Redbooks
ibm.com/redbooks

IBM z/OS DFSMShsm Primer

Keith Winnard

Jose Gilberto Biondo

Gert Laumann

Norbert Sclumberger

Jeanne Vangsness

Chao Xing Ma

z Systems

IBM

Redbooks

International Technical Support Organization

IBM z/OS DFSMShsm Primer

September 2015

SG24-5272-03

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

Fourth Edition (September 2015)

This edition applies to Version 2, Release 1 of Data Facility Storage Management Subsystem (DFSMS) (product number 5650-ZOS).

Contents

Notices xi
Trademarks xii

Preface xiii
Authors xiii
Now you can become a published author, too! xiv
Comments welcome xv
Stay connected to IBM Redbooks xv

Summary of changes xvii
September 2015, Fourth Edition xvii

Chapter 1. DFSMShsm overview 1
1.1 Introduction 2
1.2 Storage device hierarchy 2
1.3 SMS and DFSMShsm relationship 3
1.4 DFSMShsm control data sets 5
1.5 Data set lifecycle with DFSMShsm 5
1.5.1 Data set create 6
1.5.2 Data set backup 6
1.5.3 Data set update 7
1.5.4 Data set migration 7
1.5.5 Data set recall 8
1.5.6 Data set expiration 9
1.5.7 Data set recovery 9
1.5.8 Expiring backup versions 10
1.6 Space management 10
1.6.1 Recycle activity 11
1.6.2 Primary space management 11
1.6.3 Secondary space management 12
1.6.4 Interval migration 12
1.6.5 On-demand migration 13
1.7 Availability management 13
1.7.1 Backup and recovery 13
1.7.2 Dump 15
1.8 Tape processing 15

Chapter 2. Planning and reviewing your environment 17
2.1 Considerations 18
2.2 Determine how you want DFSMShsm to manage your data 19
2.2.1 Planning for the space management process 19
2.2.2 Planning for availability management 20
2.3 Tape duplexing 22
2.4 Data encryption 22
2.4.1 zEDC hardware compression 22
2.5 Using small-data-set packing 23
2.6 Expired data sets and tape recycle 24
2.7 SMS classes and ACS routines 24
2.8 Security aspects in the DFSMShsm environment 24

2.9 Invoking DSS to move data . . . 25
2.10 Interactions with object access method . . . 26
2.11 Interaction with the DFSMSrmm tape management system . . . 27
2.12 Preparation for problem diagnostic determination . . . 27

Chapter 3. Getting started . . . 29
3.1 DFSMShsm components . . . 30
3.2 DFSMShsm starter set . . . 30
3.2.1 Contents of HSM.SAMPLE.CNTL . . . 31
3.2.2 ALLOCBK1 . . . 50
3.2.3 ALLOSDSP . . . 50
3.2.4 ARCCMD01 . . . 50
3.2.5 ARCCMD90 . . . 51
3.2.6 ARCCMD91 . . . 51
3.2.7 HSMHELP . . . 52
3.2.8 HSMLOG . . . 52
3.2.9 HSMEDIT . . . 53
3.2.10 HSMPRESS . . . 54
3.3 Media that is owned by DFSMShsm . . . 54
3.4 DFSMShsm operations . . . 54
3.4.1 Stopping DFSMShsm . . . 54
3.4.2 Starting DFSMShsm . . . 55
3.4.3 MASH implications . . . 56
3.4.4 Sending commands to DFSMShsm . . . 57
3.4.5 Secondary address spaces . . . 57
3.4.6 Automation . . . 57
3.4.7 Host promotion . . . 57

Chapter 4. SMS and its interactions with DFSMShsm . . . 59
4.1 DFSMShsm in an SMS environment . . . 60
4.2 POOL storage group . . . 61
4.2.1 Space management . . . 62
4.2.2 Availability management . . . 63
4.3 Tape storage group . . . 64
4.4 Management class . . . 64
4.4.1 Expiration . . . 65
4.4.2 Partial release . . . 66
4.4.3 Migration . . . 67
4.4.4 GDG management . . . 68
4.4.5 Backup attributes . . . 68
4.4.6 ABARS attributes . . . 71
4.5 Storage class . . . 72
4.6 Values for SMS classes and groups . . . 74
4.6.1 Naming classes and groups . . . 74
4.6.2 How many classes and groups . . . 75
4.7 Automatic class selection . . . 75
4.7.1 Coding for data that is owned by DFSMShsm and control data . . . 76
4.7.2 ACS execution for data that is managed by DFSMShsm . . . 77
4.8 SMS and non-SMS configuration considerations . . . 78

Chapter 5. Implementing DFSMShsm security . . . 79
5.1 Providing security for DFSMShsm resources . . . 80
5.1.1 Defining a DFSMShsm RACF user ID . . . 80
5.1.2 Identifying started procedures to RACF . . . 81

5.1.3 Protecting DFSMShsm data sets 84
5.1.4 Controlling the use of DFSMShsm operator-entered commands 86
5.1.5 Authorizing and protecting commands by using AUTH. 88
5.1.6 Authorizing and protecting commands by using RACF. 88
5.1.7 RACF protection for ABARS 91
5.1.8 RACF protection for Concurrent Copy 92
5.1.9 ARCCATGP group 92
5.1.10 Protecting migration and backup data sets. 93

Chapter 6. Data set format support, record-level sharing, and extended address volumes 95
6.1 Record-level sharing 96
6.1.1 VSAM record-level sharing 96
6.2 RLS implementation 97
6.2.1 Defining the SHCDSs 98
6.2.2 Coupling facility cache and lock structures 98
6.2.3 Altering the SMS configuration 100
6.2.4 Storage class changes 101
6.2.5 Altering IGDSMSxx PARMLIB member 103
6.2.6 Activating the SMSVSAM address space 103
6.2.7 Activating the SHCDS. 103
6.2.8 DFSMShsm procedure changes. 103
6.3 RLS implementation checklist 104
6.3.1 DFSMShsm support for RLS-managed ICF catalogs 104
6.3.2 RLS considerations. 105
6.4 EAV support, considerations, and coexistence. 105
6.4.1 DFSMShsm EAV support and considerations 105
6.4.2 z/OS DFSMShsm earlier level coexistence 112

Chapter 7. Tape considerations 117
7.1 Basic tape setup that is needed for DFSMShsm 118
7.1.1 DFSMShsm single file tape operation. 118
7.2 Tape device allocations. 122
7.2.1 Tape device selection 122
7.2.2 Non-library tape device selection 123
7.2.3 Library tape device selection 123
7.2.4 Specifying the WAIT option. 123
7.2.5 Specifying the NOWAIT option 123
7.2.6 Specifying esoteric tape unit names to DFSMShsm. 124
7.3 Protecting DFSMShsm-owned tapes 125
7.3.1 Protecting DFSMShsm tapes 125
7.4 DFSMShsm interaction with a tape management system. 129
7.5 Running DFSMShsm with DFSMSrmm 130
7.5.1 Authorizing DFSMShsm to DFSMSrmm resources 130
7.5.2 Authorizing ABARS to DFSMSrmm resources 131
7.5.3 Setting DFSMSrmm options when using DFSMShsm 131
7.5.4 Setting DFSMShsm options when using DFSMSrmm 132
7.5.5 Setting DFSMShsm system options 132
7.5.6 Setting DFSMShsm dump definitions 133
7.5.7 DFSMSrmm support for DFSMShsm naming conventions. 133
7.6 Using an SMS-managed tape library 143
7.6.1 Converting to a DFSMS-managed library 143
7.6.2 Identifying tape library tapes 144
7.7 Using Virtual Tape Servers for DFSMShsm 145

7.7.1 Considerations for using VTS/TS7700 for DFSMShsm 145
7.7.2 Larger logical volume sizes in TS7700 145
7.7.3 The TS7700 and awareness of tape capacity. 147
7.7.4 The TS7700 and the MOUNTWAITTIME setting 148
7.7.5 The TS7700 and scratch volumes 148
7.7.6 The TS7700 and DFSMShsm DUMP parameters 148
7.7.7 The TS7700 physical limitations 148
7.7.8 The TS7700 and tape management systems. 148
7.8 Preparing for a Virtual Tape Server or an ATL 149
7.8.1 Determine the functions to process in a VTS/tape library. 149
7.9 Setting up a global scratch pool 150
7.10 Setting up a DFSMShsm-owned scratch pool 151
7.11 Preparing for SMS-managed tape 151
7.11.1 Defining a storage class 151
7.11.2 Defining a data class. 153
7.12 Preparing DFSMShsm for using tape 160
7.12.1 Usage of DFSMShsm RECYCLE 161
7.12.2 DB2 and DFSMShsm interaction 162
7.13 DFSMShsm tape functions 162
7.13.1 Disaster recovery considerations 163
7.14 Tape mount management. 175
7.15 DFSMShsm and 3592 hardware support 177
7.15.1 DFSMShsm modifications for 3592 Model J. 177
7.15.2 Planning and installation. 177
7.15.3 Input tape selection. 177
7.15.4 Output tape selection 178
7.15.5 Output tape utilization 178
7.15.6 Partial tapes 178
7.15.7 Reuse capacity table. 179
7.15.8 Worm tape cartridge rejection. 179
7.16 DFSMShsm use of performance scaling on 3592. 179
7.16.1 How to use performance scaling and segmentation for 3592. 179
7.16.2 Activating performance segmentation or scaling 180
7.16.3 List of SETSYS commands that are used for tape 180
7.16.4 Commands that address the tape management system. 181

