automatic roll-off occurs.
  - Blank signifies that no versions are kept.
- The only version is kept for twelve months, as specified in **Retain Only Version . . . 12**. In this case the UNIT field is used to determine retention units.
  - The numeric value allows for 1 to 9999, NOLIMIT or blank
  - The UNIT field can have values (D)ays, (W)eeks, (M)onths, (Y)ears, or blank.
- In the example, extra backup versions of the aggregate group are kept for four weeks, as specified in **Retain Extra Versions . . . 4, Unit . . . H.**
  - Possible values are the same as for only backup version retention.
- The backup process continues even if a shared enqueue cannot be obtained for data sets in the aggregate, as specified in **Copy Serialization . . . C**. Other values are:
  - **F - Fail**. It ensures that the aggregate is not backed up unless integrity is guaranteed.
  - **blank** - Which means that no aggregate backup is performed.
- The aggregate backup is taken using standard copy technique (**Abackup Copy Technique . . . S**). Other values are:
  - **P - Preferred**. It means that the backup takes place even if the data sets reside on a volume which does not support concurrent copy, or where concurrent copy is not available.
  - **R - Required**. It means that the backup will fail if concurrent copy cannot be used to back up the data set.

2.5.4 SMS storage group

The storage group SMS construct is a definition containing a group of disk volumes or tape drives, and allows you to define how these volumes should be managed by DFSMShsm. It is used to specify:

- Automatic dumps. Should they be performed, and to what dump classes?
- Automatic backups. Are the volumes part of the backup window?
- Space management. What actions are performed and at what occupancy level?

Since the user cannot specify a storage group at data set allocation, the ACS routines are set up to direct the allocations to a storage group based on business and performance requirements. The interaction between the storage group and management class allow data sets with different life cycles to reside on volume pools with similar performance needs. This allows storage management on a data set level, but still allows management on a volume or pool level as required.

In this section we discuss the disk storage group parameters and emphasize any influence on the management class. The following applies to disk storage groups:

- These groups contain volume names (volsers) rather than device addresses.
- Each volume must have a VSAM volume data set (VVDS) and an indexed volume table of contents (VTOC).
- A volume can only be defined in one storage group; two storage groups cannot share a volume.
- A data set can only be allocated in one storage group. It can span volumes in the same storage group but cannot span volumes on different storage groups.
- All components of a VSAM data set (including indices) must reside in the same storage group.
- We suggest that all volumes in the storage group have the same device geometry, that is 3390 disk volumes together with other 3390 disk volumes. This is not a requirement but is recommended.

In order to allow SMS data to reside on a disk volume, it must be prepared for SMS management. There are two ways of enabling data for being under SMS control:

- Moving data to the volume from a non-SMS managed volume, using a data mover such as DFSMSdss.
- Converting existing data on the volume to SMS control without moving data, using the CONVERTV function in DFSMSdss.

The method that you decide to use will depend on your current pooling implementation. If you are confident that the pools and storage groups still reflect the original design intentions, then converting without data movement is most likely the best method. If you think that the environment needs to be re-examined or if improvements can be made, then moving the data from the original volumes to new storage groups is the best solution. The ACS routines are invoked to direct allocation to the new volumes.

DFSMSdss CONVERTV

If you decide to use the CONVERTV function, we recommend that you use the TEST keyword in order to ensure that SMS assignments will end up as expected. The resulting report shows the classes associated with the data sets on a volume.

Example 2-1 shows a sample JCL of a DFSMSdss job using the CONVERTV parameter with the TEST keyword:

Example 2-1 Sample DFSMSdss job using CONVERTV

```
//STEP1 EXEC  PGM=ADRDSSU,REGION=2048K
//*
//SYSPRINT  DD  SYSOUT=*
//INVOL1   DD    VOL=SER=HG6611,UNIT=3390,DISP=SHR
//SYSIN    DD    *
 CONVERTV -
      PREPARE -
      TEST -
      DDNAME(INVOL1)
/*
```

If the volume fails to convert successfully, it will be assigned a status of INITIAL. This means that the convert process has detected data sets which failed to convert. In this state the volume will not accept new allocations nor can it be used for data sets extending to this volume.

A successful conversion will show the volume in CONVERT status. This means that the volume is fully available for SMS control, and that all data sets are cataloged in an ICF catalog, have a storage class associated with them, and the volume has an indexed VTOC. At this stage the volume has also been assigned to a storage group.

Figure 2-14 shows an ISMF volume list. You can see the physical status column for each volume.

```
  Panel  List  Utilities  Scroll  Help
------------------------------------------------------------------------------
                                VOLUME LIST
Command ===>                                                Scroll ===> HALF
                                                        Entries 1-14 of 16
Enter Line Operators below:                             Data Columns 19-24 of 40

     LINE       VOLUME DUPLEX    OTHER   SUBSYS  PHYSICAL  STORAGE   CF VOLUME
     OPERATOR   SERIAL STATUS    DEVICE  ID      STATUS    GRP NAME  STATUS
    ---(1)----  -(2)-- --(19)--  -(20)-  -(21)-  --(22)--  --(23)--  --(24)---
                HG6600 SIMPLEX     ----    8906  CONVERT   SGLSS67   ENABLED
                HG6601 SIMPLEX     ----    8906  CONVERT   SGLSS67   ENABLED
                HG6602 SIMPLEX     ----    8906  CONVERT   SGLSS67   ENABLED
                HG6603 SIMPLEX     ----    8906  INITIAL   SGLSS67   ENABLED
                HG6604 SIMPLEX     ----    8906  CONVERT   SGLSS67   ENABLED
                HG6605 SIMPLEX     ----    8906  CONVERT   SGLSS67   ENABLED
                HG6606 SIMPLEX     ----    8906  INITIAL   SGLSS67   ENABLED
                HG6607 SIMPLEX     ----    8906  CONVERT   SGLSS67   ENABLED
                HG6700 SIMPLEX     ----    8907  CONVERT   SGLSS67   ENABLED
                HG6701 SIMPLEX     ----    8907  CONVERT   SGLSS67   ENABLED
                HG6702 SIMPLEX     ----    8907  CONVERT   SGLSS67   ENABLED
                HG6703 SIMPLEX     ----    8907  CONVERT   SGLSS67   ENABLED
                HG6704 SIMPLEX     ----    8907  CONVERT   SGLSS67   ENABLED
                HG6705 SIMPLEX     ----    8907  CONVERT   SGLSS67   ENABLED
```

Figure 2-14 ISMF SMS volume list showing physical status

For more detailed information on bringing volumes under system-managed control, please refer to the manuals *DFSMSdfp Storage Administration Reference* and *DFSMS Implementing System Managed Storage*.

Storage group processing in DFSMShsm

DFSMShsm retrieves the storage group information before the following automatic functions begin:

- Automatic volume space management, including primary space management and interval migration
- Automatic secondary space management
- Volume space management initiated via command

The following must be true in order for DFSMShsm to consider the storage group for processing:

- Storage group type must be of type POOL.
- The SMS status must be either ENABLED, QUIESCED NEW, or DISABLED NEW.
- The automatic migration parameter must be set to Y, I or P.

Table 2-3 shows the storage group parameters relevant to DFSMShsm. The default values are also supplied.

Table 2-3 Storage group parameters showing default values

Storage Group Attribute	Default Value	Required?
Auto migrate	Y (supplied by ISMF)	Yes
Migrate Sys/Sys group name	<no default>	No
Auto backup	Y (supplied by ISMF)	Yes
Backup Sys/Sys group name	<no default>	No
Auto dump	N (supplied by ISMF)	No
Dump Sys/Sys group name	<no default>	No
Dump class	<no default>	No
Allocation/migration high threshold	85 (supplied by ISMF	Yes
Allocation/migration low threshold	<no default>	Yes, if AM=Y, I, P
Guaranteed backup frequency	<no default>	Yes, if auto backup=Y
Storage group status	N	Yes
Storage group status define	ENABLE	Yes

Storage group definition example

Figure 2-15 shows an example of the panel you are presented when defining a new storage group SGRES3T.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                          POOL STORAGE GROUP DEFINE
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : SGRES3T
To DEFINE Storage Group, Specify:
 Description ==>  Sample Storage Group
             ==>
 Auto Migrate . . Y  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . Y  (Y or N)         Backup Sys/Sys Group Name  . .  MVSB
 Auto Dump  . . . Y  (Y or N)         Dump Sys/Sys Group Name  . . .

 Dump Class . . .  WEEKDMP            (1 to 8 characters)
 Dump Class . . .                     Dump Class . . .
 Dump Class . . .                     Dump Class . . .

 Allocation/migration Threshold: High . . 90  (1-99)      Low  . . 60  (0-99)
 Guaranteed Backup Frequency  . . . . . . 7         (1 to 9999 or NOLIMIT)

 DEFINE    SMS Storage Group Status . . . N   (Y or N)
Use ENTER to Perform Verification and Selection;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 2-15 Pool Storage Group Define panel

These are the fields to note:

- As this is a new storage group, the **SCDS Name** cannot be ACTIVE. Any changes to the current environment need to be stored into an SCDS and then copied into the ACDS. This action is referred to as activating the CDS.

  In addition, SMS requires a Communication Data Set (COMMDS), which establishes communication between SMS address spaces in a multisystem environment. It also contains space statistics, SMS status and MVS status for each system-managed volume.

- **Auto Migrate** specifies whether data sets are migrated automatically and during which stage of DFSMShsm processing this is to occur. The possible values are:

  Y Data sets are processed during primary space management, if eligible. When SETSYS INTERVALMIGRATION is specified, then interval migration is also performed.

  N Data sets in this storage group are not eligible for automatic migration, even if SETSYS INTERVALMIGRATION is specified.

I Data sets are eligible for automatic migration either during primary space management or interval migration.

P Data sets are eligible for automatic migration during primary space management only.

- Data sets in this storage group are eligible for processing using automatic incremental backup, as specified in **Auto Backup . . . Y**. Since **Backup Sys/Sys Group Name** is set to MVSB, the data sets are only processed by automatic backup from system MVSB. The following must be true in order for the data set to be processed:
  - Management class attribute **Admin Or User Command Backup** set to ADMIN or BOTH.
  - Management class attribute **Auto Backup** set to Y
  - Data set-changed-indicator is set as follows:
    - If changed-indicator is ON, and last backup date can be determined, DFSMShsm checks management class attribute **Backup Frequency** to see the if the number of days has been met or exceeded.
    - If changed-indicator is ON, and last backup date cannot be determined, DFSMShsm backs up the data set, since this is considered a new data set.
    - If changed-indicator is OFF, then **Guaranteed Backup Frequency** attribute is checked. If last backup date equals or is less than this value, then the data set is backed up.
- Volumes in this storage group will be dumped automatically (**Auto Dump . . . Y**) using dump class WEEKDMP, as specified in **Dump Class**.
  - Dump classes are defined in ARCCMD00
  - ISMF does not check the validity of this dump class
- The high allocation threshold for the storage group is set at 90 percent and the low threshold is 60 percent (**Allocation/migration Threshold**). These values will be discussed later in the chapter.
- Based on the value of **Guaranteed Backup Frequency**, if the last backup of a data set was taken seven or more days ago, another backup will be taken and replace the previous backup.
  - This action does not increment the number of backup versions of the data set.
  - This reduces the need to run RECYCLE in order to regain tapes with small numbers of unexpired versions.

- SMS group status will remain as ENABLE. If the storage group needs to have a different status on any system, then **SMS Storage Group Status** field value should be set to Y and the status changed in the ISMF SMS Storage Group Status Define panel (Figure 2-16).

```
 Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                      SMS STORAGE GROUP STATUS DEFINE     Cursor not on choice
Command ===>

SCDS Name  . . . . : SYS1.SMS.SCDS
Storage Group Name : SGRES3T
Storage Group Type : POOL
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
 SC63       ===> ENABLE    SC64       ===> ENABLE      - Tape SG Type
 SC65       ===> DISNEW               ===>                NOTCON, ENABLE,
            ===>                      ===>                DISALL, DISNEW )
            ===>                      ===>
            ===>                      ===>             * SYS GROUP = sysplex
            ===>                      ===>               minus Systems in the
            ===>                      ===>               Sysplex explicitly
            ===>                      ===>               defined in the SCDS
```

Figure 2-16 SMS Storage Group Status Define panel

2.5.5 Automatic migration

DFSMShsm performs automatic migration during primary space management and interval migration. The latter check depends on settings in the SMS storage group, or on the SETSYS INTERVALMIGRATION or SETSYS NOINTERVALMIGRATION commands.

Interval migration

If SETSYS INTERVALMIGRATION is specified or a storage group has value **Auto Migration=I (AM=I)**, data sets are eligible for migration on an hourly basis. For storage groups with AM=I, the low and high threshold values are queried, and migration is attempted if the threshold is more than or equals the halfway mark between the two values. For example,

- For storage group SGRES3T, the high threshold is defined as 90 percent, the low as 60 percent.

- Storage group SGRES3T contains three disk volumes. SGRES2 is at 73 percent capacity, SGRES1 is at 76 percent and SGRES3 is at 77 percent capacity.
- Interval migration will check the volumes in storage group SGRES3T and migrate eligible data sets from SGRES1 and SGRES3 until the low threshold (60 percent) is reached.

Note: With AM=I, interval migration is performed on data sets in the storage group, even if SETSYS NOINTERVALMIGRATION is used.

Interval migration is often specified for storage groups defined for tape mount management (TMM) since these groups tend to fill up more during the working day. Interval migration allows DFSMShsm to ensure that disk space availability is maintained within the pool.

With SETSYS INTERVALMIGRATION, non-SMS disk volume thresholds are also queried and migration performed if needed.

Migration during primary space management

When the storage group has parameter AM=Y, P or I specified, data sets are processed for auto migration during primary space management. This will occur during the second phase (pass 2), after pass 1 and extent reduction. DFSMShsm will then migrate data sets until the low threshold value is reached; the candidates for migration were selected during Pass 1.

Note: The high threshold value in the storage group is not used during primary space management. It is used to establish data set eligibility for interval migration. SMS uses the high threshold to balance space allocations.

Guaranteed backup frequency

As seen in Figure 2-15 on page 49, the guaranteed backup frequency allows an additional backup of a data set to be made, whether the changed indicator is set or not. When using this attribute you should note the following:

- The guaranteed backup is made during automatic backup or command level incremental volume backup.
- The guaranteed backup will replace the previous backup version.
- Backups taken using (H)BACKDS or BACKVOL TOTAL commands do not interact with guaranteed backup frequency and the data set backup versions are kept in accordance with normal management class rules.

- Guaranteed backup frequency takes precedence over backup frequency in the management class and represents the maximum amount of time between backups. Backup frequency represents the minimum time between backups.
- Guaranteed backup frequency offers a method of reducing the total number of tape mounts needed to recover a disk volume when few of the data sets do not change and recovery is performed from incremental backups.
- **Auto Backup** must be set to Y for guaranteed backup frequency to work. If a BACKVOL INCREMENTAL command is used, then the attribute is honored.

2.5.6 Non-system-managed volumes

The major difference between the management of data sets on system-managed volumes and those residing on non-system-managed volumes is the level at which the functions are provided.

In a system-managed environment the management class is queried after the storage group in order to provide functionality at the data set level. The policy in place for a non-system-managed data set are set at the volume level, and the rules are defined by ADDVOL parameters in the ARCCMD00 member.

Availability management

Some of the backup implications of the volume-based operations are:

- If the ADDVOL command for a volume allows automatic backup, then all modified data sets on the volume are backed up.
- All data sets on a DFSMShsm host will have the same backup frequency.

  ALTERDS allows the backup frequency to be changed by supplying fully-qualified data set name.
- All data sets on a DFSMShsm host will have the same number of backup versions.

  ALTERDS allows the number of retained backup versions to be changed by supplying a fully-qualified data set name.

Space management

Space management considerations for non-system-managed volumes include:

- Migration age and management technique apply to all data sets on a volume.
- Deletion age and technique apply to all data sets on a volume that is managed by the deletion (delete by age) technique.
- Deletion age and technique apply to all data sets on a volume that is managed by the retirement (delete if backed up) technique.

- Data sets on a ML1 volume are migrated to ML2 volumes after a uniform time specified in the HSMplex.

Volume dumps

DFSMShsm commands are required to perform dumps against non-system-managed volumes, although the function performed is the same. In addition ML1 volumes are automatically dumped by using non-SMS-managed volume commands, since these are not system-managed.

Using the ADDVOL command

A non-system-managed volume is defined for DFSMShsm management using the ADDVOL command. An ADDVOL command must be issued for each volume that is managed or owned by DFSMShsm, including volumes used for ABARS processing. Each primary and migration level 1 volume is defined in the ARCCMD00 member and is processed at DFSMShsm startup time. The command is also used to do the following:

- Add a non-system-managed volume which is being managed for the first time to DFSMShsm control.
- Add a primary or migration level 1 volume that was used previously, but was not included in the ARCCMD00 member.
- Change the attributes of a previously defined volume.
- Change attributes so that the volume is not selected for migration output.
- Add all ABARS volumes used during ARECOVER processing, if these were not previously added.

The following restrictions apply:

- The ADDVOL command is rejected for:
  - A system-managed volume
  - A volume whose management cannot be determined, or
  - A volume that is not online
- In a JES3 environment, a primary volume cannot be added after DFSMShsm initialization.

2.5.7 Documentation

There are many IBM books available to address many aspects of System Managed Storage. Table 2-4 shows a sample list of what is provided with the DFSMSdfp product.

Table 2-4 DFSMS product publications for SMS implementation

Publication Title	Publication Number
z/OS V1R1.0 DFSMS Introduction	SC26-7397-00
z/OS V1R1.0 DFSMS: Implementing System-Managed Storage	SC26-7407-00
z/OS V1R1.0 DFSMS: Using the Interactive Storage Management Facility	SC26-7411-00
z/OS V1R1.0 DFSMS: Using Data Sets	SC26-7410-00
z/OS V1R1.0 DFSMSdfp Storage Administration Reference	SC26-7402-00
z/OS V1R1.0 OAM PISA for Object Support	SC35-0426-00
z/OS V1R1.0 OAM PISA for Tape Libraries	SC35-0427-00

In addition, in Table 2-5 you can find a topical sampling of the IBM International Technical Support Organization publications related to SMS implementation or exploitation.

Table 2-5 ITSO DFSMS redbooks

Publication Title	Publication Number
Get DFSMS FIT: Fast Implementation Techniques	SG24-2568-00
DFSMS/MVS Version 1 Release 2.0 Data Compression Implementation Guide	GG24-4251-00
Maintaining Your SMS Environment	SG24-5484-00
Storage Management with DB2 for OS/390	SG24-5462-00
Hierarchical File System Usage Guide	SG24-5482-01
CICS and VSAM Record Level Sharing: Implementation Guide	SG24-4766-00

2.6 DFSMShsm in a multiple-image environment

There are a number of considerations to be made before activating DFSMShsm in a multiple-image environment. This will depend on the implementation of your current storage management product and if it is appropriate or necessary to this processing into the DFSMShsm implementation.

Newer releases of DFSMShsm allow multiple hosts to perform space management and backup functions. These hosts can be located:

- As multiple z/OS or OS/390 images
- As multiple address spaces on a single z/OS or OS/390 system, or
- As a combination of the above

When multiple hosts use the same set of DFSMShsm control data sets, this is referred to as an *HSMplex*.

With this information, you can decide whether one or multiple HSMplexes will be required. For example, your current implementation may have the other product active on all systems using one control data set, but storage management functions are only performed on one of these systems. This configuration would work with one set of control data sets. If there is a need to run with separate production and test environments then two HSMplexes will need to be set up.

2.6.1 Resource serialization

Regardless of whether DFSMShsm runs in a single or multi-image environment, integrity needs to be assured. This is achieved using three possible methods:

- RESERVE. This is used to request exclusive use of a volume, preventing other images from updating the volume.
- ENQUEUE. Allows access to a resource and provides serialization for resources from the same address space on the same z/OS or OS/390 image. If multiple z/OS or OS/390 images share the same resource then serialization must be achieved using Global Resource Serialization (GRS) or a similar product, such as CA-Multi-image Manager (CA-MiM).
- VSAM record level sharing (RLS) manages the serialization of the VSAM data sets. This allows DFSMShsm to exploit the coupling facility.

2.6.2 Global resource serialization

In order to ensure integrity it is important to understand your current environment and the resources that are currently managed. The scopes that are used in a GRS environment are as follows:

STEP	Scope within an address space
SYSTEM	Scope within a z/OS or OS/390 image
SYSTEMS	Scope across multiple z/OS or OS/390 images.

GRS resource name lists (RNLs) are:

- **SYSTEM inclusion RNL**

  Resources specified with a scope of SYSTEM that GRS should propagate as global resources.

- **SYSTEMS exclusion RNL**

  Resources specified with a scope of SYSTEMS that GRS should treat as a local resource.

- **RESERVE conversion RNL**

  Resources specified with RESERVE which GRS should suppress the hardware reserve.

2.6.3 User data set serialization

DFSMShsm processes user data sets using a SYSDSN as the resource for non-VSAM data sets and SYSVSAM for VSAM data sets. The level of serialization required, for example shared or exclusive, depends on the function that is performed. In a multiple host environment this protection must be propagated to other processors using GRS or equivalent product.

User data sets are processed either using volume reserves or global enqueues. The default is SETSYS DFHSMDATASETSERIALIZATION, which will issue a volume reserve when processing a user data set. If you are running in a multi-image environment with GRS or equivalent product, then we recommend that you use SETSYS USERDATASETSERIALIZATION. This will allow the incremental backup performance benefits to be realized, as well as the Fast Subsequent Migration feature introduced in OS/390 2.10.

In addition, multi-volume, SMS-managed physical sequential data sets are supported only when SETSYS USERDATASETSERIALIZATION is specified. This is in order to prevent deadlock situations when volume reserves are issued.

Note: Only use SETSYS USERDATASETSERIALIZATION if GRS or similar product is actively propagating global enqueues to other images in the sysplex.

2.6.4 Control data set serialization

If you are going to run in a multiple host environment, there are keywords that will specify the method that DFSMShsm will use to serialize its control data sets. This will depend whether GRS or a similar product is actively controlling cross-system integrity.

The CDSQ and CDSR keywords in the DFSMShsm startup procedure direct how serialization is performed, either using global enqueue, reserving the volumes on which the control data sets are placed or using RLS to serialize the control data sets.

CDSQ keyword

CDSQ=YES will cause DFSMShsm to issue a global exclusive enqueue and allow multiple tasks within the single DFSMShsm address space on that host to access the data sets. All hosts running in the HSMplex must serialize in the same way and must also ensure that the QNAME of ARCENQG is propagated to all processors.

If you decide to run multiple instances of DFSMShsm on the same z/OS or OS/390 image then CDSQ=YES or CDSSHR=RLS must be specified. Running with HOSTMODE=AUX does not allow CDSQ=No.

CDSR keyword

Using CDSR=YES and CDSQ=NO, DFSMShsm serializes the control data sets using RESERVE logic. This requires that all DFSMShsm started tasks in the HSMplex must be specified as HOSTMODE=MAIN, otherwise error message ARC0006I is issued.

CDSSHR keyword

If CDSSHR=YES is used, then DFSMShsm will serialize according to the settings of the CDSQ and CDSR keywords. If CDSSHR=RLS is specified, then serialization is performed using RLS. CDSSHR=NO will cause DFSMShsm to not perform multiprocessor serialization.

Table 2-6 contains a summary of the options that are available:

Table 2-6 DFSMShsm serialization startup keywords

CDSQ keyword	CDSR keyword	CDSSHR keyword	Serialization
YES	YES	YES	Both CDSQ and CDSR options are used
YES	NO or not specified	YES	Only CDSQ option is used
With any other combination of specifications		YES	Only CDSR option is used
--	--	RLS	Uses VSAM RLS
--	--	NO	No multiprocessor serialization No other processor shares CDSs

When CDSQ and CDSR are specified, DFSMShsm monitors updates from each processor to ensure that the serialization technique matches the previous update.

Table 2-7 shows the resource names used by DFSMShsm to protect the CDSs.

Table 2-7 DFSMShsm resource names for CDSs

Major resource name (Qname)	Minor resource name (Rname)	Result
ARCENQG	ARCMCDS	Global resource serialization of the MCDS
	ARCBCDS	Global resource serialization of the BCDS
	ARCOCDS	Global resource serialization of the OCDS
	ARCRJRN	Ensures that only one processor can backup the journal

2.6.5 Serialization without a global enqueue product

If you run in a multiple processor environment without any resource serialization product, DFSMShsm uses volume reserve logic to serialize CDS processing. This prevents other z/OS or OS/390 images in the complex from updating the CDSs and the volumes on which the CDSs reside.

If you are running with this configuration, placement of the control data set components should be carefully prepared, to avoid lockouts and "deadly embraces".

2.6.6 Serialization of DFSMShsm functional processing

Table 2-8 shows the resource names used by DFSMShsm to ensure that only one image can process DFSMShsm-owned data.

Table 2-8 DFSMShsm resource names

Major resource name (Qname)	Minor resource name (Rname)	Serialization result
ARCENQG	ARCL1L2	Ensures that only 1 DFSMShsm host performs L1 to L2 migration
	ARCMCLN	Ensures that only 1 DFSMShsm host performs migration cleanup
	ARCBMBC	Ensures that only 1 DFSMShsm host performs backup versions movement
	RECYC-L2	Ensures that only 1 DFSMShsm host performs recycle of ML2 tape volumes
	RECYC-SP	Ensures that only 1 DFSMShsm host performs recycle of spill tape volumes
	RECYC-DA	Ensures that only 1 DFSMShsm host performs recycle of daily tape volumes
	EXPIREBV	Ensures that only 1 instance of EXPIREBV is running

2.6.7 RNAMEDSN keyword

If you are planning to run multiple HSMplexes within one sysplex, it is advisable to specify RNAMEDSN=Y in the DFSMShsm startup procedure. If there is any chance that you may run multiple HSMplexes at some stage then we advise that this parameter us used.

Multiple HSMplexes are defined as DFSMShsms that use more than one set of control data sets. Setting RNAMEDSN=Y means that the minor resource name (Rname) is changed to issue the CDS name as well as the function. This new translation technique is described as using dynamic resources as opposed to static resources.

All DFSMShsm hosts within the sysplex must use the same translation technique. If an inconsistency is detected, the detecting host immediately shuts down.

Table 2-9 lists the translated resource names when RNAMEDSN=Y is specified:

Table 2-9 Rname translation with RNAMEDSN=Y

Current Rname	Translated Rname
ARCBMBC	ARCBMBC&bcdsdsn
ARCCDSVF	ARCCDSVF&mcdsdsn
ARCCDSVD	ARCCDSVD&mcdsdsn
ARCL1L2	ARCL1L2&mcdsdsn
ARCMCLN	ARCMCLN&mcdsdsn
RECYC_L2	RECYC_L2&ocdsdsn
RECYC_SP	RECYC_SP&ocdsdsn
RECYC_DA	RECYC_DA&ocdsdsn
ARCxCDS	ARCxCDS&cdsdsn
ARCCAT	ARCCAT&mcdsdsn
ARCRJRN	ARCRJRN&jrnldsn
HOST||Hostid	HOST||Hostid&mcdsdsn
EXPIREBV	EXPIREBV&bcdsdsn
volser	volser&bcdsdsn
volser.TAKEAWAY	volser.TAKEAWAY&bcdsdsn

2.6.8 Serialization using VSAM RLS

Control data sets can also be serialized using VSAM record level sharing. The major benefits from this method will be seen during long-running tasks, such as secondary space management, EXPIREBV and AUDIT. For data movement functions, such as migration or backup, the performance benefits are not so noticeable, since a small percentage of the total processing time is spent updating the respective control data set records.

2.6.9 Control data sets VSAM shareoptions

The starter set creates the MCDS, BCDS, and OCDS with VSAM SHAREOPTIONS(3 3), which is in anticipation of DFSMShsm being used in a shared environment. When running in a shared environment, care must be exercised when utilities are run against the control data sets, even when using a cross-system enqueue product. We strongly recommend that utilities, such as a CDS reorganization run with DISP=OLD against the control data set. This will ensure that exclusive control of the cluster name is obtained using SYSDSN before update processing can begin.

If you are planning to run DFSMShsm on only one z/OS or OS/390 image in the sysplex, the control data sets can be defined with VSAM SHAREOPTIONS(2 3). If you choose to use this method, then the cross-system enqueue product should contain some statements to exclude SYSVSAM resources for the CDSs from being propagated. The statements are:

```
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)
```

2.6.10 Notes when using CA-Multi-Image Manager

The CA-Multi-image Integrity component of CA-MIM allows different ways to propagate enqueue and reserve requests. These can be processed either using automatic propagation of requests with a scope of SYSTEMS (PROCESS=ALLSYSTEMS) or allowing propagation of only the requests that you are selected (PROCESS=SELECT).

If you are running with PROCESS=SELECT, you will need to ensure that the following DFSMSHSM QNAMES are propagated as global enqueues:

- ARCENQG
- ARCPGA
- ARCDSN

In addition, please ensure that the following QNAMES are propagated:

- SYSDSN
- SYSVSAM
- SYSVTOC
  (This can also be converted from RESERVE to exclusive enqueue.)

2.7 Current storage management product

This section discusses the assessment of the current storage management product and details the information necessary for moving to DFSMShsm.

One of the steps that is usually performed during the conversion from your current product is the migration of data to DFSMShsm format. This is discussed in detail in Chapter 4, “Converting to DFSMShsm” on page 89.

2.7.1 Finding out the number of archive versions

In order to assess the work that is involved, it is necessary to separate the archive versions from the backup versions. This is done by running the reports from your current storage management system. In some cases additional steps are needed to separate the output data and can be achieved using a sort step which sorts based on the scratch flag.

As well as supplying the number of data sets that are involved, this information should also supply the current expiration dates of the archive versions. This information is crucial for two reasons:

- It details how many data sets are set to expire in the next weeks.
- It shows the original retention periods that are used to manage the data sets.

Using this information as input, a decision can be made on when to apply a cut-off date, that is, data sets after this date need to be converted to DFSMShsm and data sets before this date will expire in the current storage management product. It is important to remember that these data sets are still available for restore from the current product, since it will co-exist with DFSMShsm for as long as required.

The original retention periods may need to be calculated, depending on your current storage management product. The reports produced may show the original archive date and the expiration date of the data set version. A REXX procedure can be written to perform this calculation which can then be used in either designing management classes for the data sets or for assigning a data set to an existing management class.

The report should also include the size of the archived data sets involved. This allows you to work out how long the conversion of the archive versions may take This will very much depend on factors such as the number of tape drives available, seed disk volumes and system accessibility.

2.7.2 Finding out the number of backup versions

The initial considerations for backup versions are similar to the archive versions. The main differences will center around the number of data sets involved, which will usually be higher. Additionally a review is necessary in regard to the number of versions that require conversion. In general, the backup versions for most data sets do not require conversion, since these normally expire within a relatively short period of time. There may be a requirement for some data sets to be kept for an extended amount of time as well as the number of versions that are required.

As with archived copies, the backup retention period needs review as well as the number of copies of data sets that are retained. In general your current storage management product deals with an expiration date after which time the data set version is expired from the control file. As discussed in 1.3, “DFSMS design philosophies” on page 9, the number of backup versions is the key criteria to retaining a data set copy in the BCDS. If there is a requirement to go beyond the hundred versions available in DFSMShsm then aggregate backups can be used to maintain an infinite number of copies, for example. In addition ABARS allows specification of a parameter such as years, months, weeks or days for only aggregate and extra aggregate backup versions.

Another significant difference is that the original data set very likely still exists on primary disk. This aspect is discussed in more detail in 4.6, “Converting backup versions” on page 109.

2.7.3 Reviewing the number of tapes currently used

In order to better gauge the amount of data stored in your current storage management product, it is necessary to determine the number of tapes that are in use today. This figures into the total amount of time required for conversion of the data, although there are certain aspects which will affect this number.

- The tapes may include expired data: As data sets versions age on the tape, they are deleted from the control file of your current storage management product. This can mean that a number of the tapes may not be filled to capacity.
- The tapes may require multiple mounts if both backup and archive data sets reside on them: Depending on the methods used to manage data, a tape may contain both archive and backup data.
- The tapes may no longer be available: It is possible that some tapes are either lost or have been overwritten. In this case they will need removing from the control file.

2.7.4 Reviewing the current processing window

In order to define similar processing times in DFSMShsm, it is necessary to evaluate the current processing times that are used for current storage management functions. Since these times are most likely defined to work around your other batch and production processing, they are used to decide the best time to run the various DFSMShsm activities. The start times of the various cycles can be transferred to the ARCCMD00 member and the number of tasks for the functions set according to the number of tape or disk devices available for use.

Depending on the number of data sets that are modified in a given day, the function that is likely to require the most exact scheduling is automatic backup. It is best to run this at a time when as few as possible data sets are open for update processing so that the maximum number of them are processed. In the cases where the data set is always open, then other methods can be considered. These are discussed in detail in 4.4.5, “Backup considerations for data sets in use” on page 97.

If you are currently running a daily archive process against the primary disk volumes, this can be mirrored by setting the primary space management to run at a similar time. Since migration is not the only function that the space management task performs, you may want to allow migrations to run more often, In this case, interval migration is an option and this is run every hour. Please remember that data set movement is only performed when required by the threshold parameters of the storage group or volume. In this case eligible data sets are migrated as required.

2.7.5 Volume dumps

If you are using full volume dumps with your current storage management product, these can also be automatically scheduled in DFSMShsm. For more information, please refer to 3.14.3, “Automatic dump” on page 84.

2.7.6 Recycling tapes

A common process that is needed is the consolidation of tapes to remove gaps encountered when logical files are expired from the media. In DFSMShsm this process is known as *recycle* and an equivalent function is probably available in your current storage management product. This may be known as *merge* or *forward merge*.

Since this function processes offline media, this task normally runs outside of normal batch windows, either during the day or at weekends. Equivalent parameters exist in DFSMShsm to ensure tapes are selected by type (backup or migration) and also current percentage of valid data.

2.7.7 Control data set maintenance

It is important to know how often the maintenance utilities run against the control file of your current storage management product. This helps to determine how accurate the numbers are from reports that are run and prevents expired data set versions from requiring conversion to DFSMShsm. For example, if this process is only run once a month then some data sets may be needlessly converted.

2.7.8 Special processing

It is important at this stage to identify potential applications or data sets that require special processing. For example, if your current implementation has a particular job flow that must be adhered to, it is necessary to note these for special attention. It may be necessary to exclude these data sets from automatic backup, for example, and use command level functions to process them.

2.7.9 User exits and other modifications

In some cases you may have used customized user exits with your current storage management product to achieve a specific objective. It is necessary to review any exits or other modifications to see if similar changes are required or even needed.

2.8 Additional considerations

In the following sections we include additional information useful when working with DFSMShsm.

2.8.1 SMF records

DFSMShsm allows the capability of writing system management facility (SMF) records. This is performed by use of the SETSYS SMF parameter.

The records that are written are:

- Function Statistic Record (FSR)
- Daily Statistic Record (DSR)
- Volume Statistic Record (VSR)
- ABACKUP/ARECOVER Function Statistic Record (WWFSR)

The *DFSMShsm Implementation and Customization Guide* contains information on the format of the records.

In the SETSYS SMF parameter, you should supply an available user SMF record number in the range of 128 to 255. You should also note that DFSMShsm writes to two record numbers, namely the one that is supplied in the SETSYS SMF parameter and also to the SMF number+1. For example, if SETSYS SMF(223) is specified, then DFSMShsm will use SMF record numbers 223 and 224. DSR and VSR records are written to number 223 while FSR and WWFSR are written to record number 224.

If you are planning to use the DFSMS Optimizer, then you will need to activate SMF recording. For more information on the features of this product, please see *DFSMS Optimizer User's Guide and Reference*.

If you do not use DFSMS Optimizer you can also use the REXX sample program FSRSTAT, provided in SYS1.SAMPLIB(ARCTOOLS). This program will read a set of FSR records and generate a summary report.

2.8.2 Comparing DFSMShsm terminology with other products

You will already have noticed that the terms used to describe functions in DFSMShsm are sometimes different from those used in your current storage management product. In Table 2-10 we provide a cross-reference point to some of the more common terms used:

Table 2-10 Comparison of terminology between storage management products

DFSMShsm	CA-Disk	FDR/ABR	CA-ASM2	Description
Migration control data set (MCDS)	Files data set (DSNINDEX)	Archive Control File (ACF)	Integrated Product Catalog (IPC)	Control file containing information about data sets that scratched after backup
Backup control data set (BCDS)	Files data set (DSNINDEX)	No exact match. Information stored in various places, including "ABR catalog"	Integrated Product Catalog (IPC)	Control file containing information about data set backup versions
Offline control data set (OCDS)	Files data set (ARCHVOL)	Part of ABR catalog	Integrated Product Catalog	Control file containing information about migration and backup tapes and about each data set on these tapes
Migrate	Archive	Archive	Archive	Action of moving a cataloged data set from primary disk to ML1 or ML2, or from ML1 to ML2

DFSMShsm	CA-Disk	FDR/ABR	CA-ASM2	Description
Backup	Backup	Backup	Backup	Process of copying a data set residing on primary disk to a backup volume
Level 0	Primary disk (DASD)	Disk	Disk	Online disk volume containing data directly accessible by the user.
Level 1	Disk archive	Disk archive	Disk Staging Area (DSA)	Disk volume under DFSMShsm control containing data sets migrated from Level 0
Level 2	Tape	Tape	Tape	Volume (usually tape) under DFSMShsm control containing data sets migrated from Level 0 or level 1
Recall	Restore	Recall	Reload	Action of moving a migrated data set back to Level 0 from Level 1 or 2
Auto-recall	Auto-Restore (DMSAR)	ABR auto-recall	Intelligent Transparent Restore (IXR)	Recall initiated automatically, transparent to the requesting task
Recover	Restore/Recover	Restore	Restore	Action of recovering a single or multiple data sets to level 0 from backup volumes
Full Volume Dump	VBACKUP	Full Volume Backup (FDR)	$DEFRAG	Process using DFSMSdss that backs up the entire allocated space on a disk volume
Aggregate backup	Backup, including XCOPY	FDRAPPL	Backup	Process of copying data and control information from a user-defined group of data sets
Alternate tape volume	Duplicate copy tape	Duplicate copy	$COPYTP	Copy of original tape volume, created simultaneously or subsequent copy process
Recycle	Merge	FDRTSEL	Forward merge	A process that copies all valid data on a tape to a tape spill backup or ML2 volume
Expiration	Delete/Expire/ IXMAINT	Superscratch/ FDRARCH-DELETE FDRARCH-EXPIRE	Controlled Scratch/$MAINT	The removal of a user data set from a primary volume or the deletion of a migrated data set version

3

DFSMShsm activation

This chapter is written with the assumption that you have examined your storage management policies and supporting processes described in Chapter 2, “Preparing your environment” on page 13, and you feel comfortable proceeding to the next phase.

You will be performing the necessary activities to start the DFSMShsm address space on your systems. In some customer shops, this phase is performed by the storage administrators; in others, this is done by the system programmers; and in many shops, it will be done by a combination of these two groups. Activation will co-exist with your present systems, so from an end-user/operational perspective, nothing will be different until after you have completed the activation and are actively involved in conversion.

In this chapter we discuss:

- How to move from preparation to DFSMShsm activation
- How to make DFSMShsm operationally active in your environment
- Areas to set up or investigate prior to converting to DFSMShsm

3.1 Catalog environment

As said in the introduction, co-existence of DFSMShsm with any other storage management product is not possible if the data sets to be managed are not cataloged. In some environments this is easy to say but difficult to do. One of the ways that a data set can be uncataloged is if the catalog structure is not the same across all MVS images sharing disk. One way to avoid this situation is to maintain only one MVS master catalog and share it among all LPARs sharing disk. Some customers view this as a weak design creating a single point of failure. The obvious strength of this setup is that there is no chance to have the catalog structure different on any sharing LPAR.

For those customers that prefer to have a single master catalog per LPAR, we provide a catalog analysis program (ALIASCHK) that will examine the master catalogs of two systems and in the cases that the structure is different, will look to see if it is possible to automatically correct. The program uses REXX, IDCAMS, and SORT, and it requires that all master catalogs to be examined be *IMPORT connected* to a single system and all volumes containing master catalogs be online to the same system.

The program is provided with this redbook as additional material. For information on how to get the additional material, refer to Appendix C, “Additional material” on page 149.

Example 3-1 shows the parameters passed to program ALIASCHK.

Example 3-1 Parameters passed to ALIASCHK

```
//*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
//ALIASCHK PROC S1NAME='SC47',                    * SYSTEM ID #1
//              S1MCAT='MCAT.OS390R10.VO1OCAT',  * ITS MASTER CATALOG
//              S2NAME='ZOS1',                    * SYSTEM ID #2
//              S2MCAT='MCAT.OS3R9V01.VO39CAT'   * ITS MASTER CATALOG
//*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
```

The program will compare the alias structure of two systems, looking for three possible causes of these catalogs being out of sync:

1. Alias defined to one system but not the other.

   Create correction to define missing alias.

2. Alias defined to both systems but not pointing to the same user catalog.

   Check catalogs for entries:

   – If none exist or if exist in only one catalog, create correction to define alias.

   – In entries are in both catalogs then produce report detailing present structure, only manual correction possible.

3. Alias defined to both systems and the same user catalog, but user catalogs on these systems are on different volumes or device types.

   Check catalogs for entries:

   – If none exist or if exist in only one catalog, create correction to define alias.
   – In entries are in both catalogs then produce report detailing present structure, only manual correction possible.

The corrections are passive in that they are written to a file that can be used as SYSIN to a IDCAMS job to perform correction. The program has the possibility to create the wrong correction. In all cases it is assumed that the missing alias is to be defined. It may have been the intent to delete rather than define the alias, in which case the program will “correct” this with define alias rather than delete alias commands. It is always best to look at the corrections that are being created and look at why these particular entries are out of sync between the systems. This usually points to a broken process that needs repair.

Example 3-2 shows the output from the Phase 1 of ALIASCHK execution. This is known as the initial catalog analysis.

Example 3-2 ALIASCHK: initial catalog analysis (phase 1)

```
UCAT.VTC1DB1                                on TC1DB1 is likely another master catalog
UCAT.VTC2DB1                                on TC2DB1 is likely another master catalog
UCAT.VTC3DB1                                on TC3DB1 is likely another master catalog
2 systems found
172 usercats on SC47
166 usercats on ZOS1
ANY.TSTCAT TOTSMS                           is a catalog on SC47 with 1 aliases
CAT.CICS DACCI1                             is a catalog on SC47 with 1 aliases
CAT.DB2 DACDB1                              is a catalog on SC47 with 1 aliases
```

Example 3-3 shows a report of the errors found in the comparison.

Example 3-3 ALIASCHK: comparison errors for correction

```
ILSR9    is not in SC47 mcat  MCAT.ILS9CAT
IMS710   is not in ZOS1 mcat  UCAT.VTOTIM6
IODF     is not in ZOS1 mcat  UCAT.V#@$#M1
MQ52BLD  is not in SC47 mcat  UCAT.VT2MQ52
NOPE     is not in ZOS1 mcat  CATALOG.TOTICF1.VTOTCAT
OMVSZ11  is not in ZOS1 mcat  UCAT.VT2Z1S1
SAPBLU   is not in ZOS1 mcat  UCAT.VSAPBLU
SMPZ11   is not in ZOS1 mcat  UCAT.VT2Z1S1
S1BASE   is not in ZOS1 mcat  CATALOG.TOTICF1.VTOTCAT
S1HFS    is not in ZOS1 mcat  CATALOG.TOTICF1.VTOTCAT
S1INSTG  is not in ZOS1 mcat  CATALOG.TOTICF1.VTOTCAT
S1TEMP   is not in ZOS1 mcat  CATALOG.TOTICF1.VTOTCAT
TESTCENT is in both CATALOG.TOTICF1.VTOTCAT on SC47 and UCAT.V#@$#M1 on ZOS1
```

Example 3-4 shows a sample report to help in the manual correction.

Example 3-4 ALIASCHK: sample report to aid in manual correction

```
Catalog Conflict DSN Detail Report for         12/04/01    Page: 1
 Dataset Name                                UNIT VOLSER DSNType DSORG Alloc
-------------------------------------------- ---- ------ ------- ----- ------
TESTCENT.ATAV7.DATA                          DISK TOTTSW VSAM        . .
TESTCENT.ATAV7.DATA                          DISK TOTTS6 VSAM        . .
TESTCENT.ATAV7.INDEX                         DISK TOTTS6 VSAM        . .
TESTCENT.ATAV7.INDEX                         DISK TOTTSW VSAM        . .
TESTCENT.JCL.CNTL                            3390 TSMS19 NONVSAM    PO BLOCK
TESTCENT.JCL2.CNTL                           3390 TOTTSW NONVSAM    PO BLOCK
```

Example 3-5 shows the contents of the file where the correcting commands are written. There is a sample member, ALIASUPT, included in the library provided with this book.

Example 3-5 Sample corrections

```
DEF ALIAS(NAME(IODF    ) RELATE( UCAT.V#@$#M1))-
      CAT(MCAT.OS3R9V01.VO39CAT)
DEF ALIAS(NAME(MQ52BLD ) RELATE( UCAT.VT2MQ52))-
      CAT(MCAT.OS390R10.VO10CAT)
DEF ALIAS(NAME(NOPE    ) RELATE( CATALOG.TOTICF1.VTOTCAT))-
      CAT(MCAT.OS3R9V01.VO39CAT)
DEF ALIAS(NAME(OMVSZ11 ) RELATE( UCAT.VT2Z1S1))-
      CAT(MCAT.OS3R9V01.VO39CAT)
DEF ALIAS(NAME(SAPBLU  ) RELATE( UCAT.VSAPBLU))-
      CAT(MCAT.OS3R9V01.VO39CAT)
```

It was for simplicity of program logic that only two systems are compared at a time. More sharing systems than two can be analyzed by multiple executions and after the corrections are made that no discrepancies are reported (if a=b and b=c then a=c).

Once we can be assured that a valid shared catalog environment exists, then we can proceed with further activation activities and establishing a policy to delete uncataloged data sets.

3.2 DFSMShsm co-existence

Typical allocation of an existing disk data set is a sequence of two MVS macros prior to OPEN: catalog LOCATE, and VTOC OBTAIN. The result of a successful LOCATE (for allocation purpose) is the volser on which the data set resides (or the volser list, if multivolume). The result of a successful OBTAIN is the location of the data set on the volume.

DFSMShsm and all the other storage management products have hooks in the DFSMS module IGG026DU (catalog LOCATE exit). Providing all storage management products keep this code in place, it is possible to recall/restore data from whatever product has the data. Typically, a non-VSAM catalog entry with a volser of MIGRAT or ARCIVE will drive this exit for the look-up and consequent recall of a data set.

All products that have migrated data are able to recall data using this exit, and each product in turn will look-up the data set and either restore/recall the data set or allow the next product to attempt retrieval. This proceeds until we have exhausted all look-ups possible (catalog, storage management product, DFSMShsm) and the locate fails. Consider this a detailed definition of what is an incorrectly cataloged data set (if we take the path all the way to “fail locate”).

Refer to informational APAR II01949 for details concerning the co-existence between DFSMShsm and CA-ASM2. This involves the use of a dummy module $IGG26DU.

If the catalog entry for an archived/migrated data set still points to the original volser, then likely we are converting from FDR/ABR and the data set is restored from the FDR/ABR exit FDREXDSN (data set not found exit). This exit will be driven in the case of an OBTAIN failure. This retrieval is still front-ended with catalog LOCATE, prior to calling FDREXDSN, so there is no problem with retrieval of migrated data from either product provided the data set is cataloged.