Chapter 8. HSMplexes, GRSplexes, and sysplexes 183
8.1 Introduction to the HSMplex 184
8.1.1 Sample HSMplex 184
8.2 Data integrity and resource serialization 185
8.2.1 Global resource serialization. 185
8.2.2 Serialization groups 191
8.2.3 Using GRS resource name lists 195
8.2.4 VSAM SHAREOPTIONS for CDSs. 197
8.3 Starting DFSMShsm in an HSMplex 198
8.3.1 Defining a primary DFSMShsm. 198
8.3.2 Defining all DFSMShsm hosts in a multihost environment 199
8.4 HSMplex considerations 199
8.4.1 CDS backup version considerations 199
8.4.2 Problem determination aid considerations 200
8.4.3 Small-data-set-packing data set considerations 200
8.4.4 Volume considerations 200
8.4.5 JES3 considerations 200

8.4.6 Multitasking considerations . . . 201
8.5 DFSMShsm in a sysplex environment . . . 201
8.5.1 Configuring multiple HSMplexes in a sysplex . . . 202
8.5.2 Single GRSplex serialization in a sysplex environment . . . 202
8.5.3 Secondary host promotion . . . 205
8.5.4 Control data set extended addressability in a sysplex . . . 208
8.5.5 Common recall queue . . . 208

Chapter 9. Space management . . . 211
9.1 Functions in primary space management . . . 212
9.1.1 Tiers used in DFSMShsm space management. . . . 212
9.2 Components in space management . . . 213
9.2.1 Automatic primary space management function . . . 213
9.2.2 Secondary space management . . . 217
9.2.3 Event-driven space management . . . 219
9.2.4 DFSMShsm on demand migration . . . 220
9.2.5 Storage group processing priority for space management functions . . . 222
9.2.6 Automatic dynamic volume expansion space management . . . 223
9.2.7 Command space management . . . 223
9.2.8 Fast Subsequent Migration optional feature . . . 225
9.2.9 Automatic recall . . . 226
9.2.10 Common recall queue . . . 227
9.3 Space management of non-SMS-managed volumes . . . 232
9.3.1 Primary space management . . . 232
9.3.2 Automatic secondary space management . . . 235
9.3.3 Automatic primary space management. . . . 235
9.3.4 Migration subtasking . . . 240

Chapter 10. Availability management means backup . . . 243
10.1 Backup availability management. . . . 244
10.1.1 Automatic backup . . . 244
10.1.2 Backing up data sets with a command . . . 252
10.1.3 Backup non-SMS-managed data sets . . . 256
10.1.4 Backing up a data set manually and the RETAINDAYS keyword. . . . 257
10.1.5 Using DSSXMMODE . . . 262
10.1.6 Interrupting and restarting backup processing . . . 263
10.2 Dump availability management . . . 265
10.2.1 Setting up automatic full volume dump in the DFSMShsm environment . . . 266
10.2.2 Setting up automatic full-volume dump at the volume level . . . 269
10.2.3 Dumping volumes by command . . . 270
10.2.4 Interrupting and restarting automatic dump processing . . . 270
10.3 Recovering data from the DFSMShsm backup . . . 271
10.3.1 Recover a data set from a backup . . . 272
10.3.2 Restoring a volume from a full volume dump . . . 276
10.3.3 Volume recovery from incremental backup versions . . . 279
10.4 Aggregate backup and recovery support . . . 280
10.4.1 Define an aggregate group . . . 280
10.4.2 Define selection and instruction data set . . . 283
10.4.3 Defining SETSYS parameters for aggregate backup . . . 284
10.4.4 Perform an aggregate backup (ABACKUP) . . . 286
10.4.5 Perform an aggregate recover (ARECOVER). . . . 291
10.4.6 Interrupting and restarting ABARS processing . . . 295
10.5 Fast replication backup and recovery . . . 295

10.5.1 Define copy pool backup storage group 296
10.5.2 Assign storage groups to copy pool backup and copy pool construct 300
10.5.3 Perform fast replicate backup (FRBACKUP) 307
10.5.4 Perform fast replicate recover (FRRECOV) 313

Chapter 11. Maintaining your DFSMShsm environment 319
11.1 Maintaining DFSMShsm 320
11.1.1 Holding DFSMShsm functions 320
11.1.2 Releasing DFSMShsm functions 323
11.1.3 Canceling queued DFSMShsm requests 324
11.1.4 Changing DFSMShsm control parameters 325
11.1.5 Restarting DFSMShsm after an abnormal end 327
11.1.6 Expiring backup versions 328
11.1.7 Recycling tape volumes 330
11.1.8 Moving data sets to new DASD volumes 334
11.1.9 Auditing DFSMShsm 335
11.2 Journal as a large format data set 336
11.3 Maintaining SDSP data sets: Reorganization 338

Chapter 12. Control data set considerations 341
12.1 DFSMShsm journal and control data sets 342
12.1.1 How control data sets work 342
12.1.2 CDS reorganization 343
12.1.3 CDS and journal performance suggestions 349
12.1.4 CDS and journal backup 350
12.1.5 VSAM record-level sharing 356
12.1.6 RACF FACILITY classes 363
12.2 Multicluster control data sets 364
12.2.1 Multicluster performance considerations 365
12.2.2 Conversion steps 365
12.3 CDS backup procedures 367
12.3.1 Concurrent Copy 367
12.3.2 Virtual Concurrent Copy 368
12.4 CDS extended address volume 369
12.5 CDS new backup technique 369

Chapter 13. Monitoring and reporting 373
13.1 List command 374
13.2 Query command 382
13.3 Report command 390
13.4 Using the DFSMSrmm reporting tool 394
13.5 Utilities in ARCTOOLS for DFSMShsm reporting 398
13.6 Using DCOLLECT 398

Chapter 14. Problem determination 401
14.1 Problem determination documentation 402
14.1.1 Problem Determination Aid 402
14.1.2 DFSMShsm dumps 404
14.1.3 SYSLOG 407
14.1.4 JOBLOG 408
14.1.5 Activity logs 408
14.2 Problem recognition, documentation collection, and recovery 409
14.2.1 Hung situations in the DFSMShsm address space 409
14.2.2 DFSMShsm messaging 414

14.3 Data recovery scenarios 418
14.4 Auditing DFSMShsm 418
14.4.1 Introduction to the AUDIT command 419
14.4.2 Auditing copy pools 421
14.4.3 Auditing common queues 421
14.4.4 AUDIT command examples 422

Chapter 15. IBM Health Checker update 427
15.1 Health Checker function 428
15.1.1 New DFSMS health check 428
15.1.2 Upgrade and coexistence considerations 431

Chapter 16. Common DFSMShsm functions 433
16.1 Changing priorities in DFSMShsm 434
16.1.1 Priority processing 434
16.1.2 Tape Take Away in a single DFSMShsm host environment 435
16.1.3 Altering priority by using the ALTERPRI command 436

Chapter 17. Hints, tips, and suggested practices 439
17.1 Hints and tips 440
17.1.1 Use REGION=0M when you start DFSMShsm 440
17.1.2 Large format journal 440
17.1.3 CDS recovery 440
17.1.4 Delete migrated data sets without unnecessary recall 440
17.1.5 Using ARCCATGP 440
17.1.6 DFSMShsm using 3592 model E05 441
17.1.7 Experiencing a space problem during the recall process 444
17.1.8 Replacing a lost or damaged ML2 tape 444
17.1.9 ML1 pool full 444
17.1.10 Identify ML1 volumes 445
17.1.11 RECALL failures on ML1 volumes 446
17.1.12 Dump stacking and a storage group 446
17.1.13 Schedule automatic backup in multiple DFSMShsm host environments 447
17.1.14 Schedule primary space management and secondary space management . 447
17.1.15 Migration and storage group thresholds 447
17.1.16 Using INCREMENTALBACKUP(ORIGINAL) 448
17.1.17 Delete No Recall with IEFBR14 448
17.1.18 Tuning patches that are supported by DFSMShsm 448

Appendix A. Sample ACS routines 449
Sample storage class ACS routine 450
Sample management class ACS routine 455
Sample storage group ACS routine 459
Sample data class ACS routine 463

Related publications 467
IBM Redbooks 467
Other publications 467
Online resources 469
Help from IBM 469

Index 471

Notices

This information was developed for products and services offered in the U.S.A.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

DB2®
DS8000®
FlashCopy®
HyperSwap®
IBM®
IBM z™
IBM z Systems™
Magstar®
MVS™
NetView®
OS/390®
Parallel Sysplex®
RACF®
Redbooks®
Redbooks (logo) ®
Tivoli®
z Systems™
z/OS®
zEnterprise®

The following terms are trademarks of other companies:

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

Preface

DFSMShsm provides storage management to help you optimize resources, improve efficiency, direct access storage device (DASD) space, and systems availability. It works with Data Facility Storage Management Subsystem (DFSMS) closely to manage your data through data set migration and recall, and data set backup and recovery. With DFSMShsm, you can automate your storage management tasks, improving productivity by effectively managing the storage devices.