The only issue that we have found with co-existence is a conflict between the DFSMShsm component ABARS and CA-Disk Catalog Transparency. ABARS has the ability to back up migrated data. Migrated data is identified to ABARS by the return of a volser MIGRAT from catalog LOCATE. The capability introduced by CA-Disk for Catalog Transparency is to return MIGRAT as the volser for its ARCIVE data.

There exists a usermod to apply to CA-Disk for allowing ABARS to see volser ARCIVE for CA-Disk migrated data and MIGRAT for DFSMShsm data. The usermod is located in the CA-Disk library ADMSSAMP as member USERMODB. The program name “ARCWCTL” (ABARS program) needs to be added to this exit and the exit assembled/linked back to the CA-Disk load library. With this exit in place, an ABARS aggregate looking to exclude ARCIVE data will see that it is on volume ARCIVE and not MIGRAT. This data can now be identified and excluded from the aggregate.

3.3 DFSMShsm enablement

In OS/390, the IGDDFPKG member of PARMLIB is no longer used to enable DFSMS licensed components. Instead, IBM supplies a tailored SYS1.PARMLIB member, IFAPRD00, to enable the elements and features ordered with OS/390. The IFAPRD00 member is not active by default. For enablement, you must copy the contents of the sample IFAPRD00 to an active IFAPRDxx member that you establish through the PROD parameter in IEASYSxx or the **SET PROD** operator command. The DFSMShsm entry could look as shown in Example 3-6.

Example 3-6 DFSMShsm entry

```
PRODUCT OWNER('IBM CORP')                    /* EFFECTIVE OS/390 R1 */
        NAME(OS/390)
        ID(5645-001)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(DFSMSHSM)
```

Characteristics and syntax rules of IGDDFPKG, along with details about IFAPRDxx, are explained in the *MVS Initialization and Tuning Reference*.

3.4 Developing a DFSMShsm implementation plan

You can find a sample implementation plan in the publication *DFSMS/MVS V1R5 Planning for Installation SC26-4919-06.*

Using the supplied starter set in SYS1.SAMPLIB, it is possible to perform all the necessary activities to start the DFSMShsm address space for the first time in an afternoon. Most customer environments have requirements that would place additional activities on the implementation of any product, so you are not complete in an afternoon. Here are some areas to consider.

- User training

  People in this group usually fell into categories with titles like:

  - Application programming
  - Database administration
  - JCL analysts
  - Production control

  DFSMShsm is a mature product and many people have had experience with this product (possibly in another job). There are seven DFSMShsm commands that anyone can issue from TSO or batch:

  - HMIGRATE
  - HRECALL

- HBACKDS
- HRECOVER
- HLIST
- HDELETE, and
- HBDELETE

We have excluded the HALTER command because it only applies to non-SMS managed data sets. Help is available for each of these commands in TSO to remind users of syntax, function, and operands. We provide as additional material a file (called HSM) that you can include in SYS1.HELP and use for listing the DFSMShsm user commands in case they are forgotten. So, users only have to remember one command to get all DFSMShsm commands: **`TSO HELP HSM`**. For information on how to get the HSM member, refer to Appendix C, “Additional material” on page 149.

Training users is vital to receive full benefit of this conversion. They can set their service level expectation on the contents of the assigned DFSMS management class. They can be instrumental in modifying the management classes to better align this function with present business requirements and they can now issue the HRECOVER command to recover their own data sets should this be necessary. This training may also influence present and future application design to take advantage of the new environment.

- Operator training

  In addition to user training, additional training can be provided to computer operations. DFSMShsm is a started task in your complex rather than a series of scheduled batch jobs. Operators will not be able to prevent the scheduled execution of storage administration activities by draining job initiators. In some customer environments there is a value to train the operations staff in how to operate DFSMShsm and in some cases, modify the schedule of activities. Other customers would prefer to automate all corrective processes and train their operations staff to recognize problems. Any DFSMShsm implementation represents a significant change on the operation of a system and in either event, warrants operator training. Operators need to be aware of scheduled times for automated functions, expected/normal console messages and at least how to query DFSMShsm to insure that it is operating properly.

- Technical support staff training

  People in this group usually fell into categories with titles like:

  - Systems programmers
  - Storage administrators
  - Computer security
  - Storage hardware purchase administrator
  - Contingency planner
  - Business continuance planner
  - Disaster recovery administrator

There will be some people in this group already involved with the planning and execution of this project. Other people in this group generally have the need to know about the product(s) and the current plans. Some of them will need in depth knowledge to perform their daily assignments. There is education available from IBM (and other sources) to help these people become proficient with the products you are introducing.

- Change control procedures

  Procedures will vary from customer to customer, but we are activating a product that will require updates to system data sets; changes to the security system, and will soon be a required started task in your environment. Planning for the required lead times and getting approvals to make these (and possibly other changes) are necessary elements of the implementation plan.

- Testing schedules

  In some customer environments, there are stringent rules concerning when new software can be tested. These rules have the potential to effect any implementation plan if (for example) first use testing is required to be conducted only during off-prime times.

- Making DFSMShsm a required address space at startup

  Once DFSMShsm has been allowed to migrate data sets in an environment, it is a requirement that the started task be available to allow allocation to retrieve a migrated data set. All LPARs sharing catalogs need the DFSMShsm address space started at IPL (usually soon after JES starts). The DFSMShsm address space can also be monitored (as you would any other critical address space) to see that it stays up and available until the next scheduled IPL. This activity visibly transitions the project from testing to production usage.

3.5 Planning the processor environment

DFSMShsm requires a designated *primary host* processor and all other sharing processors are *secondary hosts* (with possible designation to perform primary takeover if necessary). The role of the primary host is to perform the backup of the DFSMShsm control data sets and journal at the beginning of automatic backup, moves the backup data sets from ML1 to tape and deletes expired dump copies. Because of these duties, the primary host must specify ABSTART and ADSTART parameters. These functions may be shared among the MVS images in the HSMplex by also specifying ABSTART and ADSTART commands on the additional DFSMShsm hosts. This assumes that you want to use automatic backup and automatic dump functions of DFSMShsm.

All other DFSMShsm scheduled functions may be designated or shared among the MVS images in the HSMplex. It is through the ARCCMDxx PARMLIB member and the DFSMS storage group definition that these functional assignments are made (or changed). It is through the DFSMShsm starting procedure that the host designation and primary/secondary assignment is made.

The selection of which host to designate as *primary* can range from the least used LPAR to the LPAR with the highest level of connectivity. Often you will look to see where the majority of the predecessor jobs were ran and if this MVS image is a candidate to be the *primary host*.

3.6 Planning for data security

Beyond the existing security already in place, DFSMShsm is to be considered a trusted system. Authorization to use the more sensitive DFSMShsm commands are provided in the product via the AUTH command. All DFSMShsm functions that take place can be recorded in SMF records, control data sets, journal, logs and PDA trace files.

Security is checked prior to any DFSMShsm HRECOVER command (recover a data set from a backup). The default security authorization expected for user issuing the HRECOVER command is CREATE with respect to the data set being recovered.

Security is not checked prior to data set recall. Security is performed as usual when OPEN occurs.

You can find detailed information about DFSMShsm security in the publication *DFSMShsm Implementation and Customization Guide*. Also, you can refer to *DFSMSdfp Storage Administration Reference* for more information about SMS security (as well as in *DFSMS Implementing System Managed Storage*). Another source of information regarding security implementation are *DFSMSdss Storage Administration Guide* and *DFSMSdss Storage Administration Reference*. It is a good idea to review this documentation periodically as product upgrades become available.

3.7 Preparing DFSMShsm data sets and volumes

DFSMShsm will require enough space for allocation of control/support data sets and at least one ML1 volume. These can be shared with other data sets providing a function like GRS is enabled to allow the propagation of ENQs across the systems in the HSMplex. Sharing the allocation of the CDSs with other data sets on the same volume is typically not done to avoid possible contention.

There are worksheets in the *DFSMShsm Implementation and Customization Guide* to help in sizing the DFSMShsm CDSs, and information gathered in the preparation phase will supply an accurate estimate of the number of data sets and tapes involved.

The only advise on sizing the journal is that it be large enough to contain 24 hours of DFSMShsm activity. At the beginning of the automatic backup start, the CDSs and journal are backed up and the journal is reset. Should the journal fill in the middle of processing, no further DFSMShsm update activity is allowed until CDSBACKUP has completed and the journal is reset. In most environments, an unscheduled CDSBACKUP is a painful alternative to an adequately sized journal data set.

3.8 Setting availability management policies

This activity refers to policies defined to either DFSMShsm or in various management classes to back up data sets. Areas of interest are:

- If backup is to be done
- How frequently backups are to be done
- How many backup versions to keep
- What time of day is this to be scheduled
- Can an exclusive ENQ be used when data is backed up
- How many backup tasks (tape drives) need to run concurrently

One big advantage to having DFSMShsm manage the backups is that users can recover their own data sets and select restore options to be performed.

Refer to Chapter 2, “Preparing your environment” on page 13, for specific details.

3.9 Setting space management policies

This activity refers to policies defined for migration criteria and data set retirement. Areas of interest are:

- If migration is to occur
- If default data set expiration is to occur.
- When does migration occur
- Do we use disk/tape or both to migrate
- What time of day is this activity to be scheduled

Refer to Chapter 2, “Preparing your environment” on page 13, for specific details.

3.10 Setting up tape management for DFSMShsm

This activity refers to the definition of the tape resources that will be made available to DFSMShsm. Areas on interest are

- What device type is to be used?
- Is hardware compression to be used?
- Will DFSMShsm be an External Data Manager (EDM) to the tape management system?
- Is duplexing or tapecopy desired? Will the copy be onsite or offsite?

Refer to Chapter 2, “Preparing your environment” on page 13, for specific details.

3.11 Tape operations

The typical set up for DFSMShsm is to be defined as a scratch tape application. As opposed to defining tapes for DFSMShsm to use via ADDVOL commands, DFSMShsm is set up to request scratch tapes as needed (SETSYS TAPESELECTION(SCRATCH)). The first use of any tape by DFSMShsm will request a scratch tape. This set up works well with manual and automated tape libraries as the product should be writing to tape far more frequently than needing to read from tape. It is also very typical for a DFSMShsm installation to be defined to a certain tape management products as an External Data Manager (EDM). This removes all controls in the tape management product to expire the tape. Tape expiration will only occur when the EDM requests it. This is done with a DFSMShsm exit (ARCTVEXT) that is typically supplied with tape management systems other than DFSMSrmm.

Unlike the competitive products, DFSMShsm is uniquely enabled to optimize tape usage. DFSMShsm can continue to write to the same tape volser until the tape volume is full (SETSYS PARTIALTAPE(REUSE)). Should the tape be needed by another user (like data set recall), DFSMShsm has the capability of *tape takeaway* to allow recall to be the next use of the tape.

As data sets get recalled from tape (ML2) or backup versions roll-off, areas on ML2 and backup tapes no longer contain valid data. DFSMShsm has the ability to RECYCLE tapes to improve the tape utilization by rewriting valid data to new (scratch) tapes and releasing the previously used volumes. This consolidation helps to limit the number of tapes needed to support DFSMShsm functions.

3.12 Testing DFSMShsm

The initial startup of DFSMShsm should not interfere with the present storage management procedures until we are ready for conversion. Any startup of the DFSMShsm address space will, by default, attempt to manage data based on information available in the DFSMS ACDS. It is desirable to prevent these actions at this time. The purpose of this testing is to ensure that DFSMShsm is functionally enabled on your site and that it is working as intended. To this end there are various approaches available.

3.12.1 DEBUG/NODEBUG

This DFSMShsm start-up option allows DFSMShsm to go through the motions of automatic commands and produce activity logs as though these processes have occurred, but will not backup or migrate any data. Only commands that apply to a single data set will actually be performed. These commands can be used to test the validity of DFSMShsm definitions. You can change the way DFSMShsm is working at anytime via the **HSEND SETSYS...** TSO command.

```
HSEND SETSYS DEBUG
HSEND SETSYS NODEBUG
```

3.12.2 Storage group assignment

As an alternative conversion technique (or used together with DEBUG) is placing specifications in the storage group fields **Migrate System**, **Backup System,** or **Dump System**. DFSMShsm uses these fields to assign system affinity to perform these actions and other storage management products effectively ignore this field. If no value is placed in the field, then any DFSMShsm started could perform any activity against this storage group.

For example, if the **Dump System** field on the storage group is blanks then any DFSMShsm address space with a non-zero ADSTART parameter specified is eligible to select volumes from this storage group to dump. In an active DFSMS managed environment, we can place the name of a non-existent system in these fields to prevent DFSMShsm from managing this storage group as the product will only manage this storage group when the DFSMShsm address space with that system name is started. The method of conversion used later in this scenario will encompass a storage group at a time.

3.12.3 DFSMS conversion too

If we are converting from a non-SMS managed environment to an DFSMS managed environment, it is then possible to take advantage of this fact in testing. In effect we have a boundary between the disk volumes defined in DFSMS managed storage groups and the present volume pools in existence. The only possible collision between storage management products is if this boundary gets blurred.

Other storage allocation management products reside in the DFSMSdfp allocation exit IGGPRE. This exit gets control after the DFSMS ACS routines have executed and therefore the DFSMS ACS assigned values are available to these products. One technique that we have used with success is to add a rule at the beginning of the other storage allocation management product's assignment code that will test for a non-null storage class value and exit if true. In other words, this is intended to prevent DFSMS and the other storage management product from managing the same data set. If DFSMS is managing this data set (as evidenced by a non-null storage class value assigned) then the other product is to exit immediately and perform no further services.

To cover the alternative possibility, there are never to be any ADDVOL statements for primary volumes in the DFSMShsm startup.

Once this boundary is established, we are free to proceed with testing both DFSMS and DFSMShsm with storage groups created from a subset with a small number of volumes. We can see from our testing that allocation is occurring as intended, that DFSMShsm is properly allocating the disk storage groups and tape resources assigned to it and all automatic and command processed work as expected. The volume lists used by the other storage management product remain unchanged, and you will need to be certain that the DFSMS managed volumes do not conform to the other storage management products volume filters. We can proceed from here with conversion, and when the storage pools managed from the other storage management product are empty, these jobs will end with a non-zero condition code (a flag that the job no longer serves its purpose).

3.12.4 Further testing

DFSMShsm provides in SYS1.SAMPLIB both an Installation Verification Procedure (IVP) and a Functional Verification Tests (FVT). They provide any installation a healthy testing scenario for DFSMShsm processes.

Also there are *patches* DFSMShsm to allow daily commands (like automatic backup) execute more than once per day. By resetting these fields, the same automatic function can be tested iteratively.

Testing procedures and patches are documented in *DFSMShsm Implementation and Customization Guide*.

3.13 Customizing and tuning DFSMShsm

After a successful DFSMShsm implementation and testing, there are additional activities that you may want to use. There are a number of exits that provide specific capabilities. In the following sections we will review the additional customizations you may want to implement.

3.13.1 Exits

There are many exits available in the DFSMShsm product. These exits are documented in *DFSMS Installation Exits* and samples are shipped in SYS1.SAMPLIB.

Typical DFSMShsm implementations desire customized capability in the DFSMShsm product, you can achieve some of these capabilities implementing the DFSMShsm provided exits. The following exits are of common use.

- ARCTVEXT, tape volume exit
- ARCMVEXT, space management volume exit
- ARCBDEXT, data set backup exit
- ARCMDEXT, prevent migration of small (less than 150MB) data sets from primary disk
- ARCMMEXT, prevent migration of small data sets from ML1 to tape
- ARCCBEXT, perform post processing after CDS backup has completed. Customers often backup to disk then use this exit to make a second copy to tape for offsite storage.
- ARCRPEXT, assign priority to recall requests. This exit works with deletes and recovers too, but for most customers, the recalls are most important.

Exits may be enabled and disabled as needed via a TSO command, or be always enabled in the ARCCMDxx member at start-up.

Usually the ARCTVEXT exit is provided with the tape management system product, when other than DFSMSrmm. The exit communicates to DFSMShsm when the tape management system release a tape, typically the input tapes used for RECYCLE, or as the result of a **`DELVOL ... PURGE`** command. If the tape management product is DFSMSrmm, then the EDGTVEXT programming interface is called automatically and no exit is needed.

3.13.2 Patches

The PATCH command allows changing DFSMShsm control blocks located within the DFSMShsm address space. This command can also help customers in need of an immediate fix to the product to apply a correction provided by the IBM support center in advance of the APAR fix availability.

A list of supported patches is located in the publication *DFSMShsm Implementation and Customization Guide*.

3.14 Coordinating DFSMShsm processing

One of the pleasant aspects of implementing DFSMShsm is that much of the decision work to perform storage management functions is simplified by the product. There are only four automatic functions to be set-up with start times and another one that has no start time (interval migration).

3.14.1 Interval migration

We think of interval migration as a good thing. It lets DFSMShsm be responsive to dynamic customer environment and provides some assurance that primary space management will complete in its window. By default, interval migration begins 15 minutes after the hour and will run to completion. The trigger level to begin processing a volume is when the allocation on the volume exceeds the mid-point between the storage group's high and low threshold.

If triggered, interval migration will begin migrating eligible data sets off the volume until the low threshold is reached or no more data sets are eligible for migration. If interval migration is still running at the next scheduled time, it is not started and the prior execution is allowed to complete. Interval migration is also inhibited from executing during primary space management and during CDS backup.

In environments with active batch/TSO users, this can provide the additional space needed and avoid an out of space condition. Interval migration is also integral part of TMM storage groups. The DFSMS storage group functionality allows interval migration to be selectively associated (interval migration will process some storage groups and not others).

3.14.2 Secondary space management

This is the scheduled time for data sets to move from migration level 1 (disk) to migration level 2 (tape). All ML1 volumes defined to DFSMShsm are examined to see if their high threshold is exceeded and if true, then all eligible data sets are moved to tape. The ML1 high threshold value is specified on the ADDVOL statement for the ML1 in ARCCMDxx. A sample statement could be:

```
ADDVOL UNIT(3390) MIGRATION(ML1 AUTODUMP(SYSPLEX1) SDSP) THRESHOLD(95)
```

The typical schedule for this activity is over lunch time. The reasons are: 1) we need to do this daily to support interval migration, primary space management, and command backups, 2) we need to choose a time when tape drives are available, and 3) we prefer to choose a time when we know DFSMShsm activity is low. In addition to promoting data sets to ML2, we will also be performing *migration cleanup* (deleting records from the control data sets that are no longer needed). If necessary, it is possible to run secondary space management more than once per day from a different host.

3.14.3 Automatic dump

This activity is generally scheduled as needed. Dump classes and dump cycle definitions split up this activity to various days for different volumes. This can be scheduled as often as daily or as infrequently as once per month. There are probably jobs set-up and scheduled on your system to perform this function. There is nothing that says you must use the automatic dump as supplied by DFSMShsm. The major advantage of DFSMShsm automatic dump is the registration of the data sets in the dump (for auditing and local restoration), the ability to specify dump stacking, and the enablement of the dump class at the storage group definition. This last point allows you to add volumes to the storage group and not require you to set up jobs to dump the new volumes (it happens because it is an attribute of the storage group).

3.14.4 Automatic backup

Automatic backup is usually scheduled to start somewhere in the early evening to just after midnight. Automatic backup consists of four phases performed in order:

1. CDS backup and journal resetting. All other DFSMShsm functions are inhibited during this time.
2. Movement of existing command backups from ML1 to tape.
3. Movement of backups for migrated data sets requiring a backup.
4. Backup of changed data sets in the primary storage groups based on management class definitions.

In all phases, the roll-off of old backup versions is performed if the backup is successful. Whenever you choose to start automatic backup, the rest of the storage management schedule evolves around this automatic function. It is best to be done when tape drives are available and only the primary host performs the CDS backup, so that other hosts can participate in automatic backup without incurring that activity. All DFSMShsm installations should schedule automatic backup on at least their primary processor, if for no other reason, to have DFSMShsm perform the CDS backup and journal reset.

3.14.5 Primary space management

Primary space management is typically started after automatic backup and if automatic backup is scheduled to complete before midnight, then if primary space management starts just after midnight will have an additional days worth of data sets eligible to migrate. DFSMShsm will select volumes to perform primary space management if the allocation on the volume is over the low threshold, and will stop processing the volume when the allocation is under the low threshold. Primary space management is performed in two passes of the volume. In the first pass of the volume DFSMShsm will:

- Delete temporary data sets (based on SETSYS SCRATCHFREQUENCY) specification).
- Delete expired data sets (based on SETSYS EXPIREDDATASET(SCRATCH) specification).
- Identify candidates for migration or extent reduction.

In the second pass, DFSMShsm will migrate data sets in the following sequence, until the low threshold is reached:

- Data sets that are candidates for both migration and extent reduction.
- Data sets that are candidates for extent reduction.
- Data sets that are candidates for migration, in priority order:
  - Rolled-off GDGs
  - GDG with Early Migration specified
  - Data set size
  - Age

  The intent of the priority order is that small data sets will need to age more than large data sets to actually migrate from the volume.

From this description, primary space management does just about everything possible to fragment a volume. This comment is intended to provide justification for you to at least consider using the DFSMShsm exit ARCMVEXT.

3.14.6 Additional DFSMShsm housekeeping functions

Some of the additional housekeeping functions are:

- Control data set maintenance

  Periodically, you may want to consider increasing the MCDS, BCDS, and OCDS in size. This can happen as a result of data growth in your environment. Follow the procedures in the *DFSMShsm Implementation and Customization Guide*.

  Attention: DO NOT perform a REORG of any DFSMShsm control data set without increasing the space allocation. The reason for that is DFSMShsm will detect the recompressed control intervals, and resplit them as needed. This will result in degraded performance for 2-3 week after which it will have undone any perceived space savings.

- DEFRAG ML1 volumes

  Periodically (consider once per month) the ML1 volumes would probably benefit from a DFSMSdss DEFRAG. This activity should be performed outside of the secondary space management window and have almost no impact on your environment.

- EXPIREBV

  In a DFSMS managed environment there are management class parameters that can only be enforced by expressly looking for the situation. EXPIREBV is a DFSMShsm command supplied to handle the situation that the data set no longer exists and to set dates that allow the backups taken while the data set existed to be deleted. It takes a minimum of two executions of EXPIREBV command to remove a backup version. The first execution will set dates for expiration of backups (the data set is deleted), and the second execution actually performs the HBDELETE (if the expiration date has passed).

  Since this is a long running command in most customers shops (it processes the entire BCDS looking for deleted data sets on primary disk), we advise that EXPIREBV be executed once per month during a weekend near the 15th (away from any month end processing). This can dramatically reduce the resources (backup tapes and BCDS records) needed to support DFSMShsm.

- RECYCLE

  Recycle is the DFSMShsm method of reclaiming tape media capacity (and it is also a good answer to the problem of entropy). The only time RECYCLE cannot execute is during CDS backup. RECYCLE is considered by DFSMShsm to be a long running task and therefore will be limited to one command in execution per LPAR. The command has many options, and you can (for example) recycle the ML2 tape volumes separately from the backup volumes.

  The desired tape occupancy for input to recycle is specified by SETSYS RECYCLEPERCENTVALID parameter (the default is 20%). This value can also be overridden by the RECYCLE command at any time by specifying PERCENTVALID. Taking all DFSMShsm defaults, we will need a minimum of four tape drives to execute RECYCLE (default recycle tasks is 2). We can set the minimum to 2 drives. If the duplexing option is specified, then the minimum number of drives is 3. The input and output tape devices do not need to be the same. In other words, RECYCLE is a valid method to perform a tape conversion.

4

Converting to DFSMShsm

This chapter assumes that the DFSMShsm system has been activated and tested as described in Chapter 3, “DFSMShsm activation” on page 69.