This IBM® Redbooks® publication provides technical storage specialists and storage administrators with basic DFSMShsm knowledge for implementing and customizing DFSMShsm at the IBM z/OS® V2R21 level. Hints and tips about the daily operation, monitoring, and tuning are included. Sysplex environment considerations are also included.

If you are implementing DFSMShsm for the first time, you can obtain valuable information about how to use the DFSMShsm functions. If you are experienced, you can use this publication as an update to the latest DFSMShsm functions. This book shows how to use those functions in an existing DFSMShsm installation.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Keith Winnard is the ITSO Redbooks Publications Project Leader for z/OS and related topics at the International Technical Support Organization, Poughkeepsie Center. He joined IT in 1977 and worked for various clients and business partners. He is experienced in blending traditional z/OS environments and applications with web middleware and applications. Keith is frequently a presenter on many mainframe-related topics.

Jose Gilberto Biondo is an IT Specialist in Integrated Technology Delivery, Server Systems Operations/Storage Management in IBM Brazil. He has seven years of experience in z/OS, working with storage management since 2007. Jose works mainly with IBM storage products (DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm), but he also works with other equipment manufacturer (OEM) software products. Jose's areas of expertise include installing and maintaining storage products, and process automation.

Gert Laumann is an IT Specialist in Integrated Technology Delivery, Server Systems Operations/Storage Management in IBM Denmark. He has 30 years of experience in z/OS, working with storage management since 1989. He is the team leader for the Scandinavian mainframe storage team. His experience is mostly with IBM products (DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm), but he also worked with OEM software and hardware products. His focus is automation and standardization across mainframe client platforms. He also sets up shared DASD and tape environments to mitigate data center constraints on floor space, power, and cooling. Gert is trying to achieve this same efficiency across Nordic regions by consolidating hardware in fewer data centers.

Norbert Sclumberger is an IT Architect with IBM Germany. He has over 35 years of experience in storage software and storage management for IBM and client systems, including 24 years of experience with DFSMSrmm. He developed, delivered, and taught DFSMSrmm education to clients around the world and has many tools available, such as a Tape Copy Tool to support the DFSMSrmm business. His areas of expertise include performing conversions from vendor tape management products to DFSMSrmm and new DFSMSrmm implementations. Norbert provides marketing support for DFSMSrmm, including IBM 3494 and IBM 3495 Automated Tape Libraries (ATLs), Virtual Tape Servers (VTSs), and TS7700 Virtual Engines.

Jeanne Vangsness is a DFSMS Software Technical Support Engineer specializing in the DFSMShsm product area. With IBM for 26 years, she spent 19 years working with clients and developers on the DFSMS product set. She provides feedback to DFSMShsm development in a number of areas to help improve the functionality of DFSMShsm. Jeannie co-authored several Redbooks publications, including *Implementing an Open SAN,* SG24-6116, and *z/OS 1.12 Technical Update,* SG24-7895, and provided assistance with other IBM Redbooks publications.

Chao Xing Ma is an Advisory IT Specialist in IBM China. He has five years of experience in the areas of Mainframe Technical Support and Customer Service Delivery, and client support. His areas of expertise include z/OS, IBM Parallel Sysplex®, DFSMS, SMP/E for z/OS, and Disaster Recovery solutions.

Thanks to Bob Haimowitz, Development Support Team (DST), Poughkeepsie Center, for his contributions to this project.

Thanks to the authors of the previous editions of this book.

The following individuals authored the previous edition, *IBM z/OS DFSMShsm Primer*, which was published in December 2013:

- Mary Lovelace
- Jose Gilberto Biondo
- Derek Erdmann
- Gert Laumann
- Jacqueline Tout
- Hien Truong
- Jeanne Vangsness
- Craig Vernon

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-5272-03
for IBM z/OS DFSMShsm Primer
as created or updated on June 21, 2017.

September 2015, Fourth Edition

This revision includes the following new and changed information.

New information

This edition applies to Version 2, Release 1 of Data Facility Storage Management Subsystem (DFSMS) (product number 5650-ZOS).

1

DFSMShsm overview

This chapter provides an overview of the features and functions in DFSMShsm at the z/OS V2.1 level. It introduces you to the concepts of DFSMShsm. More detailed information and implementation examples are in the remaining chapters of this book.

1.1 Introduction

DFSMShsm is a functional component of the Data Facility Storage Management Subsystem (DFSMS) family, which provides facilities for managing your storage devices. DFSMShsm ensures that space is available on your direct access storage device (DASD) volumes, so you can extend existing data sets and allocate new ones. DFSMShsm ensures that backup copies of your data sets are always available in case your working copies are lost or corrupted. It relieves you from manual storage management tasks and improves DASD use by automatically managing both space and data availability in a storage hierarchy.

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

DFSMS consists of the following five components. Their functions are described:

- DFSMSdfp: Provides storage, data, program, and device management functions through the storage management subsystem (SMS).
- DFSMSrmm: Provides tape management functions for removable media, such as virtual, and physical tape cartridges.
- DFSMSdss: Provides data movement, copy, backup, and space management functions in a batch environment.
- DFSMSoam: Provides tape hardware management, such as cartridge entry, eject, and tape configuration database (TCDB) management.
- DFSMShsm: Provides backup, recovery, migration, and space management functions with optimum automation capabilities.

DFSMShsm is a policy-driven solution to storage management, removing the requirement for batch jobs to perform backup, migration, or space retrieval functions. DFSMShsm works by rules that you can apply to manage your storage. These rules are also dynamically adjustable to allow the flexibility that is required in today's constantly changing environments. With its flexibility, you can manage your storage at the data set level, the device level, or even the device pool level. DFSMShsm provides the means to manage every data set from the time of its creation until the time that its last backup is no longer required.

In this book, we describe the major functions of DFSMShsm and how they interact with the DFSMS components, including storage management subsystem (SMS). We also provide details, different techniques, and examples to guide you to a successful DFSMShsm implementation.

1.2 Storage device hierarchy

DFSMShsm uses a hierarchy of storage devices, as shown in Figure 1-1 on page 3. The hierarchy is used in its automatic management of data, relieving users from manual storage management tasks. It categorizes storage devices into three levels:

- Level 0 volumes

  DFSMShsm managed volumes are those level 0 volumes that are managed by the DFSMShsm automatic functions. These volumes contain data sets that are directly accessible to you and the jobs that you run. Level 0 volumes and DFSMShsm managed volumes can be any DASD that is supported by DFSMShsm.

  The next two types of storage are described in detail in 1.5, "Data set lifecycle with DFSMShsm" on page 5. These volume types are managed by DFSMShsm only.

- Level 1 volumes

  When a data set is not referenced after a certain amount of time as defined by the installation, DFSMShsm moves the data set from the user data volume to a volume that is owned by DFSMShsm. These volumes are known as *migration level 1 (ML1) volumes*. These volumes are non-SMS volumes that are available for use only by DFSMShsm.

- Level 2 volumes

  When a data set resided on an ML1 volume after a determined amount of time without being referenced, DFSMShsm moves this data set to a lower-cost media that is usually high-capacity tape. These volumes are known as *migration level 2 (ML2) volumes*.

Figure 1-1 shows an overview of the DFSMShsm main functions and storage device hierarchy.