Next, we need to review the different methodologies that can be used when converting to DFSMShsm from another product. Since there are many different environments, we try to present an overview of the main issues that need consideration during a product conversion. We use the information gained from the preparation defined in Chapter 2, “Preparing your environment” on page 13.

The following scenarios are discussed:

- **Moving data under DFSMShsm control:** This scenario involves moving data from the current pool or volume to a new storage group that will be managed by DFSMShsm.
- **Conversion to DFSMShsm:** In this scenario, data sets currently controlled by the other storage management product are managed by DFSMShsm by changing management class, storage group and other attributes and removed from processing by the original product.

We also examine management considerations which apply to data sets regardless of the method used to activate SMS. Finally, we present ways to convert the copies of data sets from the previous storage management product, differentiating between archive and backup copies, and elaborating on the possible exceptions that may be encountered.

4.1 Applying assessment information

Regardless of the method you use to convert to DFSMShsm, the information gathered in Chapter 2, “Preparing your environment” on page 13 will influence the way that you plan to implement. Either way, DFSMShsm has to be active and ready to take over the management of your data.

We document different ways of converting data to DFSMShsm. There are more scenarios you can find in the real world, but these represent the most common situations.

In the first scenario, 4.2, “Case 1: Moving data sets to SMS and DFSMShsm” on page 92, we discuss the following aspects:

- Modifying SMS ACS routines to direct allocation to the new output pools or storage groups
- Changes needed to the routines of the old allocation product or environment
- Disabling the current data management policies of the old storage management product
- Ensuring that the new environment is performing as expected and that previous service levels policies are still being met

In the second scenario, 4.3, “Case 2: Converting volumes to SMS and DFSMShsm” on page 93, we discuss the following aspects:

- Changing storage group attributes to activate DFSMShsm management for the data sets and volume pools
- Modifying or adding management class attributes in order to control the backup, migration and expiration attributes of data sets
- Activating volume level attributes for data sets that remained non-system-managed
- Deactivating storage management for data sets now under DFSMShsm control

At this stage you will have probably set up either some or all of your SMS constructs and attempted to achieve similar management policy levels that were in use by the previous product. As mentioned previously, this conversion also gives the opportunity to review current practices and see if they are still applicable.

Many policies were originally designed when disk space was expensive and data recovery was usually due to hardware failures. This often meant significant downtime due to an online application needing to be recovered, for example.

Over the years, the price of disk space has diminished significantly and technology has evolved in such a way that disk subsystems today allow data to be reconstructed without noticeable interruption. The advent of Redundant Array of Independent Disks (RAID) and fault-tolerant devices has shifted the emphasis away from the traditional storage management methods and more towards data recovery either due to logical error or for disaster recovery purposes. In addition, data replication now uses immediate or near-immediate methods to provide copies in remote locations, away from the primary data center.

This does not mean, however, that the need for a storage management product has diminished. Some of the methods described above are costly in their implementation and generally the cost rises in relation to data availability requirements. In the case of a site needing to re-construct their data center at an alternate location, it is critical to ensure that key applications are available as soon as possible.

This requirement does not apply to all data and this can be made available on alternative media, such as tape. The storage management product will help ensure that the disk environment is available for the immediate requirements and that this space management policy allows the cataloging of tape media as well. In addition, this tape data can be transported to a safe location and be available if it is required. This is achieved through vaulting policies.

4.1.1 Excluding data sets using system affinity

In 3.12, “Testing DFSMShsm” on page 80 we discussed the option to assign a system name to the DFSMShsm function performing the automatic backup, auto-migration and auto-dump tasks. It is important to remember that when SETSYS NODEBUG is set and the functions are released, DFSMShsm will attempt to perform automatic backup and space management functions based on the following criteria:

- The time parameters are set for management functions
- The following parameters are set to Yes in the storage group:
  - Auto migrate
  - Auto backup
  - Auto dump
- The ADDVOL command specifies AUTOBACKUP, AUTODUMP or AUTOMIGRATION options for non-SMS volumes.

This becomes important when the current storage management product is also using SMS management class and storage group attributes to make decisions regarding archive and backup. Setting all automatic parameters to NO may prevent the other product from correctly managing unconverted data. CA-Disk, CA-ASM2, and ABR, all allow a method of ignoring SMS storage group constructs during processing and allow one method of continuation with the old product.

The other method that can be used is to define *system-affinity* in the storage group. Using these attributes DFSMShsm will check eligibility to process a storage group on a system or group of systems. If the values for **Auto Dump**, **Auto Migrate** and **Auto Backup** are set to a non-existent system, then DFSMShsm will not attempt to process the volumes in the storage group. Using normal default processing, the other storage management products to do not exploit this field and will not be affected. Please check in your old environment to ensure that the backup and archive jobs are not selecting data sets based on these attributes.

4.2 Case 1: Moving data sets to SMS and DFSMShsm

We start at the point where the testing and activation of the product are completed. The DFSMShsm started task is active and automatic operations are disabled by setting SETSYS DEBUG mode ON. As discussed in Chapter 3, “DFSMShsm activation” on page 69, this will allow automatic functions to run based on the time parameters supplied in the ARCCMD00 member, but data movement is not performed. Command level functions, such as HBACKDS or HMIGRATE, are performed, if the function is released. For example, to release migration functions, issue the TSO command:

```
HSENDCMD RELEASE MIGRATION
```

The automatic functions run in a simulation mode that shows information on the actions that would have been performed in SETSYS NODEBUG mode.

4.2.1 Modifying the current allocation manager

If you are using an allocation manager, such as CA-Allocate, MAINVIEW SRM EasyPOOL from BMC, or Allocation Control Center from DTS Software, you must ensure that these products do not interfere with the allocation request. As discussed in 2.2.1, “Pooling” on page 15, these products are capable of re-directing SMS allocation requests. If you decide to use these products for allocations to system-managed volumes, ensure that they are managing the same way as the ACS routines would, that is by assigning the correct SMS attributes.

The recommended method is to ensure that the data set is not influenced by the other allocation product if a storage class is assigned by the ACS routines.

If you are using system pooling, such as UNIT=WORK, the ACS routines can be set up to include this in pool selection or to ignore this setting completely.

4.2.2 New allocations

After the ACS routines have been written, tested and activated, new allocations are directed to system-managed volumes according to the policies defined. These data sets will now be managed by DFSMShsm. How these data sets are processed depends on the criteria set in the storage group and management class. If data sets are to be managed automatically, these functions will also need to be activated in the ARCCMD00 member, if this has not already been done.

4.2.3 Moving data into the new storage group

DFSMSdss is used to move the data sets into the correct storage group, based on the rules set up in the ACS routines. The DFSMSdss logical COPY function is used to perform data movement.

- If data sets are cataloged outside of the standard catalog search order, the INCAT keyword can be used to search for these. STEPCAT and JOBCAT cannot be used.
- Specify one or more non-system-managed volumes in the OUTDD or OUTDYNAM name. If a data set does not match the ACS routines, it will be placed onto a non-system-managed disk volume.
- If a data set is determined to be system-managed, a storage class and management class will be assigned.
- Ensure that keywords BYPASSACS and NULLSTORCLAS are not specified.

Please refer to Example 5-1 on page 120 for an example of how to code the DFSMSdss control statements.

4.3 Case 2: Converting volumes to SMS and DFSMShsm

This scenario describes a shop with little or no SMS implemented, but which may have already established pooling conventions by using an allocation manager such as CA-Allocate, MAINVIEW SRM EasyPOOL from BMC, or Allocation Control Center from DTS Software. DFSMShsm is activated but is currently not managing the data.

In this case, you have reviewed the current pooling and data set placement requirements and are happy with the way these are implemented. In order to place the data sets under system management, you decide that running DFSMSdss CONVERTV is appropriate. The ACS routines are also written to assign the appropriate constructs to the disk data sets.

In "DFSMSdss CONVERTV" on page 46, we discussed the methods that are used to ensure that all data sets are eligible for conversion to a system-managed disk volume, including running DFSMSdss CONVERTV with the TEST keyword.

After this is successfully completed, the volume status will be CONVERTED, and data sets are now system-managed, with storage class and storage group assigned. A management class is not required, but is recommended. From now on, the data sets can be managed with DFSMShsm.

4.4 Case 3: Modifying storage groups for DFSMShsm

This example describes an installation which has a partial implementation of SMS. The following statements apply:

- ACS routines are used to assign a storage class and storage group to data sets.
- Management classes are not assigned.
- The current storage management product is managing the storage groups, but bypasses management class and storage group attributes.
- A default management class is not specified in the base configuration.
- Non-system-managed data is also processed, in some cases using the same retention criteria as system-managed data sets.

In this case, we recommend that system-managed data sets have a management class assigned. This requires that the assessment of the current storage management environment cover the retention periods and number of existing copies of individual data set versions. In the cases where multiple retention periods apply to data sets within the same group or application, a decision needs to be made on which is the best retention policy. For example, if an application has a backup retention period ranging from 185 days to 5 years, which is the one that should be applied? For further information on this subject, please refer to 2.8, "Additional considerations" on page 66.

In the case where no management class is assigned, DFSMShsm will query SMS for the default management class. If none is defined, then defaults are used as described in Table 2-2 on page 36.

4.4.1 Assigning a management class to data sets

After deciding which management class best suits the data sets in the storage group, the management class name is added either using the ISMF command ALTER command or by using the IDCAMS ALTER MGMTCLAS command. If there are many data sets that require a change, an IDCAMS DCOLLECT report can be used as input to a REXX routine to generate the commands to make the change.

In the case of a VSAM data set, you should ensure that the cluster name is used as input to the ALTER command. Since the DCOLLECT active data set record (Record type D) only reports on component names, such as INDEX or DATA, the cluster name can be retrieved from the VSAM association record (Record type A). It should also not process a VSAM alternate index.

Note: If you decide to remove the management class from a data set, you must use ISMF to perform this. IDCAMS does not allow the ALTER command to set to a non-existent management class. You can, however, use IDCAMS ALTER to change to a valid management class.

Since the current storage management product is bypassing the management class and storage group rules, it can still be used to process the data sets after a management class is assigned, and prior to switching to DFSMShsm processing.

4.4.2 Assigning a default management class

Since the installation is not currently using a management class for system-managed data sets, we recommend setting a default management class. This covers any data sets which may not have a management class assigned by the process described in 4.4.1, “Assigning a management class to data sets” on page 95, as well as new allocations which may be overlooked in the ACS routines. For more information on specifying a default management class to the SMS base configuration, please refer to “Default management class” on page 35.

4.4.3 Converting non-system-managed volumes to DFSMShsm

In order for a non-system-managed volume to be processed, an ADDVOL command is added to the ARCCMD00 member for each volume. This specifies the management rules that apply to the volume, such as these:

- Should data sets be automatically migrated?
- When should migration occur?
- Should automatic backup run against this volume?
- Will the backups go to disk or tape?

- Can data sets be auto-recalled to this volume?
- Should automatic dump run against this volume?
- If yes, which dump class should be used?
- Can interval migration process the volume and what thresholds will be used?

The data sets and volumes can also be processed using DFSMShsm commands, which runs outside of the normal backup and space management windows. We recommend the automatic functions as the preferred processing method, since these allow better balancing and more efficient use of DFSMShsm resources.

4.4.4 Modifying the current storage management product

Once the data set has been moved into a storage group which is managed by DFSMShsm, it is important to ensure that the data set is not processed by the original product. The ramifications could range from the data set not being selected for backup to being deleted in error, depending on the function being performed.

Incremental backup considerations

In DFSMShsm the automatic backup process normally backs up a data set if it has been modified. In order to make this decision, the change-indicator is checked in the VTOC. If the data set was previously backed up by the other product, then this will prevent the backup by DFSMShsm.

The need to check for this situation is not immediately apparent. If the other storage management product performs on a volume level, then the data sets may not be processed if the volume in the storage group is a new volume and is not specified in the original backup jobs. Other products allow data sets to be processed on a data set level by issuing a catalog locate to retrieve the data set for processing. This allows data sets to be processed regardless of their volume placement. In such cases, we recommend that the volumes within a storage group are excluded from processing, either using a volume or storage group pattern mask.

If it is necessary to process data sets in the storage group using both products, it is important to ensure that the other backup product does not reset the change-indicator. This will ensure that correct aging and versioning of the data set is maintained. Some of the other products offer an additional backup parameter such as NORESET or NOCHANGE to prevent the change. If the other product does not do this, we highly recommend that you remove the volumes in the storage group from management by the other product.

4.4.5 Backup considerations for data sets in use

Since your current storage management product may allow a backup of data sets in use, you probably want to achieve the same functionality. We refer to this as a "fuzzy backup" since it is possible that the data set was in the processed of being updated at the time the backup was taken. In this section we discuss a number of possibilities:

- Backup-while-open candidate data sets
- SETSYS BACKUP(INUSE)
- DFSMShsm exit ARCBDEXT
- ARCINBAK program

In the following sections we review each of the listed options.

Backup-while-open

For a moment, let us consider another form of legitimate in use backup. Some CICS and IMS data sets can be defined as backup-while-open (BWO) candidates. This provides a way of backing up data sets that are known to be allocated for long periods of time. Using DFSMSdss as the data mover, standard serialization is initially attempted. If this is successful, then the status is set to signify a normal backup. If serialization fails and the data set is a BWO candidate, the data set is processed with this status. If a restore is required at some stage, the data set will have a status to reflect that it was backed up while open.

The BWO attribute is added to a qualifying data set using the IDCAMS ALTER command. This is an example of how to add the attribute:

```
ALTER dsname BWO(TYPECICS)
```

SETSYS BACKUP(INUSE)

Another method that you can use is to specify the SETSYS BACKUP(INUSE) parameter and the following subparameters:

- RETRY(Y)
  DFSMShsm should try to back up a data set once if the data set is in use or a BWO candidate performs a CA/CI split and the backup is discarded.
- RETRY(N)
  DFSMShsm will not retry the backup if the first attempt fails due to the data set being in use.
- DELAY(*nnn*)
  DFSMShsm will wait for *nnn* number of minutes before retrying the backup for a data set in use. *nnn* is a value from 0 to 999.

- SERIALIZATION(PREFERRED)
  DFSMShsm will retry the backup after a delay and backs the data set up, even if it is still in use.
- SERIALIZATION(REQUIRED)
  DFSMShsm will retry the backup after some delay and will fail the backup if the data set is still found to be in use.

ARCBDEXT exit

An additional method is the implementation of the DFSMShsm ARCBDEXT exit, a sample of which can be found in SYS1.SAMPLIB. The sample exit shows how data sets can be excluded from processing by data set name, as well as showing how data sets assigned with a particular management class can have unique serialization attributes assigned.

One thing that should be noted when using SETSYS BACKUP(INUSE) is that the retry attempt is made as a BACKDS command. This means that the backup is taken according to the settings of SETSYS DSBACKUP. This allows command level backups to be directed to either disk or tape, optionally depending on data set size. This command was first introduced in DFSMShsm release 2.10. Prior to this release, command backups can only be output to disk ML1 devices. Using the ARCBDEXT exit will allow data sets selected for backup even if serialization fails to be backed up to tape.

ARCINBAK program

The fourth method that is available for data sets in use is inline backup which is invoked by executing the ARCINBAK program. This is useful in batch jobs where there is a requirement to back up data sets in the middle of a job to achieve a backup at a particular point in the processing cycle. It also allows a data set to be addressed as a reference to a previous DDNAME in the job or as a relative generation number of a generation data group.

Since these backups are designed to process data sets that are in use, the backup versions created are flagged as unserialized in the BCDS record, regardless of their in-use status at the time. For more information on this subject, please refer to *DFSMShsm Managing your own data* and also the *DFSMShsm Storage Administration Guide*.

4.4.6 Space management considerations

In this section, we discuss some considerations to take into account when trying to achieve the space management functionality you have in your current storage management product.

Migration/archive

Data sets that are now managed by DFSMShsm should also be excluded from the archive and space management process of the other storage management product. Since SMS is now managing the life-cycle of the system-managed data set, it is now the policy manager. Functions performed by the other storage management product may no longer apply and invalidate the new policy. The most visible policy change is in migration or archival. Regardless of the storage management product, an archived or migrated data set will usually have a volume serial name which does not match a real volume. This volume is also known as a “pseudo volume”.

Other storage management products provide different methods of co-existence with DFSMShsm for the interim period before conversion is fully completed. This allows a recall or restore to be directed to the correct storage management product. In some cases, however, the exact location of the migrated copy of the data set may not be apparent, since some storage management products allow use of the same pseudo-volser (MIGRAT) as DFSMShsm. Others, such as CA-Disk, are able to return a volume serial of MIGRAT to the requesting application, unless it has been specifically excluded from doing so. This means that further investigation may be required, either by the end-user or storage administrator to determine the exact location of the data set.

In general, if a data set list is viewed in ISPF using option 3.4 and the volser listed is MIGRAT1 or MIGRAT2, then the data set has been migrated using DFSMShsm. The catalog entry of the migrated data set contains a device type reflecting the migration level. If the device type is disk, then the list shows MIGRAT1; if the device type is tape, then the list displays MIGRAT2. In some cases, issuing the user command HRECALL allows the data set to be successfully recalled, irrespective of the product.

Space release and extent reduction

The management class allows partial space release to be specified and this setting is honored either at initial allocation time (during close) or by DFSMShsm during primary space management.

```
ARC0734I ACTION=PARTREL FRVOL=VPSI0A TOVOL= *** TRACKS=11
RC=0,REASON=0,AGE=1, DSN=TST.PROD.JCLLIB
```

In addition, extent reduction is also performed during primary space management. This is performed based on the parameter SETSYS MAXEXTENTS. This is performed against non-VSAM non-extended format data sets by migrating and recalling the data set. This appears in the space management activity log as:

```
ARC0734I ACTION=REDUCED FRVOL=VPWRKE TOVOL=VPHL1D TRACKS=10 RC=0,
REASON=0,AGE= 1, DSN=TST.INPUT.DATA
```

Other storage management products can process the data sets using different parameters and these may not match the policies defined in the management class and the parameters in DFSMShsm.

Fragmentation on primary disk volumes

It is possible to initiate defragmentation jobs using the ARCMVEXT exit in DFSMShsm. This exit is called during volume space management and is invoked for each primary disk volume, but not for ML1 or ML2 volumes. Volume space management occurs:

- During automatic primary space management
- During interval migration
- When a MIGRATE VOLUME command is issued

It is important to remember that a defragmentation run may not be desirable at all hours of the day, since this may well contend with existing processing. In this case, we recommend either that the exit runs generally in disabled mode and is activated using the SETSYS EXITON(MV) command or that the exit submits a job scheduled to run at a more convenient time.

A sample exit can be found in SYS1.SAMPLIB in member ARCMVEXT.

4.5 Converting archive to migrated data

This section deals with the activities that are required to convert the archive copies from your current storage management product to migrated versions in DFSMShsm.

4.5.1 DFSMShsm control data set backup considerations

The DFSMShsm starter set provided in SYS1.SAMPLIB member ARCSTRST, generates a sample backup job for the control data sets on disk volumes. This is found in ALLOCBK1 and allows for four backup versions to be written to disk.

The process of converting both the archive and backup versions from your current storage management product creates information in both the control data sets and these updates are also recorded in the journal. In addition, if you decide to run the FIXCDS commands to modify the records, these activities are also recorded in the journal. If the journal reaches capacity, a backup of the control data sets is required in order to reset it. This is achieved using the BACKVOL CDS command and is also performed as the first stage of automatic backup. During the CDS backup process, DFSMShsm CDS update processing is halted and resumes after the backup has completed.

For the duration of the conversion of the versions from your current storage management product we recommend that you create six additional groups of data sets in order to allow more backups of the control data sets. This allows for ten backup copies and allows for a greater fallback period in cases where the control data sets require multiple backups in one day.

4.5.2 Extracting archive versions data from your current product

In 2.7.1, “Finding out the number of archive versions” on page 63, we discussed the need to separate archive from backup versions of data sets. This is achieved by running a utility from your current storage management product to report on the contents of the control data set. The type of output provided by this report depends on the product. The following reports provide the archive entries:

- CA-ASM2: $RSVP command, using IPCUTYPE EQ ARCHIVE.
- CA-Disk: LISTD report.
  - Use DUPLICATES option to ensure all versions are listed, not just the latest.
  - This report can be customized using the FIELDS= parameter.
  - Specifying FLGSC as part of the FIELDS= parameter will show archive data sets with Y in the SCR column.
- ABR: FDREPORT using DATATYPE=ARCHIVE.
  If FDREPORT is not available use FDRABRP with RECALL=YES.

These reports can be customized and formatted to suit your requirements. The field types that are of interest are:

- **Data set name:** This is the data set name.
- **Source volume:** This volume is either used for the restore or can be used for volume selection decisions.
- **Tape archive volume:** Requests are sorted based on this field, to avoid duplicate tape mounts.
- **Archive date:** This field is used as input to generate FIXCDS commands to modify the MCD record in the MCDS to reflect the migration date of the converted data set.
- **Archive time:** This will be added to the previous FIXCDS command.
- **Expiration date:** This is used to decide whether a data set version should be converted to DFSMShsm.
- **Data set size:** Optional.

- **Version number:** If the version number is not the latest, then this version should either not be processed or should be restored as a backup version. In general, we find that older archive versions can be discarded.

4.5.3 Exceptions

The report process from your current storage management product should have provided a report of all the archived data set versions in the control file. These reports include data set exceptions that need attention prior to starting the conversion process.

Most of the exceptions listed refer to catalog exceptions. The reports can be used as input to a REXX process that can generate IDCAMS LIST CATALOG commands to report on current status.

An alternative, faster method is to use the Catalog Search Interface (CSI), which is a read-only general-use programming interface to obtain information about entries contained in ICF catalogs. Information on this interface can be found in *DFSMS Managing Catalogs*, including a sample REXX procedure and assembler programs that are customizable to suit your needs.

Using the data set name as the input parameter, the following conditions can be detected:

- **Data set is archived but also exists on primary disk:**

  In this case the report shows a data set that was archived at some stage, but now the user catalog shows the data set allocated on primary disk.

  Some of the storage management products do not automatically remove entries from the control file if the data set is restored. For example, a system parameter is required in order to allow the control data set maintenance utility to delete these entries, and if this parameter was not set at restore time, the data set is kept until the original expiration date, if not manually deleted.

  We recommend that these entries are not restored or converted to DFSMShsm.

- **Data set is archived, but no catalog entry exists:**

  This is a condition which needs close examination as this may have been the original intention of the user.

  The following scenarios could have led to this condition:

  – The data set was intentionally archived this way, by specifying a scratch and uncatalog at archive time.

    The data set is probably a candidate for migration. In order for DFSMShsm to migrate, it must be cataloged when restored.

– The data set at some stage was uncataloged, either using IDCAMS DELETE NOSCRATCH, HDELETE or UNCATALOG commands.

  It is necessary to examine the original intention of the uncatalog action. If the data set is no longer required, then there is no need to migrate. If the reason was to lessen the number of data set entries in ISPF option 3.4, then it may require migration. If this is the case, the data set must be cataloged when restored.

– The data set is a generation data set which was removed from the sphere.

  This topic is discussed later.

– Some other unknown process caused the data set to be uncataloged.

  The data set will require research to see if migration is required. If so, it must be cataloged when restored.

- **Data set listed as archived, but no alias exists:**

  A user may have left the organization, and the userid and alias were deleted. Data sets that were allocated with their high-level qualifier may still be needed and will therefore require conversion. If this is the case, then you should define the alias, otherwise all the data sets are cataloged in the master catalog.

- **Data set listed as archived, but now matches GDG base name:**

  This is not a common occurrence but can happen when a data set is uncataloged and at some time in the future a GDG base name is allocated using the same name as the data set. In this case, the restore of the data set will fail. If the data set is required, then it must be restored using a new name.

- **Data set listed as archive, but physical tape is lost or damaged:**

  In this case, you may not be able to restore the data on the physical tape. If a duplex copy exists, then this can be used. Otherwise the entries on this tape should be deleted from the current storage management product.