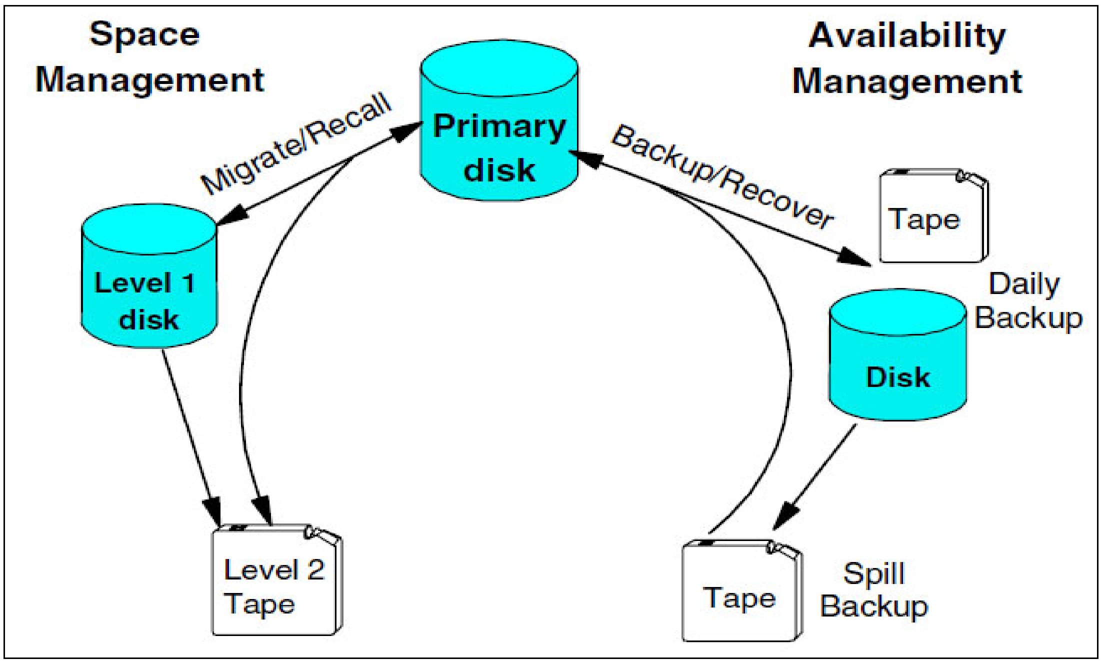

Figure 1-1 DFSMShsm main functions overview

1.3 SMS and DFSMShsm relationship

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

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 required by applications. The storage class contains availability and performance attributes, such as response time and cache requirements, for data sets.

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

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

SMS classes and storage groups can be assigned for data sets by automatic class selection (ACS). ACS routines are written by the storage administrator. They automatically assign SMS classes and groups to data sets, database data, and objects. Data allocations are processed through ACS routines. With ACS routines, you can enforce installation standards for data allocation and override user specifications for data, storage, and management classes and requests for specific DASD volumes. You can create up to four ACS routines in an SMS configuration, one for each type of SMS class and one for storage groups.

SMS and DFSMShsm work together. Using the SMS attributes specified in the management class and storage group provides space and availability management at the data set and volume level.

Figure 1-2 shows the order in which ACS routines are processed. Data becomes system managed if the storage class routine assigns a storage class to the data set. If the routine does not assign a storage class to the data set, it cannot reside on an SMS-managed volume.

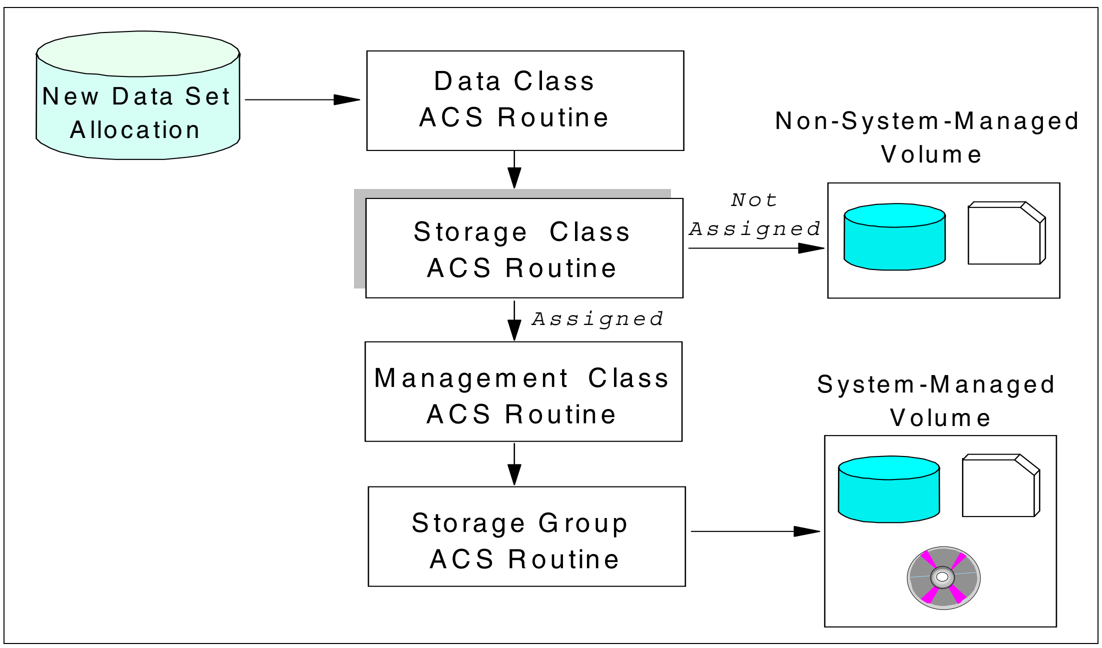

Figure 1-2 Processing order of ACS routines

SMS attributes for a user or group can be defined by IBM RACF® DFP segments (by using the **ALTUSER** or **ALTGROUP** command), simplifying ACS routines. For more information about these RACF commands, see *z/OS Security Server RACF Command Language Reference*, SA22-7687.

1.4 DFSMShsm control data sets

The DFSMShsm program requires the following data sets to support full function processing:

- Control data sets (CDSs): Migration control data set (MCDS), backup control data set (BCDS), offline control data set (OCDS), and journal data set
- Control data set and journal data set backup copies
- Problem determination aid (PDA) log data sets
- DFSMShsm logs
- Activity log data sets
- Small-data-set packing (SDSP) data sets
- System data sets

The CDSs are Virtual Storage Access Method (VSAM) key-sequenced data sets (KSDSs). The journal file is a physical sequence data set that is allocated in one extent. These data sets must not reside on SMS-managed volumes or volumes that contain other system data sets or catalogs.

The following list provides the CDSs:

- Migration control data set (MCDS):
  - Contains information about the migration environment.
  - Required for DFSMShsm to function.
- Backup control data set (BCDS):
  - Information about the backup and dump environment is stored here.
  - Only needed if you intend to use backup and dump functions.
- Offline control data set (OCDS):
  - Contains information about the backup, dump, and migration tape volumes.
  - Required if you intend to use tape devices for backup, dump, or migration volumes.
- Journal data set:
  - Records each critical update to any of the three CDSs.
  - Required for DFSMShsm CDS recovery.

Next, how DFSMShsm and SMS work together to accomplish your goals for your storage environment is explained.

1.5 Data set lifecycle with DFSMShsm

When a data set is allocated on a DASD volume that is controlled and monitored by DFSMShsm (either an SMS-managed or a non-SMS-managed volume), it goes through the following different stages of its lifecycle.

1.5.1 Data set create

During the creation of a data set, the system performs a set of basic activities. SMS ACS routines are given control and decide whether a data set is SMS-managed:

- For SMS-managed data sets, the data set is usually assigned to a management class that allows DFSMShsm to manage the volume on which it exists.
- For non-SMS-managed data sets, the storage administrator defined the non-SMS volumes to DFSMShsm so that DFSMShsm manages the volume and the data set.

These activities allow DFSMShsm to manage the volume and the data sets on those volumes. Figure 1-3 shows the flow of a data set from job control language (JCL) allocation to physical creation.

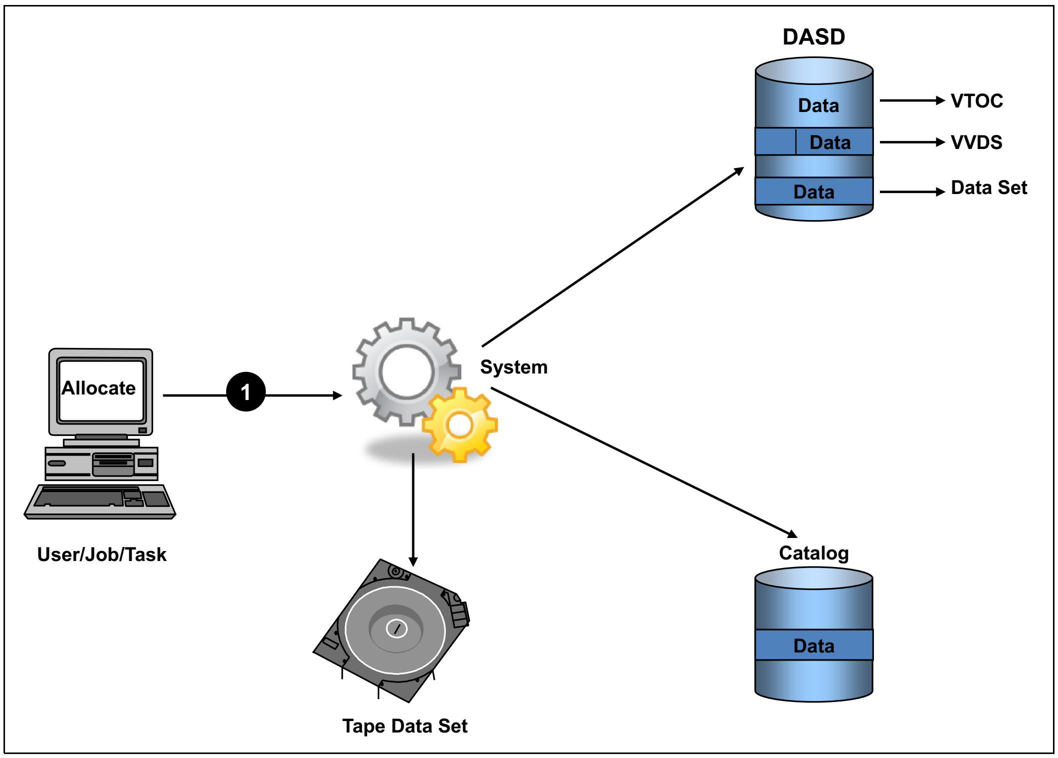

Figure 1-3 Data set creation flow

1.5.2 Data set backup

If the volumes where the data sets exist are managed by DFSMShsm (either SMS or non-SMS volumes), DFSMShsm checks whether the volume and data set are eligible for backup. Depending on how you define your backup parameters to DFSMShsm, the volume and the data set will be backed up (see Figure 1-4 on page 7).

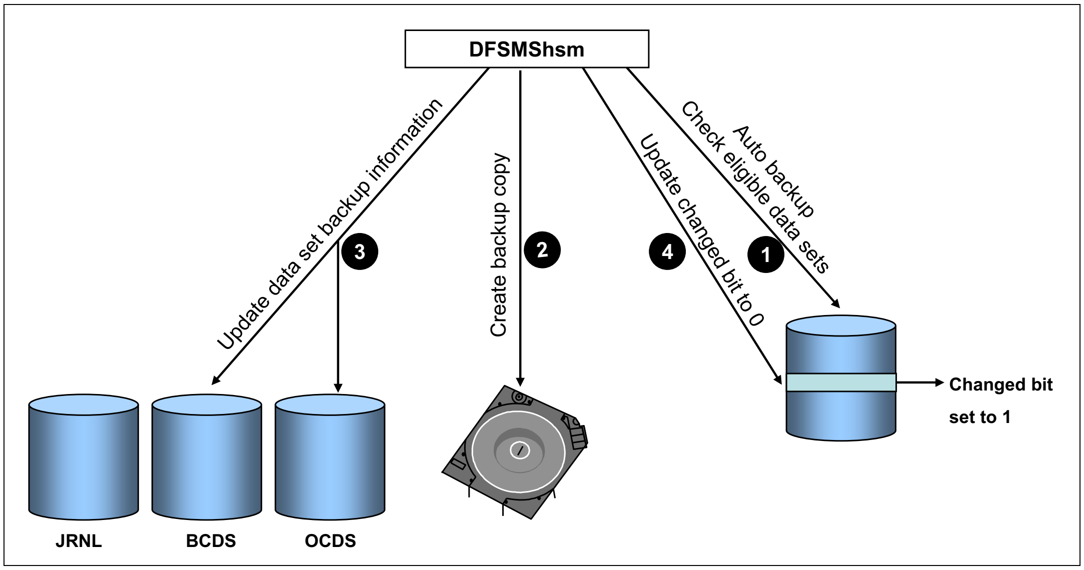

Figure 1-4 Data set backup flow

1.5.3 Data set update

The next step in the data set lifecycle is usually when the data set is updated. Whenever a user, job, or task performs output processing to a DASD data set, part of this process causes an update to the FMT1 DSCB to show that the data set was changed with the last referenced date being updated. Figure 1-5 shows the process when the data set is updated.

> **Note:** Changes to a data set usually cause a new backup of the data set to be captured, depending on your backup parameters and settings.

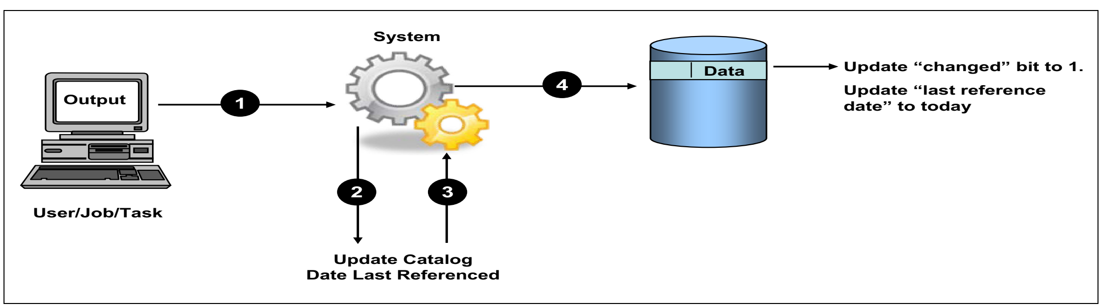

Figure 1-5 Update to data set

1.5.4 Data set migration

The next step in the data set lifecycle is based on the duration since a data set was referenced, which is determined by your storage administrator. For SMS volumes, the duration is the Primary Days parameter value in the management class. For non-SMS volumes, it is based on the value of the MIGRATE parameter when the volume is added to DFSMShsm. When the date is passed, DFSMShsm migrates the data set from the primary user volume to a lower-cost DASD volume. During this process, DFSMShsm records the updates in its CDSs and updates the volume table of contents (VTOC) and the catalog (see Figure 1-6 on page 8).

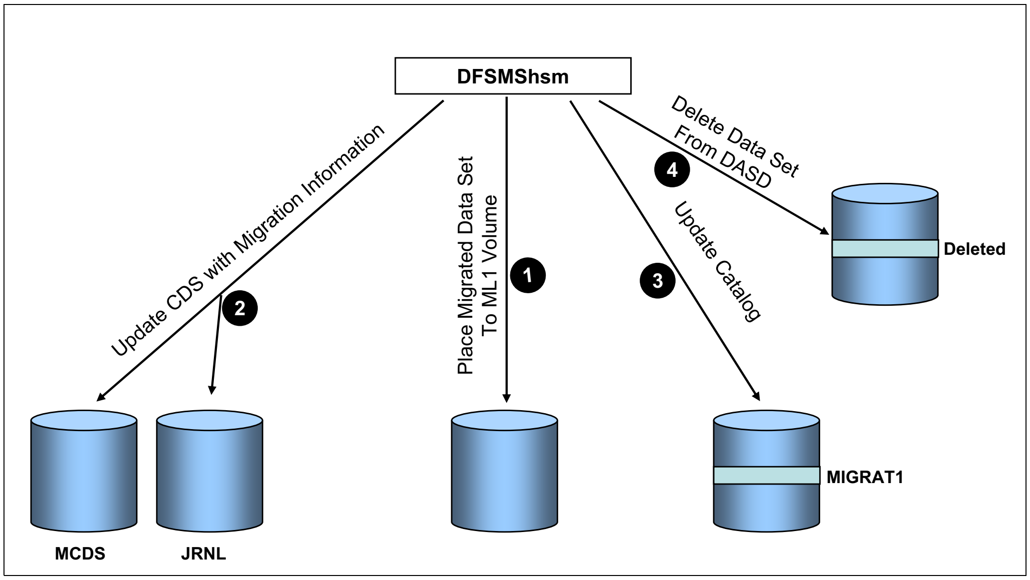

Figure 1-6   Data set migration flow

1.5.5 Data set recall

When a data set is migrated and a user needs this data set for input or output, DFSMShsm recalls it. The DFSMShsm CDSs, catalog, and VTOCs are updated to reflect these changes (see Figure 1-7).