- **Archived generation data sets:**

  Depending on your policies, you may find generation data sets that were archived and subsequently scratched from the GDG sphere. If these data sets are restored for migration, they may affect the current active GDG sphere. We therefore recommend reviewing the requirement to convert these data sets to DFSMShsm. If they are required, we recommend that they are restored using a different name so that the active GDGs are not affected.

- **Data sets with more than one archive:**

  If a data set has been restored and the control data set information was not deleted at the time of the restore, there is the possibility that the data set will eventually be archived again. The older archive version is then not considered to be a valid entry. In such cases we recommend removing this older version from the control file and not restoring it.

- **Data sets with backup taken after archive version:**

  The scenario mentioned in the previous bullet discusses a data set where the archive version is restored but the control file entry is not deleted. In these cases it is also possible that the data set is modified at some stage and a subsequent backup version is made. In these cases, we recommend that if a data set requires conversion to DFSMShsm and it is no longer on primary disk, then the later backup version should be converted instead of the original archive version.

- **Unsupported data set formats:**

  You may have archived data sets in your control file which are not supported by DFSMShsm migration. These can include formats such as ISAM data sets, but can also include data sets which are not cataloged on primary disk. These data sets can be processed using backup commands by supplying a volume and unit parameter, but cannot be migrated unless cataloged.

4.5.4 Comparing control file and tape management system

In order for the conversion process to proceed as smoothly as possible, it is recommended that a comparison is run to ensure that the tape management system still controls the archive data sets. In some cases, we have found tapes that no longer contained the data from the storage management system. Unless a duplex copy exists for these volumes, the data is not retrievable.

Using the control file as the input a list of the tape volumes can be prepared and this list is compared to the tape management system list. Any differences requires further investigation.

4.5.5 Creating commands from the extract data

The extract step mentioned in 4.5.2, “Extracting archive versions data from your current product” on page 101, is the basis for generating commands to convert the data from archive format to DFSMShsm migrated data.

The need to restore the archived data onto primary disk cannot be avoided. The formats of your current storage management product and DFSMShsm are not compatible and the data must be restored into its original format for subsequent migration.

The flow of actions will be as follows:

- Generate restore commands from extracted data
- Restore data sets from tape
- Generate HMIGRATE commands from successful restore
- Optionally generate FIXCDS commands
- Run HMIGRATE commands
- Run FIXCDS commands if required
- Note exceptions

Figure 4-1 shows a graphical representation of the sequence of events used to convert the archived data sets from your current storage management product to DFSMShsm migrated versions.

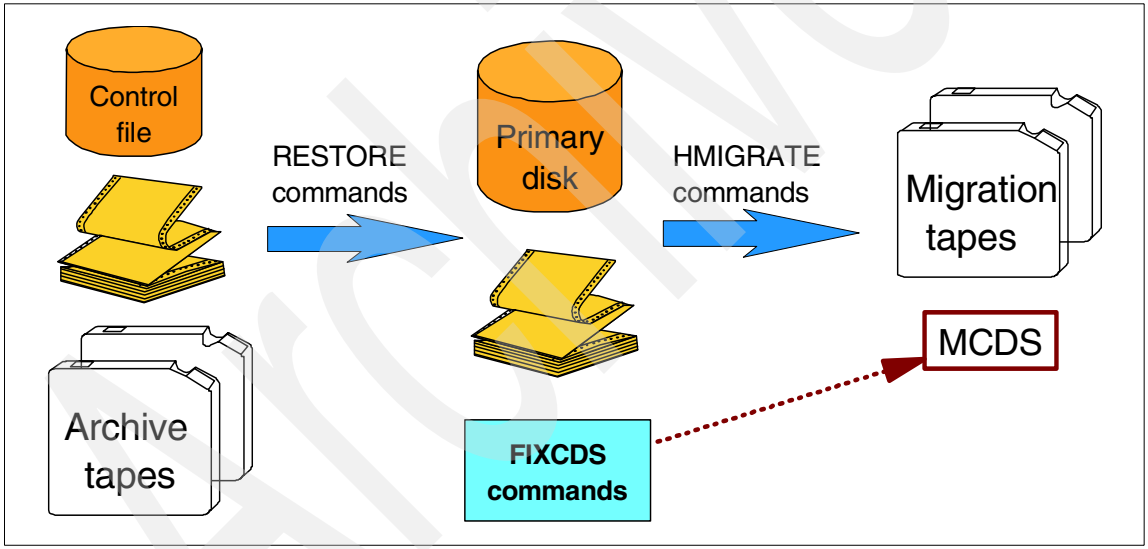


Figure 4-1 Converting archive versions to DFSMShsm migration entries

In the following sections we detail each of the actions.

Generate restore commands from extracted data

The commands to restore the data from your current product are taken from a list of the archived data sets. Depending on the product you are using the data set version information is stored in a control file separated from the backup versions or they may be stored in the same control file.

In general, a REXX program can be used to generate the restore commands from the list. In some cases your current product may offer another way to generate multiple restore commands. This allows generation of restore commands and to sort them based on the tape volume where the version is currently located. The objective is to mount an archive tape one time only and is preferable to trying to convert all data sets that resided on a specific source volume at one time. This will greatly speed up the whole restore process as well as allow you to control disk volume usage better.

Restore data sets from tape

The job to restore the data sets versions from the tape can be run concurrently or single-threaded. The method that you decide to use will depend on the number of tape drives that you have available as well as the amount of disk space available for the restores. If you are restoring to an existing storage group or pool then you will want to ensure that enough space is available for existing operations. If you have a pool set aside for the conversions, then parallel input processing is recommended.

If you are running in an environment which is not using tape robotics, then it is recommended to generate a tape pull list. This will ensure that the volumes are readily available when required and that the process is not delayed due to long mount times.

Generating HMIGRATE commands

When the data sets are available on primary disk there are two methods that can be used to migrate them to DFSMShsm.

The first method will migrate the data sets by generating explicit HMIGRATE commands for each data set. We recommend using the REXX language to do this. These commands are then submitted as a batch TSO job and are processed as single commands. For data sets migrating directly to level 2 we recommend that the NOWAIT parameter of HMIGRATE is used, to prevent the output tape from dismounting after each request. The command generated would be:

```
HMIGRATE 'dsname' ML2 NOWAIT
```

In our conversions we have found that some data sets archived with other storage management products are stored with the change-indicator set, meaning that they require a backup. When these data sets are restored they are placed on the primary disk with the same VTOC settings as at the time of archiving. The consequence is that DFSMShsm issues a ARC1280I message saying that the data set requires a backup before it can be migrated. In these cases there are a number of possible ways to remedy this condition:

- In the REXX procedure used to generate the HMIGRATE commands, use the LISTDSI function to query the setting of the change-indicator of a data set. The SYSUPDATED variable returns the value YES if the data set has this indicator set. An HBACKDS command can conditionally be issued against these data sets prior to invoking the HMIGRATE command.
- Use DFSMSdss to reset the change-indicator on all data sets prior to attempting migration. This can be done using the JCL in Example 4-1.

Example 4-1 Sample DFSMSdss job to reset change-indicator

```
//STEP1  EXEC  PGM=ADRDSSU,REGION=2048K
//*
//SYSPRINT  DD  SYSOUT=*
//OUTDD1 DD DUMMY
//SYSIN   DD  *
      DUMP OUTDD(OUTDD1) -
        LOGINDY(SBOX77) -
        DS(INCL(**) -
           BY((DSCHA EQ 1))) RESET
```

- Assign the data sets to a management class that does not require backups. This is done via the ACS routines and can be limited to processing just these data sets by filtering on a job name, for example.

Generating FIXCDS commands

Important: The FIXCDS command is a powerful diagnostic and modification tool used to make minor changes to the control data sets. It should be used with caution.

An additional consideration during this process is the preservation of data set aging in accordance with your management class policies. When the data set is migrated to DFSMShsm, the migration control data set data set record (MCD) contains the current time and date. These are found in MCDTMIG and MCDDMIG fields which are found at offsets x'20' and X'24' respectively. DFSMShsm offers the authorized user the ability to display, modify or create control data set records using the FIXCDS command. This is issued from a TSO user session or via batch TSO (PGM=IKJEFT01).

The REXX procedure use previously to generate the HMIGRATE commands can be altered to additionally create the FIXCDS commands. There are a number of things to note when doing this:

- The FIXCDS command batch job must be submitted with TYPRUN=HOLD in the JCL job card. This is to allow the HMIGRATE commands previously submitted to complete, since they were issued using the NOWAIT parameter.

- The FIXCDS output should be written to an output data set using the OUTDATASET parameter. This allows you to see the results of the commands.
- The FIXCDS results are also transmitted to the TSO userid associated with the batch job using the TPUT macro. This can produce a number of output messages, depending on the sum of the data sets that are processed. In this case it advisable to submit this job with a "dummy" userid that is not logged on to TSO. In this case the messages are discarded, but the results are still available in the output data set.
- Depending on the current storage management product it may be necessary to alter additional fields in the MCD record. This can occur if the data set creation date and reference date are reset to the current date during the restore process. If this is the case, then the MCDTLR(*last reference time*), MCDDLR(*last reference date*) and MCDDLC(*creation date*) fields can also be modified with FIXCDS.
- The date and time format may not match the format from the original storage management product. For example, the time field is able to store thousandths of seconds as well as hours, minutes and seconds. You should ensure that the date stamp matches the correct DFSMShsm format, which is obtained from the TIME DEC macro and is in format X'0cyyddds' where

c	Represents the century. For 1900 to 1999 c=0, 2000 onwards c=1.
yy	Represents the last 2 digits of the year.
ddd	Shows the day of the year.
s	Is a 4-bit sign character, usually 'F'.

For more information on use of the FIXCDS command, please refer to *DFSMShsm Diagnosis Guide* and the *DFSMShsm Diagnosis Reference* manuals.

Run HMIGRATE commands

The JCL stream and REXX process used to generate the HMIGRATE commands can also contain additional steps to submit the HMIGRATE and FIXCDS commands. As previously noted, the FIXCDS command should be submitted with TYPRUN=HOLD or run at a time when the migration commands have completed.

Using interval migration for archive conversion

An alternative method of migrating the data sets from primary disk with explicit commands is the use of interval migration. If the pools and storage groups are large enough to accommodate the volume of data that is restored, then the hourly interval migration cycle can also be used to convert the data set to

DFSMShsm format. The requirement to ensure that the data sets are backed up before migration as well as the need for the FIXCDS commands may still apply in this case. The REXX procedure can be modified to remove the HMIGRATE commands, but still perform the other requests.

Archive conversion exceptions

The conversion process involves many data sets and errors may occur that prevent the versions from converting successfully. In the early stages of the conversion process, it is better to review the errors as they occur to ensure that there is not a fundamental problem that requires immediate correction. Other problems may be, for example, space related and can be corrected after the majority of the data is converted.

Some of the errors that we have seen at this stage include:

- Restore is attempted to a system-managed volume, but the data set is not eligible.

  Action: Correct ACS routines or restore to a non-system-managed volume.

- Space related abends

  This can occur due to space miscalculation or fragmentation on volumes.

  Action: Either pre-allocate the data set or add disk volumes to storage group.

- Data set are already allocated on disk

  Action: Probably no need to convert the data set.

- Data set cannot be converted to DFSMShsm format

  Restored data set is an unsupported format, such as ISAM.

  Action: Group similar data sets together and write to tape as DFSMSdss physical copy.

4.6 Converting backup versions

As we previously mentioned in 2.7.2, “Finding out the number of backup versions” on page 64, a careful review is required to see which backup versions need to be converted into DFSMShsm backup versions. Under normal circumstances, the majority of the backup versions will expire from your current storage management product within a few weeks or months and it is probably not worth converting these copies. There may, however, be a requirement for certain data sets to be kept for longer periods of time and require conversion. We will describe one method that can be used to convert these data sets.

4.6.1 Extracting backup versions data from your current product

The initial step is similar to extracting the archive data set information in 4.5.2, “Extracting archive versions data from your current product” on page 101, except that now we are looking for backup versions. The reports used are:

- CA-ASM2: $RSVP command, using IPCUTYPE EQ BACKUP.
- CA-Disk: LISTD report.
  - Use DUPLICATE option to ensure all versions are listed.
  - This report can be customized using the FIELDS= parameter
  - Specifying FLGSC as part of the FIELDS= parameter will show backup data sets with ’ ’ in the SCR column, as opposed to Y for archive versions.
- ABR: FDREPORT using OLDBACKUP=ALL on the PRINT statement.

  If FDREPORT is not available use FDRABRP with PRINT BACKUP and OLDBACKUP=ALL.

These reports can be customized and formatted to suit your requirements. The field types that are of interest are:

- **Data set name:** This is the data set name.
- **Source volume:** This volume is either used for the restore or can be used for volume selection decisions.
- **Tape backup volume:** Requests are sorted based on this field, to avoid duplicate tape mounts.
- **Archive date:** This field is used as input to generate FIXCDS commands to modify the MCB and MCC records in the BCDS to reflect the backup date of the converted data set.
- **Backup time:** This will be added to the previous FIXCDS command
- **Expiration date:** This is used to decide whether a data set version should be converted to DFSMShsm
- **Data set size:** Optional.
- **Version number:** This is used to assign a version number to the data set version during the restore process. In the described method the data set will also be backed up to DFSMShsm using the version number.

4.6.2 Exceptions

As in the archive conversions, there are exceptions in the backup category which require attention prior to the conversion progress:

- Number of backup versions greater than 100:

  In some cases you may find that a particular data set has more than 100 backup copies stored in control file of your current storage management product. The maximum number of versions that can be stored in the DFSMShsm BCDS is 100.

  With the method that we are describing to convert the backup versions, this is not a problem since the copies will be restored each with a unique data set name. The format after restore with a new name is:

  ```
  DSNAME.V1
  ```

  If you require that the data set is stored in the BCDS with the original data set name, then the older copies greater than 100 will need to be processed outside of availability management. The alternatives could be using DFSMSdss or as an aggregate backup of DFSMShsm.

- Unsupported data set formats:

  As with archived data sets, there may be backup versions in your control files which are not supported formats in DFSMShsm, such as ISAM data sets. It is best that these data sets are not converted. If there is a requirement to retain the data sets, then DFSMSdss can be used to process a physical copy of the data.

4.6.3 Creating commands from backup extract data

There are a number of differences in the work flow used to generate the commands for restoring the backup versions of data sets. The major difference between the two approaches covers the fact that multiple versions of the same data set name exist and in order to process these data sets, they need to be restored using unique data set names. This is for two reasons:

- In most cases the data set already exists on primary disk
- Using a unique data set name allows multiple versions of the same data set to be restored and these can be streamed into DFSMShsm.

The action flow is as follows:

- Generate restore command from extracted data
- Restore data sets with new name from tape
- Generate HBACKDS commands from successful restore
- Optionally generate FIXCDS commands
- Run HBACKDS commands
- Run FIXCDS commands if required
- Note exceptions

The sequence of events for converting the backup copies from your current storage management product to DFSMShsm backup versions is represented in Figure 4-2.

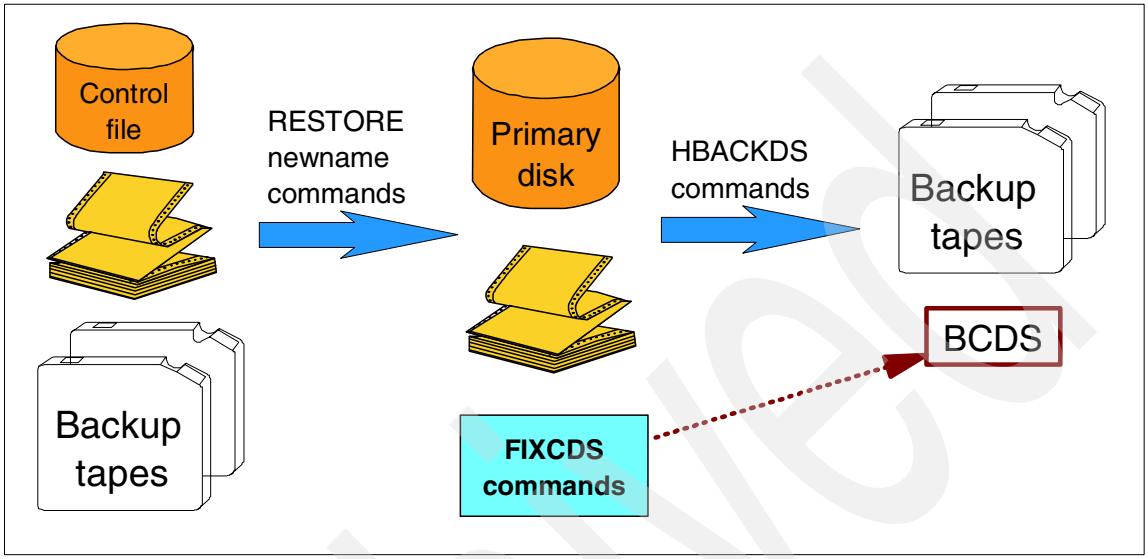


Figure 4-2 Converting backup versions to DFSMShsm backup copies

Generating commands from backup extract data

As with the archive data sets, the commands generated are taken from the listing produced in 4.6.1, “Extracting backup versions data from your current product” on page 110.

A REXX procedure in this case can also be used to create the restore commands for the data sets. The restore command is somewhat different to the one used for the archive data sets, since versioning now must be performed. Here are some ways that this might be achieved with the products:

- In ABR, use the ADATE parameter on the SELECT statement to restore a version from a specific date. If multiple backups were taken in a day, then the OLDBACKUP parameter is needed for versions other than the last one on a particular date. These statements are in addition to the NEWNAME parameter.
- In CA-ASM2, the $RB command is used with NEWNAME. UDATE and UTIME are used to specify the unload date and time.
- In CA-Disk the RESTORE command is generated with a NEWNAME and TIMEDATE parameter.

Tip: It is preferable to use a date and time parameter rather than a version number whenever possible. This avoids version expiration problems if the commands are generated some days before the restore jobs are run.

As previously mentioned in the archive data conversion, it is recommended to sort the restore commands using the tape volume serial number as the sort field. This will require that backup tape is mounted only once, making the restore process more efficient.

Generating HBACKDS commands

After the data sets are successfully restored to the volumes, the DFSMShsm HBACKDS commands can be generated. This can be done using a restore report showing the data sets that were restored without error.

The method used to back up the data sets from the disk volume depends on the release of z/OS or OS/390 that you are running. DFSMShsm releases prior to OS/390 2.10 only allowed HBACKDS commands to output to disk volumes. In this case the generated command looks like this:

```
HBACKDS 'dsname.v1' WAIT
```

In this case, we have specified the WAIT command in order to serialize the processing. This is useful if a subsequent command, such as DELETE, is needed.

If your release of DFSMShsm is z/OS or OS/390 2.10, then the following command is usable to back the data set directly to tape:

```
HBACKDS 'dsname.v1' TARGET(TAPE) NOWAIT
```

We have specified NOWAIT in this case, as the WAIT parameter can cause the tape to rewind and unload for each HBACKDS request. OS/390 introduced an alternative method if you do require the WAIT parameter. The following command can be coded to prevent DFSMShsm from demounting a tape used for command level backups:

```
SETSYS DSBACKUP (TAPE(TASKS(2) DEMOUNTDELAY(MINUTES(60))))
```

This command specifies that a maximum of 2 tape tasks can run concurrently and that the tape associated with the task remains mounted for 60 minutes, waiting to process requests made using the WAIT parameter. This processing can also be utilized to allow setting a return code from backup processing. This is useful in order to receive an extended set of return and reason codes from DFSMShsm interactive processing. In this case the command is set up like this:

```
HBACKDS 'dsname.v1' TARGET(TAPE) WAIT EXTENDRC
```

The return and reason codes are then used to debug backup failures.

Generating FIXCDS commands

Note: The FIXCDS command is a powerful diagnostic and modification tool used to make minor changes to the control data sets. It should be used with caution.

As with the migration phase, the generation of FIXCDS commands is an optional step. The benefit of this is that the data set version can be expired in accordance with management class attributes and still honor the original backup date as specified in your current storage management product. In this case there are two records in the BCDS which need modification, the BCDS data set record (MCB) and the BCDS backup version record (MCC). The fields to be modified are:

- MCBTBU (offset x'10') and MCBDBU (offset x'14') on the MCB record.
- MCCTSBUT (offset x'38') and MCCTSBUD (offset x'3C') in the MCC record.

For each data set backed up there is one MCB record, but there can be multiple MCC records, since these contain information regarding the individual backup versions. There are two ways to retrieve this record using FIXCDS. We recommend that you add the GEN(xx) parameter, which allows the version to be addressed as a backup generation, otherwise you need to know the name generated by DFSMShsm.

If you are using the HBACKDS command with the WAIT parameter, the FIXCDS command can be added to run immediately, specifying GEN(0) as the version. Since the data sets were restored using a new name, there should only be one version of the data set.

Please refer to "Generating FIXCDS commands" on page 107 for additional information regarding date formats and other considerations when using the FIXCDS parameter.

Running HBACKDS commands

The REXX JCL to generate and run the HBACKDS process can also be used the generate the FIXCDS commands.

HBACKDS to disk

In order to expedite the processing you may wish to run the HBACKDS commands to ML1 disk. As the output disks fill up it may be necessary to force the backup versions to be written to tape. In this case, the command to initiate movement of the backup versions to daily backup volumes is:

```
FREEVOL ML1BACKUPVERSION
```

These data sets are considered part of spill processing and are directed based on the SETSYS SPILL parameter.

Automatic backup considerations

You may consider using automatic backup to process the data set versions after they are restored onto primary disk. In this case, the change-indicator must be considered.

Depending on your current storage management product, the restore process may leave the change-indicator not set. In the case of a system-managed data set, the storage group is queried in order to determine backup eligibility based on guaranteed backup frequency, since the data set is not considered changed. For non-system-managed data sets, the setting of SETSYS INCREMENTALBACKUP is significant. The normal recommended setting of this parameter is CHANGEDONLY, which means that the data sets are not processed. You may want to consider setting this parameter to ORIGINAL for this run. This allows non-system-managed data sets to be processed regardless of the change-indicator.

Backup conversion exceptions

The exceptions generated during the backup conversion process are similar to those from the archive conversion phase. Please refer to “Archive conversion exceptions” on page 109 for more information.

4.7 Conversion cleanup

After the conversion of data is completed, a review of the old storage management environment is required in order to resolve any outstanding actions.These include:

- Running control file maintenance in order to remove data set versions and return tapes to the tape management system.
  - Depending on expiration dates set in the old storage management products, data set versions may have to be manually selected and expired.
  - If any data sets versions remain, merge these to new tapes so that the total number of in-use tapes is minimized.
- Reviewing catalogs and seeing if any data set entries are still cataloged to the pseudo-volume of the old storage management product.
  - If these data sets failed during conversion, establish the reason for failure and attempt to re-convert. If this does not work, find out if they are required. If not, delete the catalog entry.

- Reviewing data set versions that were not converted and informing the owners of this issue.

4.7.1 Review ISPF configuration table

Using TSO/ISPF PDF option 3.4 to list data sets, you are given the option of deleting individual data sets using the **D**(elete) command.

If this command is used against a migrated data set, an alternative panel is presented. Figure 4-3 shows an example of this panel.

```
------------------------------------------------------------------------
                        Confirm Migrate Delete
Command ===>

Data Set Name . : YCJRES3.SC64.SPFTEMP1.CNTL
Volume  . . . . :


The above volume indicates that the data set has been migrated.
It will therefore be deleted using the command: HDELETE

Enter "/" to select option
   Set data set delete confirmation off

Instructions:
  Press ENTER to confirm delete.
  Press CANCEL or EXIT to cancel delete.
```

Figure 4-3 ISPF panel ISRUADC5 when deleting a migrated data set

In Figure 4-3 the ISPF configuration tables have been left at default values, that is, the migration pseudo-volser is set to MIGRAT and the command HDELETE is issued to delete the data set. This allows a data set to be deleted without an automatic recall taking place.