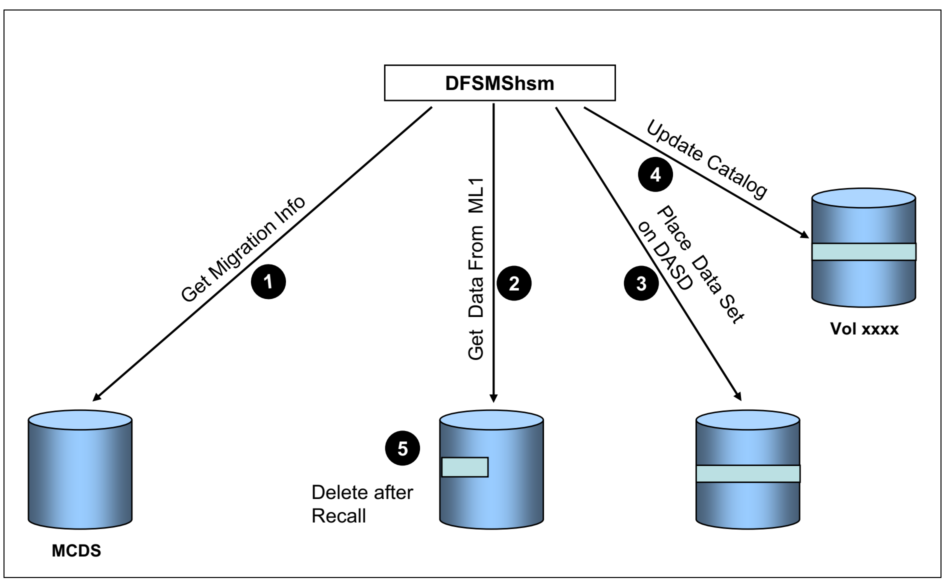

Figure 1-7   Data set recall flow

1.5.6 Data set expiration

If the user never requested a recall and the data set reached its retention period, DFSMShsm expires the data set. It performs this expiration based on “Expire non-usage”, “Expire date/days” or “Ret limit” in the management class, or “DELETEBYAGE” if the volume is a non-SMS-managed volume. During this process, the DFSMShsm CDSs, the catalog, and any VTOCs are updated, as appropriate (see Figure 1-8). Depending on the retain backup versions that are set in the storage management policies, any backups of the deleted data set are still retained.

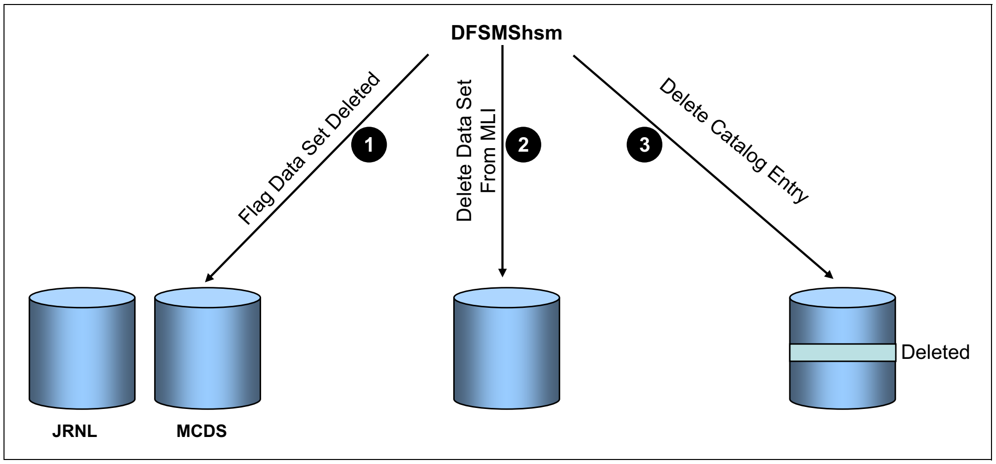

Figure 1-8 Data set expiration

1.5.7 Data set recovery

If something happens to your user data set and you set up DFSMShsm so that it backs up your data set, you are able to recover it from a backup copy (see Figure 1-9).

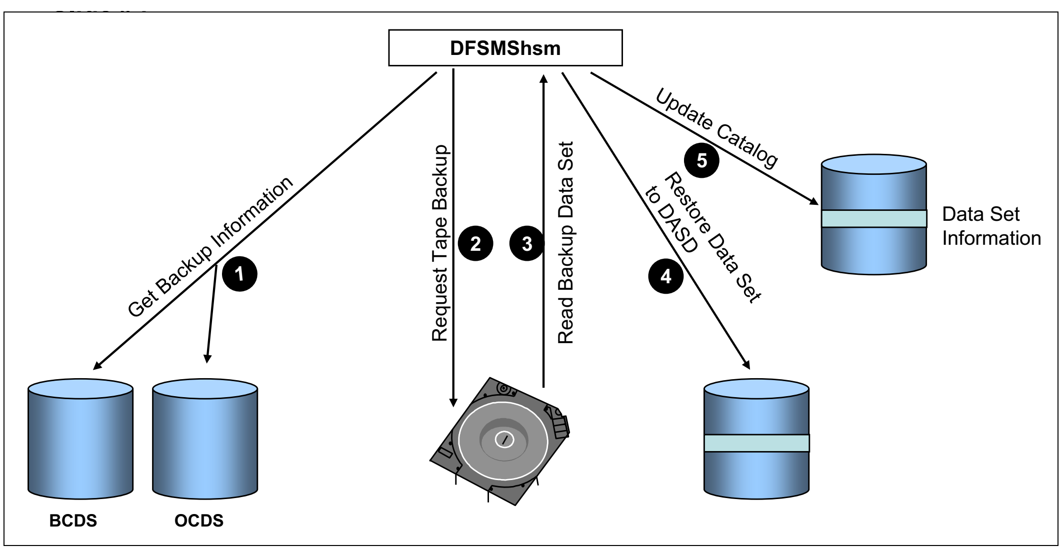

Figure 1-9 Data set recovery

1.5.8 Expiring backup versions

One of the last steps in a data set's lifecycle is deleting the backup versions of a deleted data set (see Figure 1-10). This task is not performed automatically by DFSMShsm. The storage administrator handles this task by issuing the DFSMShsm `EXPIREBV` command. This command expires old and unnecessary backup versions of deleted data sets. DFSMShsm updates its CDSs to reflect that these backup versions were deleted.

Note: When the backup version is deleted, the space where it was is no longer available and cannot be used to recover the data set.

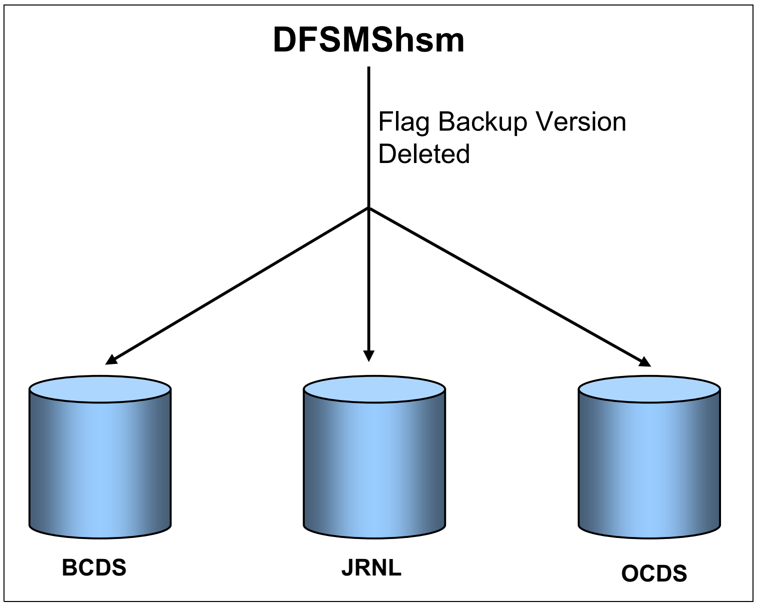

Figure 1-10 Expire backup versions

1.6 Space management

A major function of DFSMShsm is space management. With space management, you can keep DASD space available for users on a daily and hourly basis to meet the service-level objectives for your system. The purpose of space management is to manage your DASD storage efficiently so you do not have to.