It is possible that a modification may have been made at your site to cater for a different pseudo-volser, such as *ARCIVE* in CA-Disk. If this is the case, we recommend setting the values back to default. The publication *ISPF Planning and Customizing* describes how to build the ISPF configuration table. There are two keywords that may be modified:

- DELCMD: command to delete migrated data sets.
- DELVOL: volume of migrated data sets

4.8 Uninstalling the old storage management product

Once you have successfully converted all of the data set versions to DFSMShsm, there may be a requirement to uninstall the old storage management product.

Since these products often install hooks into the operating system, a removal should be approached with extreme caution. Some products modify open processing and an incorrect removal of a key component could prevent the operating system from initializing correctly. We therefore suggest a careful review before removing libraries defined in SYS1.PARMLIB, for example. It may be safer to wait until the next operating system upgrade to completely remove these components.

Hooks that are installed dynamically, such as auto-recall or auto-restore hooks, can often be uninstalled using a program. Once the requirement no longer exists for auto-recalls, these can be removed.

5

Exploitation

This chapter discusses topics that take the conversion project past the replacement of the storage management product. These activities could be incorporated into the initial project plan, deferred until after conversion or deferred indefinitely.

These topics are not presented at such a level that project plans can be immediately extracted from this text. The intent is to provide you reasons to continue the expansion of the storage management enhancements beyond replacement of another storage management product.

5.1 DFSMSdss as exclusive backup/restore utility

Prior to your conversion to DFSMShsm, DFSMSdss may or may not have been licensed in your environment. After conversion; DFSMSdss is a required prerequisite to using DFSMShsm. When DFSMShsm needs to backup or migrate certain data set types, it will call DFSMSdss through a documented interface to perform this service. The list of supported data set types for DFSMSdss is provided in the *DFSMShsm Implementation and Customization Guide*.

Initially there are situations that will require substitution of DFSMSdss for the replaced product. Example 5-1 shows a good first example, such as the task to move all the data sets with the high level qualifier of IBMUSER to volume DISKXX.

Example 5-1 Data set movement example

```
DFSMSdss:

//MYJOB  JOB ....
//MOVE  EXEC  PGM=ADRDSSU,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 COPY DATASET  (INCLUDE(IBMUSER.**))  -
      CATALOG SPHERE DELETE     -
      OUTDYNAM(DISKXX)          -
      ALLDATA(*) ALLEXCP ADMIN
/*

FDR:

//MYJOB  JOB ....
//MOVE     EXEC PGM=FDRCOPY,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSPRIN1 DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 MOVE     TYPE=DSF
 SELECT   CATDSN=IBMUSER.**,NVOL=DISKXX
/*

CA-Disk:

//MYJOB  JOB ....
//MOVE     EXEC  DMS,S='*'
//SYSIN  DD  *
 FIND DSN=IBMUSER./
 MOVE TOVOL=(DISKXX)
/*
```

There are many other situations in which DFSMSdss would be a viable and possibly superior alternative to an competitive product that is not replaced specifically as the result of this conversion. Competitive products would include:

- Compactor
- FDR
- FDRDSF
- FDRCOPY
- CA-Faver

DFSMSdss functions such as DEFRAG/COMPRESS/RELEASE, logical dump, and logical copy are the best suited for replacing these utilities. As you begin exploiting DFSMS extended data types, there is no additional activity of researching support of backup/restore of these data types if DFSMSdss is used.

5.2 ABARS

ABARS stands for Aggregate Backup and Recovery Support, and is a component of DFSMShsm and primarily associated with the task of application disaster recovery. If you have a DFSMShsm license, you have ABARS. One unique aspect of ABARS is that any cataloged data set could be subject to a backup (no matter where it resides in the storage hierarchy). ABARS also has the ability to not only copy, but also allocate data and support backups taken independently of ABARS (accompany).

A typical approach to disaster recovery is to backup all the volumes via full volume dump (on a Sunday with the system quiesced) and place these backups offsite. As you site grows and requirements for full availability of the system include the scheduled Sunday dump time, then this approach becomes impractical. If you still try to use full volume dumps without the system being quiesced, then you sacrifice the synchronization of the dumps across the volumes. This can hurt you when you restore as you find uncataloged data sets, phantom catalog pointers, and partially updated multi-volume data sets. Site growth can also effect you by requiring the dump window expand to accommodate the number of disk volumes involved and/or require the acquisition of additional tape drives to accommodate the bandwidth requirement of the dump window (and otherwise remain idle).

The ABARS approach is to backup only the critical data sets needed by an application in conjunction with the application cycle. The intent is to reduce the amount of data needed for application recovery and to move the backup window from Sunday to when a backup makes sense for the application. The objective of any recovery plan is not necessarily to restore everything, but to support the restart of required business applications in a “known state”.

This restart may require the re-run of some portion of the application cycle to bring the application current. As long as we have all the data sets (synchronized) needed by the application to be used as input or update, this restart (or re-run) is possible. This capability is intended to reduce the amount of data needed to provide disaster recovery and to change the backup window from a planned system outage to staggered application backup windows. This allows higher system availability and makes better use of available tape resources.

A by-product of the ABARS approach to disaster recovery is that the application recovery can be prioritized, so data for the most important applications can be restored first, and these applications started, while other applications wait their turn. ABARS changes the complexion of the recovery from volume recovery (where all applications must wait until after the last volume is restored) to business recovery.

Another by-product is that the aggregate can be used to distribute the application (as in a test environment). We have customers that uses the ARECOVER command with rename to refresh their test environments.

5.3 Extended Format VSAM

IBM successfully replaced usage of IAM (Innovation Access Method) with Extended Addressing Compressed VSAM. The initial customer requirements were that it should:

- Exceed the 4 GB limit
- Not require a backup and restore (re-organization) of the file to enlarge it (to add volumes)
- Not need additional licenses for new LPARS they were planning

There are some requirements for DFSMS compression to be a good fit for any application use. The Hardware Assisted Data Compression (HADC) incorporates the LZH compression algorithm. LZH is optimized to read activity, so it's more efficient if the data set access has a high read/write ratio. The effects of HADC are also less noticeable if the dominate access is random rather than sequential. For more information on compression see the redbook *DFSMS/MVS Version 1 Release 2,0 Compression, GG24-4056.*

5.3.1 Partitioned Data Set Extended (PDSE)

The business case for converting to using PDSE is easy. This is a replacement data structure for partitioned data sets that you do not need to compress.

PDSEs externally work like PDSs. The creation of a PDSE is directed by specifying DSNTYPE=LIBRARY on the allocation. In most cases, there are no application changes necessary to use the new access method (In one case that we know of, the application was reading the PDS with EXCP and expecting the data structure to be PDS).

For example, with PDSE you can:

- Manage multiple files as a unit, viewing the member list.
- Create new PDS from JCL.
- Access members as sequential files, including concatenation.
- Concatenate libraries (this implies a union of directories).
- Pack multiple members on disk track.
- Create member without overhead and serialization of new data set.
- BPAM, BSAM, & QSAM requests mapped to PDSE.
- Emulate directory read with BSAM/QSAM (Fixed 256, PS, KEY=8).

PDSE was created as a result of an IBM User Group writing a paper titled "PDS PAIN". The problems with PDSs eliminated by PDSE include these:

- Periodic compress not needed: Maintains list of reclaimable space
- Fixed size directory eliminated (a source of high connect time): Indexed based entry
- Need to re-write entire directory when adding members
- Cross system sharing of members and data sets
- Directory cannot be overwritten
- Change in BLKSIZE making older members unusable
- Storage efficiency improved in many cases (although small members can take up more space)
- 16 extent restriction (PDSE allows 123 extents)
- Need to delete aliases when a member is deleted: Alias deletes when member is deleted

And finally, what then is a PDSE?

- A single volume data set up to 123 extents: A 4K block non-VSAM SMS-managed data set
- Can be non-SMS (check informational APAR II12221)
- Pages of a member may be non-contiguous
- Maximum member size 15,728,638 (15*2**20-1) logical records
- Directory size is expandable: Directory size can contain 524,286 members (2**19-2)
- TTRs are device-independent
- Contains either data or program objects

It is possible to make PDSEs the default in your environment through either the SMS startup PARMLIB member or by assigning an SMS data class to the allocation.

5.4 Control of out-of-space conditions

When a data set is allocated for the first time, it occupies a requested amount of disk space for the life of the data set. There are limits placed on the size of each data set based on the information supplied on this request. There are also limits placed on the size of a disk volume and the number of disk volumes available to this allocation. If any of these limits are exceeded, an out-of-space condition is raised and the program writing data to this data set is abnormally terminated.

Examples of other storage management products that attempt to address this problem are MAINVIEW SRM StopX37/II from BMC, and SPACE RECOVERY SYSTEM from DTS Software.

SMS attempts, through improved allocation algorithms, to prevent allocation failure for new data sets and when extending multi-volume allocation using space constraint relief. SMS cannot prevent all out-of-space abends (no product can), but it can make them less likely to happen. The best defense for this type of problem is current and accurate estimates of space needed for every data set allocated in your environment. Beyond accurate allocation, this is a list of items to be checked if your installation wants the highest implementation possible to prevent out-of-space conditions.

- High allocation thresholds for all SMS storage groups:

  100% allocation of a volume is usually not possible if more than one data set exists on the volume. The intent here is to use a high allocation threshold (of 90% for example) to stop any new data set allocation to the volume when the threshold is exceeded and allow existing data sets on the volume to get secondary extents from the remaining space.

- Use of SPILL storage group:

  You can have a set of volumes available that will only be selected if SMS sees that this next new allocation will cause all enabled volumes in the selected storage group to exceed their high threshold. Allocations can now complete to the SPILL storage group until you have a chance to make space available in the primary storage group. One advantage of using SPILL storage groups is that allocations from any storage group can go to SPILL. One disadvantage is that only new data sets and not the extension of existing data sets are eligible for allocation in the SPILL storage group.

- Use of SPILL volumes within a storage group:

  If the space requirements some data sets is extremely volatile, you can reserve entire volumes in a storage group. By placing some volumes in a SMS storage group in a QUINEW status, these volumes will be available for new allocation and new volume extents for existing data sets within the storage group that they are defined,

- Use of **`ALTER entryname ADDVOLUME(*)`** to allow multi-volume allocation or extend existing multi-volume:

  SMS allows existing data sets to expand to multi-volume (VSAM and non-VSAM). SMS also allows to extend to any volume in the storage group with an '*' designation. For this ALTER to take effect, the altered data set must be closed (if it is open) and opened. Multiple volumes can also be specified in a single ALTER command.

- SMS space constraint relief:

  Introduced in DFSMS 1.4 was the ability to retry an allocation with a percent reduction on the requested primary allocation (should the initial allocation attempt fail). This applies to extended format VSAM data sets and allows for up to 256 extents per volume (and a primary allocation in 128 extents as opposed to the usual 5 extents). This value applies to new allocations and existing allocations when extending to an additional volume.

- SMS additional volume amount:

  SMS can allow you to specify when VSAM extends to additional volumes that it do so only with the secondary space specified (and not start with primary specification). This may allow you to allocate the data set with the intended space but direct expansion to occur in smaller increments when needed. The requirements to specify secondary space amount is that the data set be allocated as extended VSAM, RECORG is specified in the DATACLAS and that the volume count be greater than 1.

- System Determined Blocksize (SDB):

  This allows an optimum blocksize to be used for the data set on what ever device type the allocation resides. Typically it is half track blocking and this will improve the device allocation efficiency and therefore provide more room for other data sets. Works with non-VSAM data sets with DSORG = PS or PO and record formats of Fixed or Variable (will not work with RECFM=U).

  This recommendation has a beneficial and possibly detrimental side effects. The good news is that SDB will also make jobs run faster and use fewer I/Os than before. This is always true if the value calculated for SBD is larger than the present blocksize. Many of data sets in customer environments still have blocksizes like 3120, 6160, 13030, or 19069. These are not terrible, but SDB is better. The detrimental side effects are that with SDB there will be more

storage used for buffers (usually not a problem unless the REGION= parameter is too accurate in the JCL) and there can be no blocksize dependencies in your applications (specification of COBOL File Definition of 'BLOCK CONTAINS ZERO RECORDS').

- SMS extended format compression:

  This can improve allocation efficiencies more than compression can provide. The discussion in 5.3, "Extended Format VSAM" on page 122 provides some caveats to the applicability of this data structure, but not only is there space savings due to compression, but the data set containing the compressed records is also allocated at half track blocking (this has no connection to the BLKSIZE specified for the data set).

- Creation of LARGE and SMALL storage groups via SMS:

  You can implement data-set-size-based storage groups to help you deal with free-space fragmentation, and reduce or eliminate the need to perform DFSMSdss DEFRAG processing. Customers often use DEFRAG to reclaim free space in large enough chunks on each volume to prevent abends due to space constraints.By implementing data-set-size-based storage groups, one storage group, for example, can contain data sets smaller than 25 MB. With this approach, when a data set is deleted or expired, it leaves behind a chunk of free space that is similar in size to the next data set to be allocated. Since large data sets are not directed to this storage group, they are directed to other groups that might have less overall space, but in larger contiguous chunks. The end result is that the fragmentation index is high, but since space constraint abends do not occur, DEFRAG processing is not required.

- DFSMShsm extent reduction - SETSYS MAXEXTENTS(x):

  During primary space management, DFSMShsm will migrate/recall data sets with the specified number of extents to re-allocate the primary allocation to better suit the present data set requirements and allow the data set to grow with secondary allocation from here. Any value > 5 for MAXEXTENTS is recommended. Specify a value of 0 and the function will not be performed. The MAXEXTENTS parameter does not apply to VSAM data sets or to data set organizations not supported by DFSMShsm. You use this parameter for active, non-VSAM data sets. This parameter only applies to those non-VSAM data sets that are architecturally limited to 16 extents. It does not apply to multivolume data sets, striped data sets, extended format data sets, PDSE or HFS data sets.

- DFSMSdss DEFRAG or use DFSMShsm exit ARCMVEXT:

  Periodic use of DEFRAG is necessary in most customers' environments to maximize the available contiguous freespace.The source of fragmentation is using the volume via data set allocation, deletion, and release of unused space. Volumes constantly used in this manner tend to be the most

fragmented (with the exception of temporary space volumes). If your environment is shared disk, then you should either be varying the volume to be defragmented offline to all but the LPAR performing the DEFRAG or have implemented GRS or equivalent product to provide cross system serialization of data set allocation (refer to *DFSMSdss Storage Administration Guide* for more detail). The DFSMShsm exit offers the ability to automate the DEFRAG process (based on fragmentation index, or whatever criteria you wish to support) for volumes defined in a SMS storage group with either the AM=I or AM=P attribute.

- DFSMShsm interval migration:

  This is an automatic function in DFSMShsm to examine the SMS storage groups with the attribute (AM=I) on an hourly basis and perform migration to meet the low threshold when volume occupancy is greater than half way between the low and high threshold. By monitoring SMS storage groups and keeping them from filling, we will help prevent allocation failures from occurring.

5.5 Backup activity log analysis

Data set backup is usually the first system access of a newly allocated data set. This is also the first opportunity for DFSMShsm to report any error that might exist with this data set allocation. In an attempt to take advantage of this situation (and stop small problems from getting bigger), we have created a small utility called BUAL. This utility is provided with this book as additional material. For information on how to get the additional material, refer to Appendix C, “Additional material” on page 149.

BUAL performs DFSMShsm Backup Activity Log Analysis/Reporting and (if allowed) will correct errors as they are encountered.

The utility requires that SETSYS ACTLOGTYPE(DASD) be specified. DFSMShsm will then allocate activity logs on disk for recording the results of DFSMShsm activity. The DFSMShsm backup activity log will be called ’HSMACT.Hhostid.BAKLOG.Date.Time’. Activity logs are many small (usually less than 10 track) data sets that need to have a quick retirement (about 3 days is usually long enough).

The intent of BUAL is to be executed after backup is complete and report (and possibly repair) any errors. The only required change to BUAL is the specification of the HOSTid in which auto backup is scheduled to start. BUAL produces a synopsis of the backup performed, a summary of the errors, and a detail report of the errors found.

Example 5-2 shows a sample BUAL synopsis report.

Example 5-2 BUAL synopsis

```
LISTC level('HSMLOG.HA.BAKLOG') name
MIGRATED DATASETS:     0 DATASETS WITH       0 TRACKS
 PRIMARY DATASETS:     0 DATASETS WITH       0 TRACKS

 Backup started at 2001/11/28 20:00:02 and ended at 20:51:54
READY
```

Example 5-3 shows a sample BUAL summary report.

Example 5-3 BUAL summary

```
ERROR SUMMARY OF ERRORS IN DFSMS/HSM AUTOMATIC BACKUPS   2001/11/28
RC RC  TEXT
 2  36   LOOKUP ARC13XX MESSAGE  2  36                      1  DATASETS
16   6   LOOKUP ARC13XX MESSAGE 16   6                      1  DATASETS
16  12   LOOKUP ARC13XX MESSAGE 16  12                     10  DATASETS
19   0   DATASET IN USE (NON-VSAM)                         13  DATASETS
19  12   DATASET IN USE (VSAM)                             92  DATASETS
51   0   LOOKUP ARC13XX MESSAGE 51   0                      6  DATASETS
68 412   DATASET SERIALIZATION FAILURE                    254  DATASETS
99   4   UNSUPPORTED DSORG                                  9  DATASETS
99  50   LOOKUP ARC13XX MESSAGE 99  50                      5  DATASETS
```

Example 5-4 shows a BUAL detail report.

Example 5-4 BUAL detail

```
ERROR CATEGORIZATION OF DFSMS/HSM AUTOMATIC BACKUPS     2001/11/28
RETURN CODE REASON CODE  DSN
          >>>>>>>>>>>>>>> LOOKUP ARC13XX MESSAGE  2  36
          2         36  SMP.OTABLES
                            1 DATASETS
           >>>>>>>>>>>>>>> LOOKUP ARC13XX MESSAGE 16   6
         16           6  NFSCLNT.SC59.LOG1
                            1 DATASETS
           >>>>>>>>>>>>>>> LOOKUP ARC13XX MESSAGE 16  12
         16          12  OMVS.MQV1BRK.HFS
                         OMVS.SC53.MQ520.HFS
```

A

Sample questionnaire

This appendix contains a sample questionnaire which you can use to gather information about your environment.

Current storage management environment

1. What release and level of CA-Disk/CA-ASM2/FDR/ABR are you currently running?

   _____

2. How many systems/LPARs currently share each control data set?

   Please explain! (2 LPARS, 1 control data set)

   _____

3. Are all systems sharing the files data set also sharing catalogs?

   Please explain! (2 LPARS, 1 control data set sharing 1 catalog)

   _____

4. Do you currently utilize GRS or another resource serialization product? Yes / No

   Please explain! (GRS or CA-MiM across 3 systems)

   _____

5. What size is your control data set? Cylinders_____ Dev.Type _____
6. What device type will your DFSMShsm control data sets reside on?

   Device Type:_____ Model:_____ (3380, 3390)

7. Are you currently using an ATL, Silo or other robotic interface? Yes / No

   If YES, please explain! (IBM 3495 w/BTLS)

   _____

   _____

8. Is DFSMShsm currently active? Is it being used?

   _____

9. Will all systems run as one HSMplex? If not, how many?

   _____

10. Do all systems have access to all volumes?

    _____

11. Will all ML1 (disk archive) volumes be online to all systems?

    _____

12. Which storage management functions are currently being used?

    a. Archive

       _____

b. Backup

_____

c. Automatic restore/recall

_____

d. Merge/recycle of tapes

_____

e. Copy/moving data

_____

f. ISPF

_____

g. Reporting

_____

h. Billing

_____

13. Are end-users performing storage management functions against their own data sets?

_____

14. Is SMS implemented? If yes, what is the current percentage managed by DFSMS?

_____

15. What is the percentage of DFSMS data sets that have a management class associated?

_____

16. How many management classes are defined?

_____

17. Are you using a pool allocation product to direct non-SMS allocations? (CA-Allocate, Allocation Control Center)

_____

18. What types of backups being utilized?

a. Full volume backup

_____

b. Incremental backup

_____

c. Application backups

_____

d. Combination of all

_____

19. What is the average length of time that backups are kept for full volume backups or non-SMS managed data?

_____

20. How many backup versions do you use?

_____

21. How often is archive run?

_____

22. How often is control data set management run? (IXMAINT, FDRARCH, $MAINT)

_____

23. Are tapes being duplexed? Backups, archives?

_____

24. Are you using any user exits? If yes, please document which ones and what they are being used for?

_____

25. What kind of physical tapes are in use? (3490E, 3590)

_____

26. Are you using a virtual tape system? (VTS, VSM)

_____

27. Which disk/RAID subsystems are currently in use? (IBM ESS, EMC, Hitachi)

_____

28. Number of tapes containing archive data needing to be migrated (if known)

_____

29. Number of data sets needing to be migrated (if known)

_____

30. Are you currently using EDM-controlled tapes?

_____

31. Do you currently have a disaster recovery plan in place? If yes, how often do you perform tests?

_____

Processor complex identification

In Table 5-1, enter the following for **each system** in the processor complex:

- Provide the SMF SYSID. The SYSID is set in SYS1.PARMLIB member SMFPRMxx
- Identify the NAME used to reference the system (SYSA, PROD)
- Identify the control data set name on each system (CADISK.FILES, CAASM2.IPC, IDP.APPL.BACKUP)

Table 5-1 Systems and control data set information

SMF SYSID	System name	Control data set name
example 1	PROD	CADISK.FILES
example 2	TEST	CAASM2.IPC

System software

Complete Table 5-2, describing your current software product levels and other profile information.

Please include any explanation of possible upgrades to any and all software taking place using the migration to DFSMShsm and when it is scheduled to take place.

Table 5-2 Software products and levels

Product	Vendor	Current level (include other vendor)	Target level
example DFSMS	IBM	1.4	z/OS 1.1
DFSMS	IBM		
OS/390	IBM		
DFSMShsm	IBM		
DFSMSrmm	IBM		
JES2	IBM		
JES3	IBM		
RACF	IBM		

Product	Vendor	Current level (include other vendor)	Target level
DFSORT	IBM		
Other security system			
Other Storage management system			
Other tape management system			
Other sort package			

DFSMS status

Indicate the current status of your SMS environment:

1. Not installed (planning to run DFSMShsm) version
2. Inactive
3. Active with a minimal configuration
4. Active with a configuration managing disk or temporary data sets
5. Currently managing tape data sets (TMM)
6. System managed tape (Automated tape library)

B

DFSMShsm user training

This appendix provides an example of SMS and DFSMShsm Application Development training to be incorporated as part of the DFSMShsm implementation.

Start of the presentation

Prior to starting this presentation, it is intended that the audience have a basic knowledge of the SMS concepts. The intent of this presentation is to reinforce these basic SMS concepts, and to expand this information on the areas of the system most visible to Application Development. Also it is intended to start the discussion about the desired Service Levels required by the business, as part of the SMS implementation. This will aid in supplying detail for the implementation.

Foil 1: SMS considerations for application design

Figure B-1 Presentation title

Foil 2: SMS considerations — agenda

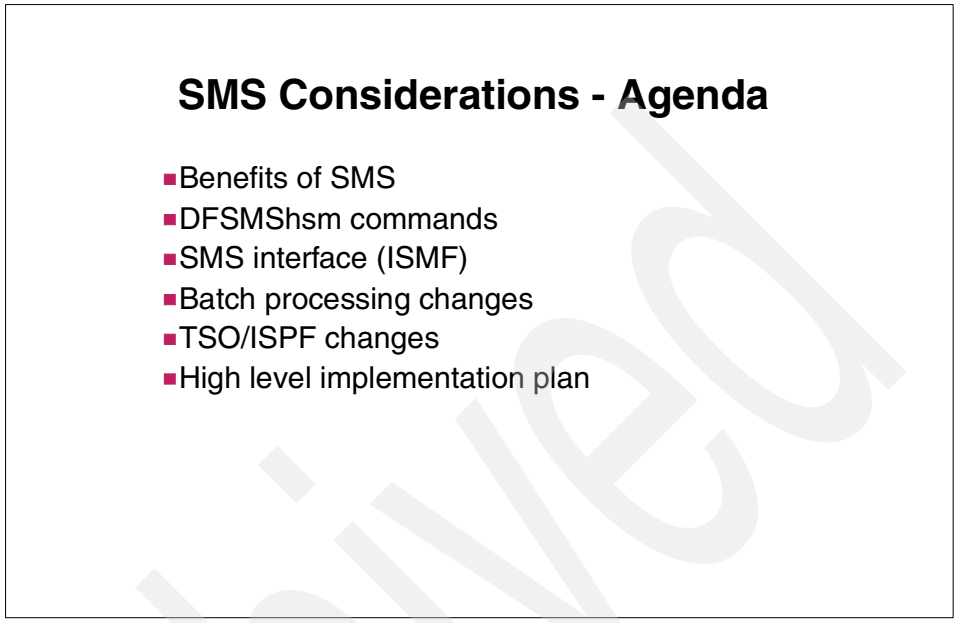


Figure B-2 SMS considerations — agenda

The presentation starts with a list of benefits you can achieve by implementing SMS, and will cover the main areas in application development affected by the SMS implementation. These areas are:

- DFSMShsm commands
- SMS user interface (ISMF)
- Changes required in the batch processing environment
- Changes required in the TSO/ISPF applications
- High level implementation plan

Foil 3: Benefits of SMS

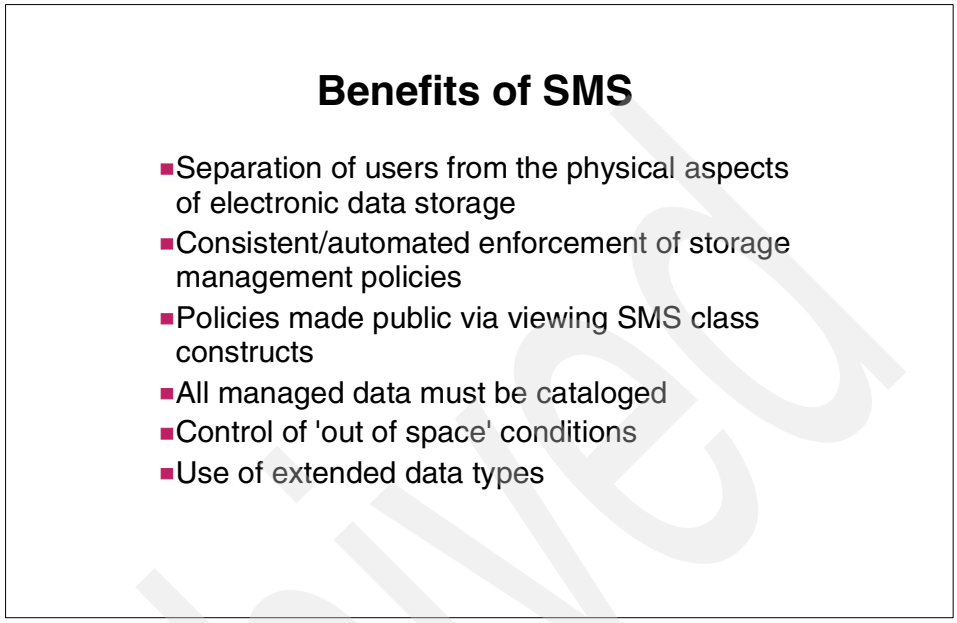


Figure B-3 Benefits of SMS

The SMS benefits are listed in some detail in Chapter 1, “Introduction” on page 1, but the list presented can be modified to cover the details that better suit the specific audience of the presentation. For example, if users have been concerned with their VSAM data sets getting too big or having production delays due to maintenance jobs compressing PDSs, you could list Extended Addressability (EA) and PDSE under extended data types to address these concerns.

Foil 4: DFSMShsm commands

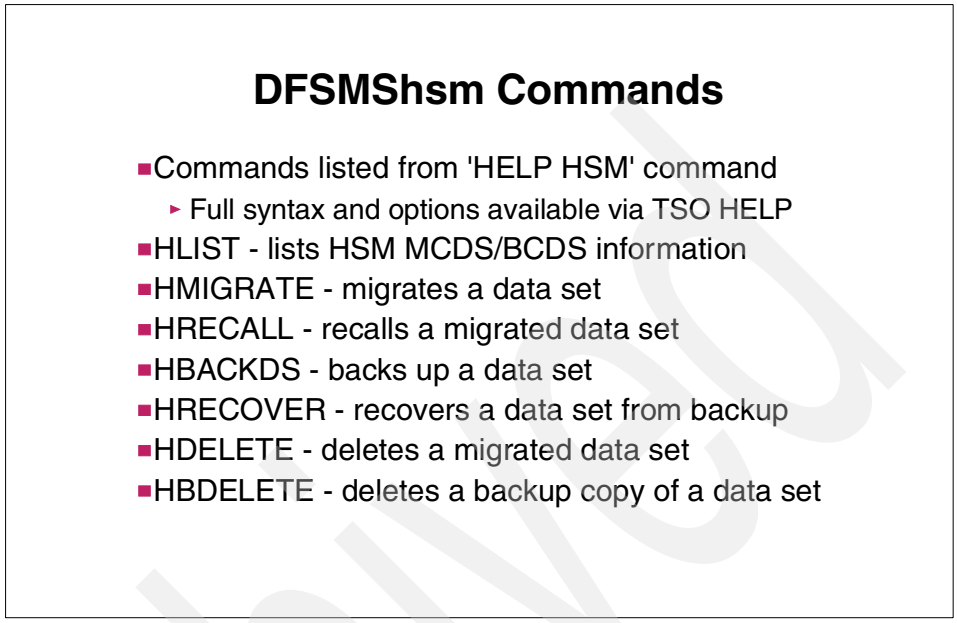


Figure B-4 DFSMShsm commands

The commands list in this foil is obtained by issuing the command **`TSO HELP HSM`**, after copying the member HSM, provided as additional material with this book, in SYS1.HELP. If you do not have this option enabled, then you should omit the first statement and discuss the remainder of the page. The commands are organized as List, Space Management, Availability Management, and Delete commands. There is also an HALTER command that is not included in this list because it applies only to non-SMS managed data sets.

Foil 5: Interactive Storage Management Facility (ISMF)

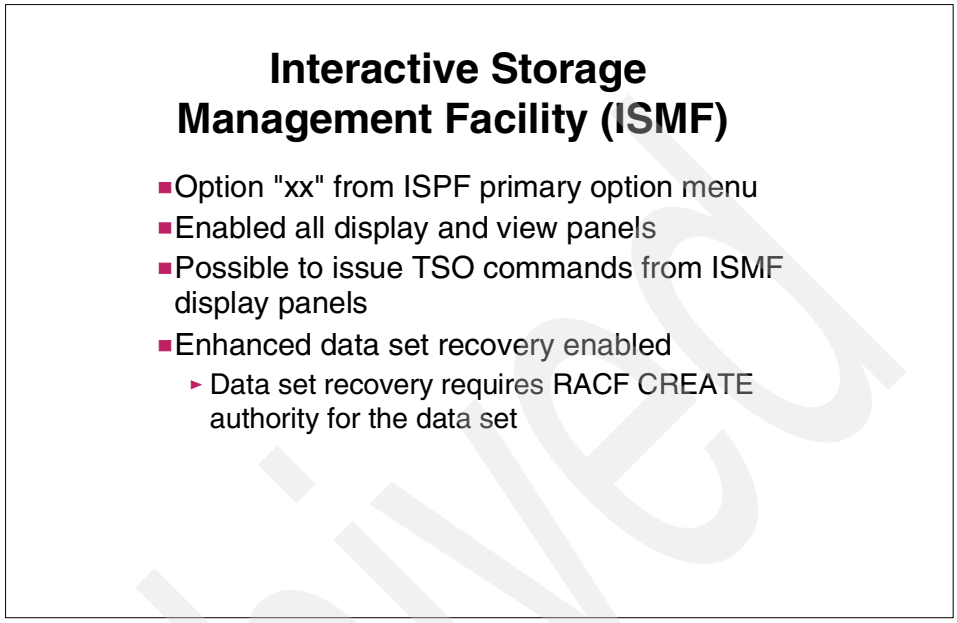


Figure B-5 Interactive Storage Management Facility (ISMF)

The ‘xx’ on the first statement needs to be specified as it is in your installation. The ISMF panels in ISPF/PDF are useful for users, assuming that security measures have been taken that prevent any one accessing ISMF to acquire storage administrator privileges. The options to display volumes, data sets, and SMS class construct contents provide good information. The SMS class constructs can also serve to set users expectations concerning what system services will be provided on their behalf.

Foils 6, 7, 8, 9: Batch/JCL differences

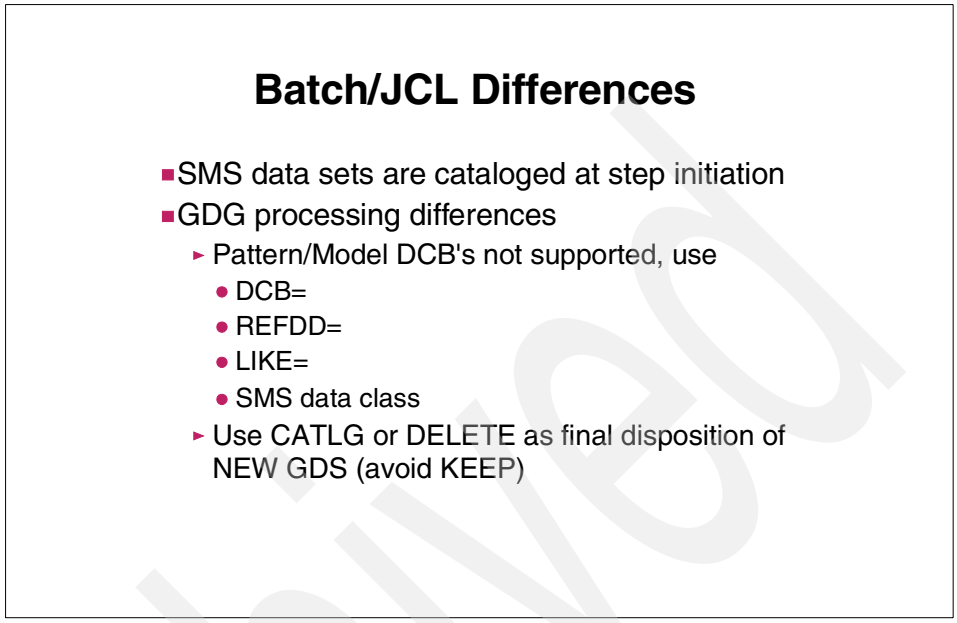


Figure B-6 Batch/JCL differences (Foil 1 of 4)

Most users like the first bullet. They prefer to have a JCL error stating a duplicate data set at the beginning rather than a 'NOT CATLG 2' at the end of the step.

For most users this list does not represent any changes to the way they use the system or how they allocate data. In the past, this has created some discussion, such as what happens at allocation time and why some things work the way they do (for example the defaults of the DISP parameter). A good knowledge of JCL and possibly a little review of the JCL manual will prepare you to lead this discussion.

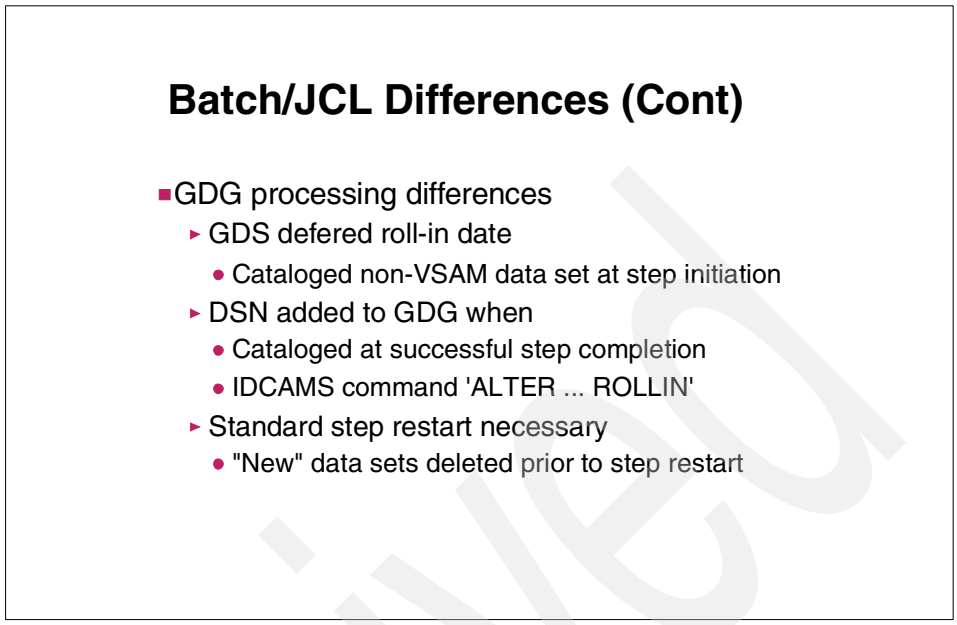


Figure B-7 Batch/JCL differences (Foil 2 of 4)

Some users have gotten very sophisticated in their use of GDGs. When everything works correctly, it should not be any problem in changing to SMS management of GDGs on disk. The area that gets confusing and possibly prone to error is job restarting. If a GDS is in deferred rolling status and is not deleted prior to restarting the step, it will be re-used when the step is restarted. It is not part of the GDS yet (so a **'LISTCAT gdgbasename ASSOC'** command will not show the new member), but is cataloged as a non-VSAM data set.

If the reason for the restart was a x37 abend (out of space) on the new GDS, and corrections are made to the JCL prior to restart to make the size bigger, the changes to the JCL will appear to have no effect. The job will use the existing allocated GDS in deferred rolling state and will get the same abend as before. *Typical* restart from a job scheduling package would delete the outputs created by the step prior to the restart. The processing changes to preserve the integrity of the GDS, and still catalog at start of step, are expecting typical restart logic.

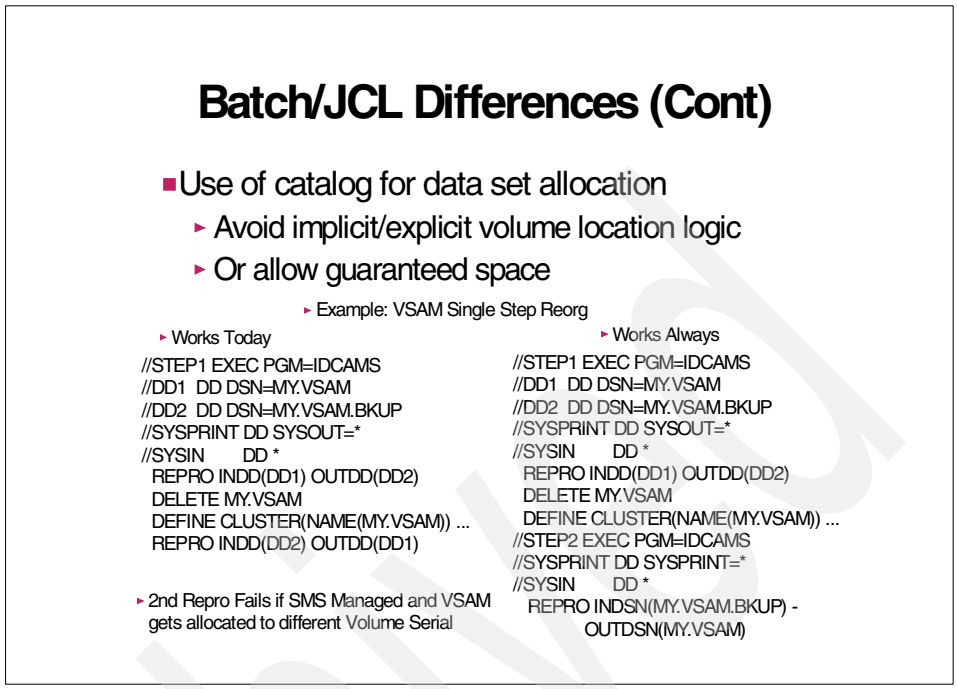


Figure B-8 Batch/JCL differences (Foil 3 of 4)

We are of the opinion that allowing the use of Guaranteed Space removes all of the benefits of having SMS to manage data set placement. We would rather look to see if this single step logic is in use and modify it rather than simply assign Guaranteed Space in the Storage Class. The problem is the use of DD cards that are resolved at Step Initiation: the delete and define of the data set (likely moving the data set to another volume); and the last REPRO failing because the cluster no longer exists on the volume it did when the step started.

There are two techniques to resolve this situation and both are illustrated in this example. The first is step replication. If the last REPRO is performed in a separate step (all JCL replicated) the all will work as intended. The second resolution is to change from using DD cards to DSNs (and dynamic allocation) and then this will continue to work as a single step Reorg. Whether the data set is SMS managed or not, the second example will work (so this modification can be done well ahead of time).

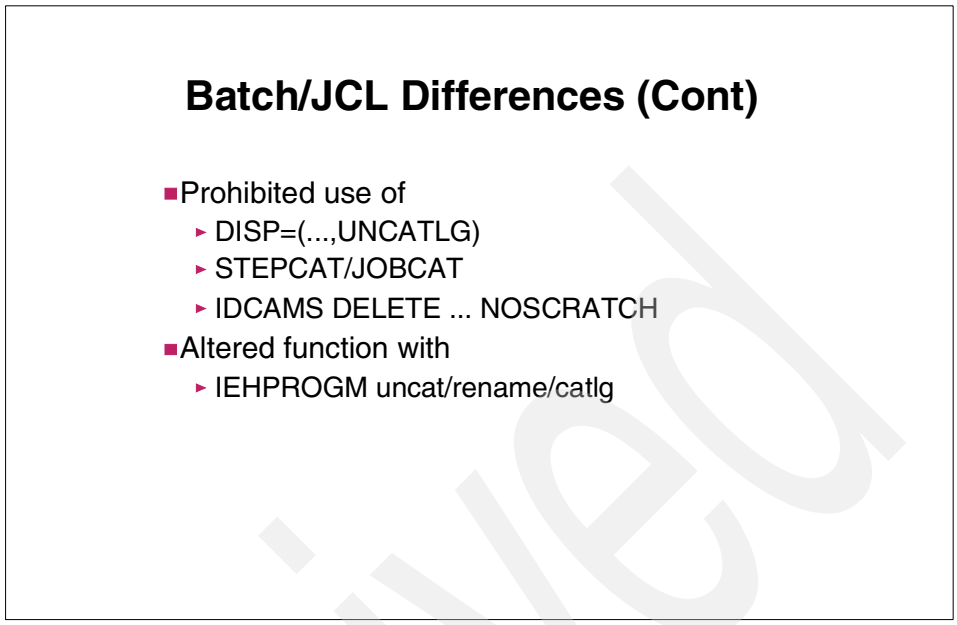


Figure B-9 Batch/JCL differences (Foil 4 of 4)

Most folks do not like to see the word “prohibited”. These items address the DFSMS requirement that the data set be cataloged and accessible via Standard Catalog Search. The only surprise here is that this requirement is actually enforced (and this is the only thing that these items address). If you agree that all data sets should be cataloged and accessible from standard catalog search, then there should be no objection to removing these functions.

We did have some DBAs who had a technique to expand an IMS OSAM database by uncataloging the database and re-cataloging with a volume list. The SMS equivalent is the single IDCAMS statement ‘ALTER entryname ADDVOLUME(*)’. Either technique requires that the database be closed and re-opened to see this change and allocate on the new volume. The SMS advantage is that the system will look for a volume in the storage group with the available space, rather than requiring the DBA to specify the volume serial.

Foil 10: TSO ISPF changes

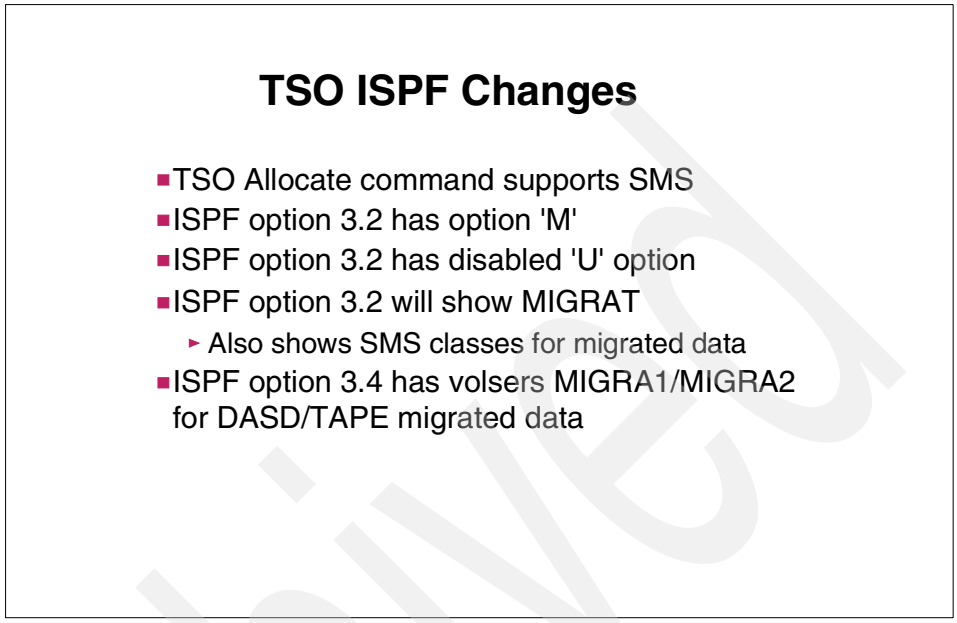


Figure B-10 TSO ISPF changes

Many of the changes to support SMS in TSO or ISPF could easily go unnoticed until they are needed. All the options available in JCL to specify named SMS constructs are incorporated into the TSO Allocate command and option 3.2 of ISPF/PDF.

Foil 11: High level implementation plan

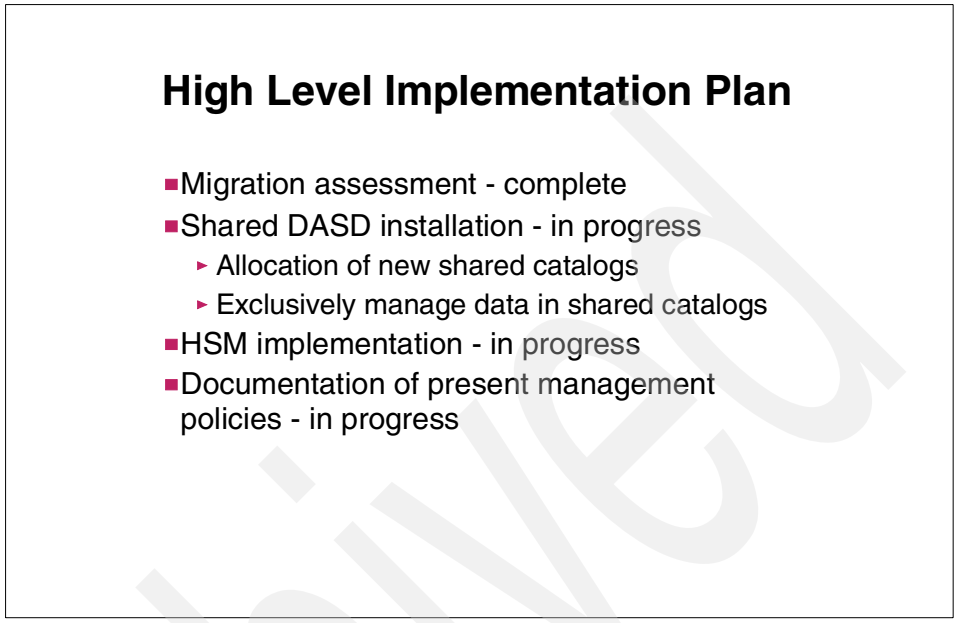


Figure B-11 High level implementation plan

The intent of this information is to show that the project is underway, that the audience is in a position to influence the project's direction, and that there exists a plan to complete the project. It is important to share whatever has been discovered, as some of the audience may have knowledge as to why certain processes exist. I had one customer who was backing up their user catalogs every 6 hours, and could not find a single person in the enterprise who remembered when the last time a catalog needed to be recovered.