Space management encompasses the following activities:

- Freeing over-allocated space through partial release
- Deleting expired data sets

- Migration of data by moving eligible data sets that were not used recently to an ML1 or ML2 volume:
  - Data movement from expensive volumes to less expensive storage is called *migration*. Data can be migrated from user volumes that users can access to ML1 volumes. The ML1 volumes must be DASD volumes. Data can also be migrated to ML2 volumes, which are usually tape volumes. Although ML2 volumes can be DASD, SMS supports only ML2 tape.
  - Recall: When the data set is needed again, DFSMShsm returns it to the user's control. This process is called *recall* and is usually processed automatically by DFSMShsm. Recall is also possible by issuing an `HRECALL` command from your Time Sharing Option (TSO) session or batch job. The SMS ACS routines ultimately determine the volume to which the data set is recalled.
- Extent reduction

Space management is achieved through the following phases:

- Primary space management
- Secondary space management
- Interval migration
- On-demand migration

This process ensures that only active data occupies space on DASD volumes. Space management parameters control how DFSMShsm makes space available on level 0 volumes. Space management is specified as a combination of parameters in the management classes and storage groups. When planning for space management, you want to strike a balance between enough available DASD space for new data set allocations and frequent recalls of migrated data sets. The management class attributes apply on a data set basis so that different data sets on the same volume can be migrated based on different criteria.

1.6.1 Recycle activity

The data that DFSMShsm stored on tapes is invalidated over time by the expiration of migrated data sets or the generation of more recent backup data sets. The *recycle* function provides the capability of moving the valid data sets out from the original tapes and consolidates the data on another tape. One tape contains all unexpired data sets and leaves the recycled tapes available for scratching.

1.6.2 Primary space management

During primary space management, DFSMShsm combines a set of specific functions to accomplish DASD usage levels that are coded in storage group (SG) definitions. For non-SMS-managed volumes, they are defined in the DFSMShsm PARMLIB member ARCCMDxx.

The following functions are performed by DFSMShsm during primary space management:

- Release unused over-allocated space of eligible data sets
- Delete expired and temporary data sets
- Migrate data sets to ML1 or directly to an ML2 volume
- Whether or not to delete or migrate rolled-off generations if the data set was a member of a generation data group (GDG)

> **Note:** All data sets that are stored in the same non-SMS volume have the same backup frequency, number of versions to keep, migration, and deletion rules.

The first functions that are started by primary space management include checking level 0 volumes for temporary and expired data sets that are eligible for deletion. DFSMShsm also releases unused and over-allocated space from eligible data sets and migrate-eligible data sets that were recalled recently, by using the Fast Subsequent Migration function, if available.

If the previous tasks were sufficient to meet the low threshold value that is coded in the SG where the volume resides, or the value that is specified in the DFSMShsm PARMLIB member ARCCMDxx, the space management ends for that volume, and the next volume is selected for processing. Otherwise, DFSMShsm continues primary space management processing.

The next steps in primary space management are selecting and migrating the largest data set that is eligible for migration in the selected volume. All of the data sets are migrated to ML1 DASD, or ML2 tape, according to management class (MC) polices or ARCCMDxx specifications. Migration processing moves to the next volume when either of these conditions is met: Volume met the low threshold utilization that is coded in the SG, or no more data sets are eligible for migration. In a non-SMS environment, all data sets that are eligible are migrated, regardless of the volume space usage.

A data set that is migrated from a non-SMS-managed volume can be recalled to an SMS-managed volume if SMS is active and if the data set meets the conditions for selecting SMS management for that data set. DFSMShsm recalls a non-SMS-managed data set to a non-SMS-managed volume only if the data set does not meet the conditions to become an SMS-managed data set, or if a command forces the recall to a non-SMS-managed volume.

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

1.6.3 Secondary space management

In secondary space management, in addition to moving data sets from ML1 volumes to ML2 volumes, DFSMShsm performs migration cleanup. These processes are performed in parallel. Migration cleanup deletes expired data sets, erases deleted data from SDSP data sets, and deletes unwanted records from the DFSMShsm MCDS.

DFSMShsm also performs these tasks:

- Schedules `TAPECOPY` commands for migration tape copy-needed (TCN) records
- Deletes expired data sets from the migration volumes
- Deletes obsolete MCDs, volume statistics records (VSRs), and DSRs during migration cleanup
- Moves data sets (under the control of the management class) from ML1 to ML2 volumes

1.6.4 Interval migration

DFSMShsm interval migration ensures that a specified amount of space is available on an hourly basis. DFSMShsm performs interval migration, as needed, for all storage groups that are eligible for interval management. In interval migration, DFSMShsm performs a space check on each DFSMShsm volume that is being managed. A volume is considered eligible for interval migration based on the AUTOMIGRATE (AM=) and THRESHOLD settings of its storage group.

When AM=`I`, the space that is used must exceed the halfway mark between high and low thresholds to make the volume eligible. When AM=`Y`, the space that is used must exceed the high threshold to make the volume eligible. DFSMShsm migrates eligible data sets to ML1 or ML2 volumes. This process continues until the volume that is managed by DFSMShsm reaches the low threshold or no more data sets are eligible.

1.6.5 On-demand migration

On-demand migration is available as an alternative to interval migration. On-demand migration performs near immediate space management on eligible SMS-managed volumes that exceed the specified volume high threshold, instead of waiting for interval migration to run at the top of each hour. On-demand migration migrates data sets to ML1 or ML2 volumes and continues to process the volume until the low threshold is reached or no eligible data sets remain.

On-demand migration is based on an event notification from SMS. You can configure your system to manage your space needs.

1.7 Availability management

The other major function that DFSMShsm provides is availability management. Availability management enables a user to recover a lost or damaged data set easily. A storage administrator can recover a damaged volume easily at the current level.

Availability management (backup) automatically and periodically performs the following backups:

- Back up CDSs and journal DFSMShsm
- Back up data sets from DASD to tape volumes
- Back up changed data sets on DASD volumes to other DASD or tape volumes
- Back up data sets of an application to tape volumes so they can be taken offsite for recovery

By performing these functions, automatic backup can help in these ways:

- Prevent users from accidentally losing or incorrectly changing data sets
- Recover data after a volume is lost due to a hardware failure
- Protect against loss of data due to a disaster
- Ensure that critical data is retained

  New data retention laws cause an increased need for backups of critical data.

The following functions provide valid backup or dump copies so that you can recover:

- Backup
- Dump
- Aggregate backup and recovery support (ABARS)
- Fast replication backup

1.7.1 Backup and recovery

Different methods are available to back up your data. DFSMShsm provides an automatic backup function, inline backup, fast replication backup, and aggregate backup.

Automatic backup

The *autobackup* function is a data-set-level function. It relies on the guaranteed backup frequency attribute of the storage group for each data set to determine whether to copy the data set. After a data set is backed up, this function uses the data set's management class attributes to decide how the data set is treated for the creation and retention of backup versions (how many backup versions are kept and how long they are kept).

In a non-SMS environment, the backup frequency is defined for the entire volume. All data sets that changed since the last backup process ran are eligible for a new backup.

Recovery restores data sets from the daily backup volumes or the spill backup volumes to the level 0 volumes. Recovery can be performed only by command, for individual data sets or complete volumes.

Inline backup

With the *inline backup* function, you can back up data sets in a batch or online environment. You can back up data sets in the middle of a job in a batch environment, or directly through Time Sharing Option (TSO). Inline backup writes the backup version on an ML1 volume or to a tape volume. If the data set is moved to an ML1 volume, you can minimize unplanned tape mounts. In that case, the backup version is later moved to the backup volume during automatic backup or by command.

ABARS

Aggregate backup and recovery support (ABARS) is a function of DFSMShsm that is designed for use in disaster recovery. ABARS facilitates a point-in-time backup of a collection of related data in a consistent manner. This group of related data is defined to ABARS as an aggregate. An aggregate is user-defined and usually a collection of data sets that are related, such as all of the specific data sets required for an application (for example, payroll). ABARS backs up the data directly from DASD (either SMS-managed or non-SMS-managed), DFSMShsm-owned ML1 DASD, or DFSMShsm-owned ML2 tape without needing intermediate staging capacity. The backup copies are created in a device-independent format.

ABARS can be used also for moving applications across non-sharing systems. If ABARS is used as a disaster recovery tool, at the recovery site, it is used to recover the data sets and allocates user-defined empty data sets that the client requires for their operating environment.

The aggregate backup and aggregate recovery functions provide the capability to back up and recover a user-defined group of data sets. The user-defined group of data sets can belong to an application or any combination of data sets that you want treated as a separate entity. ABARS treats SMS and non-SMS data sets identically.

With the aggregate backup and aggregate recovery functions, you can perform the following actions:

- Define the components of an aggregate
- Back up data sets by aggregate
- Recover data sets by aggregate
- Recover a single data set
- Duplicate your aggregates at a remote site
- Resume business at a remote location, if necessary

Fast replication

Fast replication is a function that uses volume-level fast replication to create backup versions for sets of storage groups. You can define a set of storage groups with the SMS copy pool construct. Fast replication target volumes contain the fast replication backup copies of volumes that are managed by DFSMShsm. Fast replication target volumes are defined with the SMS copy pool backup storage group type.