Foil 12: Discussion topics

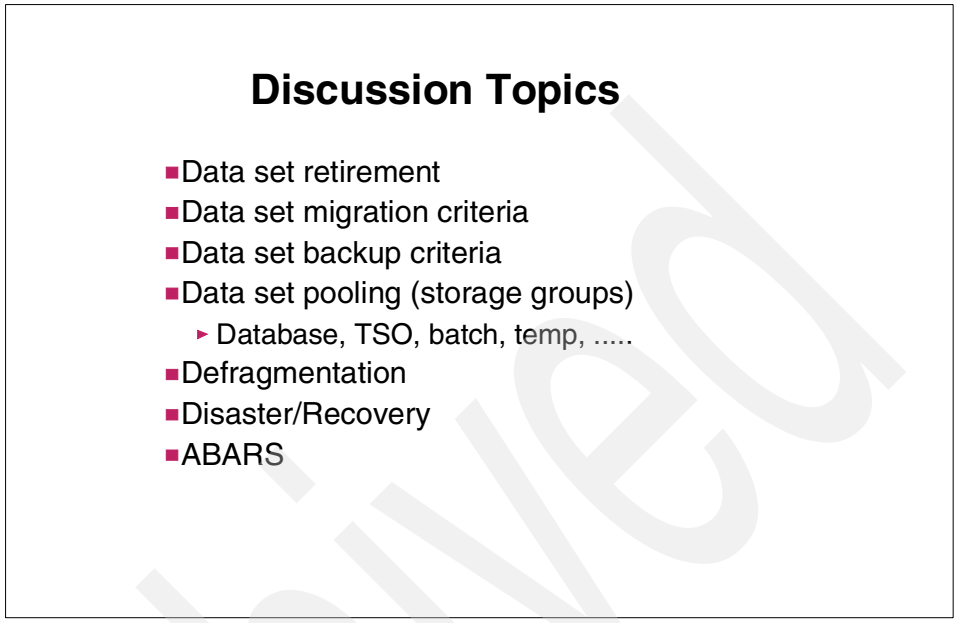


Figure B-12 Discussion topics

This last page is mainly intended to discover additional business requirements either to incorporate with the project or to proceed with it at a later date.

C

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246524

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246524.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	*Description*
samples.zip	Sample utilities and presentations

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 1 MB
Operating System: Any
Processor: Any
Memory: Any

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. Read the README.TXT file included in the material for more information on how to upload the utilities to your MVS system.

Glossary

This glossary contains a list of terms and abbreviations used within this redbook.

A

ABARS. Aggregate backup and recovery support.

ABR. Aggregate backup and recovery record.

access method services. A multifunction service program that manages VSAM and non-VSAM data sets, as well as integrated catalog facility (ICF). Access method services provides the following functions:

- Defines and allocates space for data sets and catalogs
- Converts indexed-sequential data sets to key-sequenced data sets
- Modifies data set attributes in the catalog
- Reorganizes data sets
- Facilitates data portability among operating systems
- Creates backup copies of data sets
- Assists in making inaccessible data sets accessible
- Lists the records of data sets and catalogs
- Defines and builds alternate indexes
- Converts CVOLs to ICF catalogs

accompany data set. In aggregate backup and recovery processing, a data set that is physically transported from the backup site to the recovery site instead of being copied to the aggregate data tape. It is cataloged during recovery.

accompany list. An optional list in the selection data set that identifies the accompany data sets.

ACDS. Active control data set.

ACS. Automatic class selection.

active control data set (ACDS). A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACS is shared by each system that is using the same SMS configuration to manage storage.

active data. Data that is frequently accessed by users and that resides on level 0 volumes.

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

AG. Aggregate group.

aggregate backup. The process of copying the data sets and control information of a user-defined group of data sets so that they may be recovered later as an entity by an aggregate recovery process.

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

aggregate group. A Storage Management Subsystem construct that defines control information and identifies the data sets to be backed up by a specific aggregate backup.

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

ATL. Automated tape library.

audit. A DFSMShsm process that detects discrepancies between data set information in the VTOCs, the computing system catalog, the MCDS, BCDS, and OCDS.

authorized user. In DFSMShsm, the person or persons who are authorized through the DFSMShsm AUTH command to issue DFSMShsm system programmer, storage administrator, and operator commands.

automated tape library. A device consisting of robotic components, cartridge storage frames, tape subsystems, and controlling hardware and software, together with the set of volumes which reside in the library and may be mounted on the library tape drives.

automatic backup. In DFSMShsm, the process of automatically copying eligible data sets from DFSMShsm-managed volumes or migration volumes to backup volumes during a specified backup cycle.

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

automatic class selection (ACS). A mechanism for assigning SMS classes and storage groups.

automatic dump. In DFSMShsm, the process of using DFSMSdss to automatically do a full volume dump of all allocated space on DFSMShsm-managed volumes to designated tape dump volumes.

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

automatic primary space management. In DFSMShsm, the process of automatically deleting expired data sets, deleting temporary data sets, releasing unused overallocated space, and migrating data sets from DFSMShsm-managed volumes.

automatic secondary space management. In DFSMShsm, the process of automatically deleting expired migrated data sets from the migration volumes, deleting expired records from the migration control data set, and migrating eligible data sets from level 1 volumes to level 2 volumes.

automatic space management. In DFSMShsm, includes automatic volume space management, automatic secondary space management, and automatic recall.

automatic volume space management. In DFSMShsm, includes automatic primary space management and automatic interval migration.

availability management. In DFSMShsm, the process of ensuring that a current version (backup copy) of the installation's data sets resides con tape or disk.

B

backup control data set (BCDS). A VSAM, key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

backup copy. In DFSMShsm, a copy of a data set that is kept for reference in case the original data set is destroyed.

backup cycle. In DFSMShsm, a period of days for within a pattern is used to specify the days in the cycle on which automatic backup is scheduled to take place.

backup frequency. In DFSMShsm, the number of days that must elapse since the last backup version of a data set was made until a changed data set is again eligible for backup.

backup version. Synonym for backup copy.

backup volume. A volume managed by DFSMShsm to which backup versions of data sets are written.

backup. In DFSMShsm, the process of copying a data set residing on a level 0 volume, a level 1 volume, or a volume not managed by DFSMShsm to a backup volume.

base configuration. The part of an SMS configuration that contains general storage management attributes., such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

base sysplex. A base (or basic) sysplex is the set of one or more MVS systems that is given a cross-system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A base system does not include a coupling facility.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment.

BCDS. Backup control data set.

BCS. Basic catalog structure.

C

CDS. Control data set.

CF. Coupling facility.

COMMDS. Communications data set.

communications data set (COMMDS). The primary means of communications among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS.

compaction. In DFSMShsm, a method of compressing and encoding data that is migrated or backed up.

compress. To reduce the amount of storage required for a given data set by having the system replace identical words or phrases with a shorter token associated with the word or phrase.

compressed format. A particular type of extended-format data set specified with the COMPACTION parameter of data class. VSAM can compress individual records in a compressed-format data set. SAM can compress individual blocks in a compressed-format data set.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

construct. One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

control data set. (1) In DFSMShsm, one of three data sets (BCDS, MCDS, and OCDS) that contain records used in DFSMShsm processing.

coupling facility (CF). The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

D

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

Data Facility Sort (DFSORT). An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copy operations, as well as providing versatile data manipulation at the record, field, and bit level.

Data Facility Storage Management Subsystem (DFSMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

device category. A storage device classification used by SMS. The device categories are as follows: SMS-managed disk, SMS-managed tape, non-SMS-managed disk, non-SMS-managed tape.

DFSMS. Data Facility Storage Management System.

DFSMSdfp. A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data management.

DFSMSdss. A DFSMS functional component or base element of z/OS, used to copy, move dump, and restore data sets or volumes.

DFSMShsm. A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm-managed volume. (1) A primary storage volume, which is defined to DFSMShsm but which does not belong to a storage group. (2) A volume in a storage group, which is using DFSMShsm automatic dump, migration, or backup services.

DFSMShsm-owned volume. A storage volume on which DFSMShsm stores backup versions, dump copies, or migrated data sets.

DFSMSrmm. A DFSMS functional component or base element of z/OS, that manages removable media.

disaster backup. A means to protect a computing rm2 definition.

disaster recovery. A procedure for copying and storing an installation's essential business data in a secure location, and for recovering that data in the event of a catastrophic problem.

dummy storage group. A type of storage group that contains the serial numbers of volumes no longer connected to a system. Dummy storage groups allow existing JCL to function without having to be changed.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMShsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

E

EA. Extended addressability.

esoteric unit name. A name used to define a group of devices having similar hardware characteristics, such as TAPE or SYSDA.

expiration. The process by which data sets or objects are identified for deletion because their expiration data or retention period has passed. On disk, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse.

extended addressability. The ability to create and access a VSAM data set that is greater than 4 GB in size. Extended addressability data sets must be allocated with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

extended format. The format of a data set that has a data set name type of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different.

extent reduction. In DFSMShsm, the releasing of unused space, reducing the number of extents, and compressing partitioned data sets.

F

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully or partially-qualified data set names or of certain data set characteristics.

FSR. Functional statistics record.

functional statistics record (FSR). A record that is created each time a DFSMShsm function is processed. It contains a log of system activity and is written to the system management facilities (SMF) data set.

G

GB. Gigabyte.

GDG. Generation data group.

GDS. Generation data set.

generation data group (GDG). A collection of data sets with the same base name, such as PAYROLL, that are kept in chronological order. Each data set is called generation data set (GDS).

generic unit name. A name assigned to a class of devices with the same geometry (such as 3390).

global resource serialization (GRS). A component of z/OS used for serializing use of system resources and for converting hardware reserves on disk volumes to data set enqueues.

global scratch pool. A group of empty tapes that do not have unique serial numbers and are not known individually to DFSMShsm. The tapes are not associated with a specific device.

GRS. Global resource serialization.

H

hierarchical file system (HFS) data set. A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using z/OS UNIX System Services.

HMT. HSM Monitor/Tuner.

HSM complex (HSMplex). One or more z/OS images running DFSMShsm that share a common set of control data sets (MCDS, BCDS, OCDS, and Journal).

I

inactive data. Copies of active or low-activity data that reside on DFSMShsm-owned dump and incremental backup volumes.

incremental backup. In DFSMShsm, the process of copying a data set that has been opened for other than read-only access since the last backup version was created, and that has met the backup frequency criteria.

inline backup. The process of copying a specific data set to a migration level 1 volume from a batch environment. This process allows you to back up data sets in the middle of a job.

in-place conversion. The process of bringing a volume and the data sets it contains under the control of SMS without data movement, using DFSMSdss.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS that allows users and storage administrators access to the storage management functions.

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

journal data set. In DFSMShsm, a sequential data set used by DFSMShsm for recovery of the MCDS, BCDS, and OCDS. The journal contains a duplicate of each record in the control data sets that has changed since the MCDS, BCDS, and OCDS were last backed up.

K

KB. Kilobyte; 1024 bytes.

level 0 volume. A volume that contains data sets directly accessible by the user. The volume may be either DFSMShsm-managed or non-DFSMShsm-managed.

level 1 volume. A volume owned by DFSMShsm containing data sets migrated from a level 0 volume.

level 2 volume. A volume under control of DFSMShsm containing data sets that migrated from a level 0 volume, from a level 1 volume, or from a volume not managed by DFSMShsm.

M

management class. A named collection of management attributes describing the retention, backup, and class transition characteristics for a group of objects in an object storage hierarchy.

manual tape library. Installation-defined set of tape drives defined as a logical unit together with the set of system-managed volumes which can be mounted on the drives. The IBM implementation includes one or more 3490 subsystems, each connected by a Library Attachment Facility to a processor running the Library Manager application, and a set of volumes, defined by the installation as part of the library, which resides in shelf storage located near the 3490 subsystems.

MB. Megabyte; 1,048,576 bytes.

MCB. BCDS data set record.

MCC. Backup version record.

MCD. MCDS data set record.

MCDS. Migration control data set.

MCT. Backup volume record.

MCV. Primary and migration volume record.

MEDIA2. Enhanced Capacity Cartridge System Tape.

MEDIA3. High Performance Cartridge Tape.

MEDIA4. Extended High Performance Cartridge Tape.

migration control data set (MCDS). In DFSMShsm, a VSAM key-sequenced data set that contains records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMShsm.

migration level 1. DFSMShsm-owned disk volumes that contain data sets migrated from primary storage volumes. The data can be compressed.

migration level 2. DFSMShsm-owned tape or disk volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled.

ML1. Migration level 1.

ML2. Migration level 2.

MTL. Manual tape library.

N

NaviQuest. A component of DFSMSdfp for implementing, verifying, and maintaining your SMS environment in batch mode. It provides batch testing and reporting capabilities that can be used to automatically create test cases in bulk, run many other storage management tasks in batch mode, and use supplied ACS code fragments as models when creating your own ACS routines.

non-DFSMShsm-managed volume. A volume not defined to DFSMShsm containing data sets that are directly accessible to users.

O

OAM. Object access method.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

object. A named byte stream having no specific format or record orientation.

OCDS. Offline control data set.

offline control data set (OCDS). In DFSMShsm, a VSAM, key-sequenced data set that contains information about tape backup volumes and tape migration level 2 volumes.

P

parallel sysplex. A sysplex with one or more coupling facilities, and defined by the COUPLExx members of SYS1.PARMLIB as being a parallel sysplex.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

pool storage group. A type of storage group that contains system-managed disk volumes. Pool storage groups allow groups of volumes to be managed as a single entity.

primary space allocation. Amount of space requested by a user for a data set when it is created.

primary storage. A disk volume available to users for data allocation. The volumes in primary storage are called primary volumes.

R

RACF. Resource Access Control Facility.

recall. The process of moving a migrated data set from a level 1 or level 2 volume to a DFSMShsm-managed volume or to a volume not managed by DFSMShsm.

record-level sharing (RLS). An extension to VSAM that provides direct shared access to a VSAM data set from multiple systems using cross-system locking.

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

relative track address (TTR). Relative track and record address on a direct-access device, where TT represents two bytes specifying the track relative to the beginning of the data set, and R is one byte specifying the record on that track.

Resource Access Control Facility (RACF). An IBM licensed program that provides access control by identifying users to the system; verifying users of the system; authorizing access to protected resources; logging detected, unauthorized attempts to enter the system; and logging detected accesses to protected resources. RACF is included in z/OS Security Server and is also available as a separate program for the MVS and VM environments.

Resource Measurement Facility (RMF). An IBM licensed program or optional element of z/OS, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

restore. In DFSMShsm, the process of invoking DFSMSdss to perform the program's recover function. In general, it is to return to an original value or image, for example, to restore data in main storage from auxiliary storage.

RLS. Record-level sharing.

RMF. Resource Measurement Facility.

S

SCDS. Source control data set.

SDSP. Small data set packing.

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full.

service-level agreement. (1) An agreement between the storage administration group and a user group defining what service-levels the former will provide to ensure that users receive the space, availability, performance, and security they need. (2) An agreement between the storage administration group and operations defining what service-level operations will provide to ensure that storage management jobs required by the storage administration group are completed.

shelf location. A single space on a shelf for storage of removable media.

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

small data set packing (SDSP). In DFSMShsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

small-data-set-packing data set. In DFSMShsm, a VSAM key-sequenced data set allocated on a migration level 1 volume and containing small data sets that have migrated.

SMF. System management facilities.

SMS complex. A collection of systems or system groups that share a common configuration. All systems in an SMS complex share a common active control data set (ACDS) and a communications data set (COMMDS). The systems or system groups that share the configuration are defined to SMS in the SMS base configuration.

SMS configuration. A configuration base, Storage Management Subsystem class, group, library, and drive definitions, and ACS routines that the Storage Management Subsystems uses to manage storage.

SMS control data set. A VSAM linear data set containing configurational, operational, or communications information that guides the execution of the Storage Management Subsystem.

SMS. Storage Management Subsystem.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF.

space management. In DFSMShsm, the process of managing aged data sets on DFSMShsm-manages and migration volumes. The three types of space management are: migration, deletion, and retirement.

specific scratch pool. A group of empty tapes with unique serial numbers that are known to DFSMShsm as a result of being defined to DFSMShsm with the ADDVOL command.

spill storage group. An SMS storage group used to satisfy allocations which do not fit into the primary storage group.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage control. The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of disk volumes, or a group of disk, optical, or tape volumes treated as a single object storage hierarchy.

storage hierarchy. An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, disk cache), primary storage (disk containing uncompressed data), migration level 1 (disk containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format).

storage location. A location physically separate from the removable media library where volumes are stored for disaster recovery, backup, and vital records management.

Storage Management Subsystem (SMS). A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage management. The activities of data set allocation, placement, monitoring, migration. backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automated these processes for you, while optimizing storage resources.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

sysplex. A set of MVS or z/OS systems communicating and cooperating with each other through certain multi-system hardware components and software services to process customer workloads.

system management facilities (SMF). A component of z/OS that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

system-managed data set. A data set that has been assigned a storage class.

system-managed storage. An approach to storage management in which the system determines data placement and an automatic data manager handles data backup, movement, space, and security.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual.

system-managed volume. A disk, optical, or tape volume that belongs to a storage group.

T

tape configuration database (TCDB). One or more volume catalogs used to maintain records of system-managed tape libraries and tape volume.

Tape Library Dataserver. A hardware device that maintains the tape inventory associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes.

tape library. A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives.

tape mount management. The methodology used to optimize tape subsystem operation and use, consisting of hardware and software facilities used to manage tape data efficiently.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

TB. Terabyte.

TTOC. Tape table of contents record.

TTR. Relative track address.

U

unit affinity. Requests that the system allocate different data sets residing on different removable volumes to the same device during execution of the step to reduce the total number of tape drives required to execute the step. Explicit unit affinity is specified by coding the UNIT=AFF JCL keyword on a DD statement. Implicit unit affinity exists when a DD statement requests more volumes that devices.

use attribute. (1) The attribute assigned to a disk volume that controls when the volume can be used to allocate new data sets; use attributes are *public*, *private*, and *storage*. (2) For system-managed tape volumes, use attributes are *scratch* and *private*.

V

virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

vital records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement.

volume status. In the Storage Management Subsystem, indicates whether the volume is fully available for system management:

- "Initial" indicates that the volume is not ready for system management because it contains data sets that are ineligible for system management.
- "Converted" indicates that all of the data sets on a volume have an associated storage class and are cataloged in an integrated catalog facility catalog.
- "Non-system-managed" indicates that the volume does not contain any system-managed data sets and has not been initialized as system-managed.

volume. The storage space on disk, tape. or optical devices, which is identified by a volume label.

volume pool. In DFSMShsm, a set of related primary volumes. When a data set is recalled, if the original volume that it was on is in a defined volume pool, the data set can be recalled to one of the volumes in the pool.

VTOC. Volume table of contents.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 165.

- *DFSMShsm Primer*, SG24-5272
- *Get DFSMS FIT: Fast Implementation Techniques, SG24-2568*
- *Maintaining Your SMS Environment*, SG24-5484
- *DFSMShsm ABARS and Mainstar Solutions*, SG24-5089
- *DFSMS/MVS V1R2 Data Compression Implementation Guide*, GG24-4251
- *Storage Management with DB2 for OS/390*, SG24-5462
- *Hierarchical File System Usage Guide*, SG24-5482
- *NaviQuest Demonstration and Hands-On Usage Guide*, SG24-4720
- *DFSMS Optimizer Usage Guide*, SG24-2235
- *DFSMS Optimizer: The New HSM Monitor/Tuner*, SG24-5248
- *CICS and VSAM Record Level Sharing*: Implementation Guide, SG24-4760
- *CICS and VSAM Record Level Sharing*: Planning Guide, SG24-4765

Other resources

These publications are also relevant as further information sources:

- *z/OS V1R1.0-V1R2.0 DFSMS Implementing System Managed Storage*, SC26-7407
- *z/OS V1R1.0-V1R2.0 DFSMS Introduction*, SC26-7397
- *z/OS V1R1.0-V1R2.0 DFSMS Managing Catalogs*, SC26-7409
- *z/OS V1R1.0-V1R2.0 DFSMS Using the Interactive Storage Management Facility*, SC26-7411
- *z/OS V1R1.0-V1R2.0 DFSMSdfp Storage Administration Reference*, SC26-7402

- *z/OS V1R1.0-V1R2.0 DFSMSdss Storage Administration Guide*, SC35-0423
- *z/OS V1R1.0-V1R2.0 DFSMSdss Storage Administration Reference*, SC35-0424
- *z/OS V1R1.0-V1R2.0 DFSMShsm Implementation and Customization Guide*, SC35-0418
- *z/OS V1R1.0-V1R2.0 DFSMShsm Managing Your Own Data*, SC35-0420
- *z/OS V1R1.0-V1R2.0 DFSMShsm Storage Administration Guide*, SC35-0421
- *z/OS V1R1.0-V1R2.0 DFSMSrmm Implementation and Customization Guide*, SC26-7405
- *z/OS V1R1.0-V1R2.0 DFSMS Using Data Sets*, SC26-7410
- *z/OS V1R1.0-V1R2.0 DFSMS Using the Volume Mount Analyzer*, SC26-7413
- *z/OS V1R1.0-V1R2.0 DFSMShsm Diagnosis Guide*, LY35-0114
- *z/OS V1R1.0-V1R2.0 DFSMShsm Diagnosis Reference*, LY35-0115
- *z/OS V1R1.0-V1R2.0 DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*, SC35-0427
- *z/OS V1R1.0-V1R2.0 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426
- *DFSMS Optimizer V1R2 User's Guide and Reference*, SC26-7047
- *DFSMS/MVS V1R5 NaviQuest User's Guide*, SC26-7194
- *DFSMS/MVS V1R5 Planning for Installation*, SC26-4919
- *z/OS V1R2.0 MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS V1R2.0 ISPF Planning and Customizing*, GC34-4814

Referenced Web sites

These Web sites are also relevant as further information sources:

- DFSMS

  http://www.storage.ibm.com/software/sms/index.html
- DFSMS/MVS Optimizer

  http://www.storage.ibm.com/software/opt/index.html
- Storwatch DFSMShsm Monitor

  http://www.storage.ibm.com/software/storwatch/swhsm/index.html
- z/OS V1R1 DFSMS Migration Planning Wizard

  http://www-1.ibm.com/servers/eserver/zseries/zos/wizards/dfsms/migplanasst/

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

C

D

E

F

G

H

I

J

L

M

N

T

U

V

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Converting to DFSMShsm:

A Practical Approach

Want to convert to DFSMShsm? This should be your starting point

Simplify your storage management tasks with SMS

Exploit new hardware functions as they come

The intended audience for this IBM Redbook are customers that have achieved partial implementation of the IBM DFSMS product suite and are looking to further their implementation through the use of the DFSMShsm product. It should be reassuring to know that many other customers have converted from products such as FDR/ABR, CA-ASM2 or CA-Disk to DFSMShsm. It is also important to acknowledge that such conversions are not completed overnight, and that the customer's personnel will need to be intensively involved in their conversion project from its conception.

Most conversions require a period of product coexistence to allow the migration to proceed. This provides time for the aging of the prior product's archive/backup data and reduces the data migration requirements. In some circumstances, it may be possible to hasten the conversion by installing enough disk to allow pre-restoration of data to subsequently backup/migrate with DFSMShsm.

This book will take you through everything we have learned about performing these conversions. We cover the elements of preparation, activation, conversion, and exploitation, using examples to illustrate these activities from projects that we have performed.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6524-00 ISBN 0738424412