The DFSMShsm **FRBACKUP** command creates a fast replication backup version for each volume in every storage group that is defined within a copy pool. Volumes that have a fast replication backup version can be recovered either individually or at the copy pool level with the DFSMShsm **FRRECOV** command. This function enables the backup and recovery of a large set of volumes to occur within a short time frame. For more information about fast replication and its requirements, see the *DFSMShsm Fast Replication Technical Guide,* SG24-7069.

1.7.2 Dump

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

To use the dump function, a dump class must be assigned to the required storage groups. Next, you need to define the dump class to DFSMShsm if it is not already set. The dump class definition includes a brief description, the dump date or day of the week to be dumped, the retention period, and the expiration date.

1.8 Tape processing

DFSMShsm can use tape for backup, migration, dump, and ABARS. When DFSMShsm maintains its tape environment, it performs tape copy and recycle of backup, dump, ABARS, or ML2 volumes. You implement a DFSMShsm tape processing environment by specifying **SETSYS** commands in the DFSMShsm PARMLIB member ARCCMDxx.

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

DFSMShsm duplexing

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

When DFSMShsm requests a scratch tape for output, the tape management system returns a scratch tape that is based on the type of tape that is requested and the scratch pool that is assigned. You can define a global scratch pool or a specific scratch pool. A *global scratch pool* is a repository of empty scratch tapes for use by any user. A *specific scratch pool* is a repository of empty scratch tapes whose use is restricted to a specific user or set of users.

Global scratch pools are recommended because mount requests can be responded to more quickly and easily than when tapes are in a specific scratch pool. By using a global scratch pool, you can take advantage of automatic cartridge loaders easily, reducing the tape mount wait time. Global scratch pools also enable the use of a tape management product, such as DFSMSrmm.

DFSMSrmm and DFSMShsm interaction

Interaction between DFSMSrmm and DFSMShsm occurs during the whole tape lifecycle. SETSYS commands that are added in ARCCMDxx specify how DFSMShsm processes the tapes as they enter the scratch pool, are inventoried as active data by DFSMShsm, are recycled for backup and migration tapes, and are returned to the scratch pool.

DFSMShsm is responsible for managing all of its tape retention, and lets the tape management system know when the tapes can be returned to the scratch pool. If your system does not use DFSMSrmm, DFSMShsm communicates with your tape management program through the exit ARCTVEXT.

For more information about DFSMSrmm, see *DFSMSrmm Primer,* SG24-5983.

DFSMShsm records its information about the content on the tapes that it uses in the DFSMShsm offline control data set (OCDS). It stores the information in a control data set (CDS) record that is called the tape table of contents (TTOC).

2

Planning and reviewing your environment

In many ways, planning is the most important phase of the DFSMShsm implementation. This chapter identifies key areas to address during planning.

If you are new to DFSMShsm, you can use this information as input to set up your environment based on your requirements. If you are an existing DFSMShsm user, you can use this information to review your current environment and perhaps change the environment to reflect the current features and functions.

2.1 Considerations

Many considerations are involved as you prepare to set up your DFSMShsm environment. If you have an existing DFSMShsm environment, many of these items are valuable to review periodically because your configuration and environment changes.

Consider the following key questions and information:

- How much data do you want DFSMShsm to manage?
  - How much available space is needed throughout the day for applications to run successfully?
  - How often is the data referenced again after creation?
- How do you want DFSMShsm to manage the data?
  - Manage the space available on volumes?

    Use the small-data-set packing (SDSP) facility?
  - Manage your backup and recovery data?
  - Will you use encryption?
  - Expire the migration and backup copies of data sets?
- Data retention:
  - How long do you need to keep unreferenced data?
  - How much data is created that has lengthy retention requirements?
  - If you have lengthy retention requirements, will you use high capacity tape to store this data?
- Disaster recovery:
  - What requirements do you have to recover critical data in a disaster?
  - Will you employ the DFSMShsm duplexing function?
  - Will you use encryption?
- Is your environment storage management subsystem (SMS) or non-SMS, or a combination?
- Security aspects

  Will you use RACF or anther form of a security product?
- Interactions with other Data Facility Storage Management Subsystem (DFSMS) products:
  - Will you use cross memory to invoke DSS for data movement?
  - Object access method (OAM)
  - RMM (or tape management system) for managing tape volumes
- Preparing for diagnostic procedures

  Problem determination aid (PDA) files, log files, and dumps
- Storage requirements for DFSMShsm to manage your data:
  - Volumes for the control data sets (CDSs) and journal
  - Extra direct access storage device (DASD) for migration level 1 (ML1) volumes for employing an ML1 policy
  - Extra tapes and tape drives to handle DFSMShsm backup, migration, and full volume dump data

2.2 Determine how you want DFSMShsm to manage your data

DFSMShsm provides functions for the following capabilities to manage data:

- **Space management**: Space management is the function of DFSMShsm to keep DASD space available for users to meet the service-level objectives for your system. The purpose of space management is to manage your DASD storage efficiently. Space management automatically and periodically performs the following functions:
  - Moves low-activity data sets from user-accessible volumes to DFSMShsm volumes.
  - Reduces the space that is occupied by data on both the user-accessible volumes and the DFSMShsm volumes.
- **Availability management**: DFSMShsm backs up your data either automatically or by command to ensure availability in the 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 onsite as protection from media damage, or offsite as protection from site damage. Disaster backup and recovery are also provided for user-defined groups of data sets (*aggregates*) so that critical applications can be restored at the same location or an offsite location.

Data can be categorized in the following manner:

- System data: All system-related data that the system requires for its operation.
- Databases: Each database management system has unique data sets and facilities to support its online environment. These differences change the recommended storage management procedures. Database data has diverse space, performance, and availability requirements; however, dividing your database data into categories helps you identify the required storage management subsystem (SMS) services and implement a staged migration to system-managed storage.
- Data that is created by batch jobs (generation data groups (GDGs) and Virtual Storage Access Method (VSAM) data sets).
- Data that is used by Time Sharing Option (TSO) users (user profiles and user job control language (JCL) files).

2.2.1 Planning for the space management process

By using space management features, you allow your data to be migrated by DFSMShsm from primary volumes (user volumes that are also known as *L0 volumes*) to DFSMShsm volumes, either ML1 (L1) or ML2 volumes (L2).

Consider the following factors for the space management process:

- You want to migrate your data: Identify the data that can be migrated based on how long the data is inactive and not referenced. In general, system data and production databases must not be migrated because these types of files are referenced fairly frequently. Depending on your own service level agreements (SLAs), you develop policies to migrate data based on those needs and how often the data is referenced after it is created:
  - TSO data sets (data that is owned by TSO users or application development) are usually smaller with lower I/O activity than production data sets. These data sets might not always have predictable access or update patterns. Migration policies for these data sets can allow them to reside on user volumes longer.
  - Batch data is data that is processed regularly, usually as part of a production cycle. Most of these data sets are usually sequential data sets that are members of GDGs. You develop specific management class attributes for handling GDG roll off processing, either delete or migrate.
- Length of time to keep the migrated data: This factor depends on your installation requirement. Certain data might need to be retained longer than other data. You develop specific management class attributes based on these needs.

  For non SMS-managed volumes, this value is determined when you add the volume to the DFSMShsm configuration.
- Time of day to schedule space management: Ideally, the migration process starts and completes before the batch window because the migration generates more free space for the batch jobs to use. The length of the migration window depends on a few factors:
  - How much data is migrated in a certain day?
  - How much of this data is migrated to DASD or tape?
  - If the data is being migrated to tape, how many tape drives are available for processing?

In a normal environment, data is migrated first to ML1 volumes, which are a pool of non-SMS-managed volumes that are dedicated for the use of DFSMShsm. If a data set continues to be unused, it is eventually migrated from an ML1 to an ML2 volume, which is usually high capacity tape. In certain cases, large data sets are often migrated straight to ML2 tape volumes.

You control the automatic migration parameters. As you monitor your environment to ensure automated functions complete on time, you adjust your environment.

2.2.2 Planning for availability management

Backing up your data is vital to protect your information. By using the backup functions, you allow DFSMShsm to back up your data based on the parameters that you define.

Consider the following factors:

- **Automatic backup**: You can configure DFSMShsm to back up your data for you. During the automatic backup process, DFSMShsm first backs up the CDSs and journal.

  If you decide that you do not want to use the automatic backup process, ensure that you back up the CDSs and journal manually so that you can recover your DFSMShsm environment, if necessary.

- **Time of day to schedule the backup and for how long**: Choose the backup window when the system is the least active so that DFSMShsm is more likely to back up data sets that are normally in use.

  The length of the backup window depends on the number of data sets that are modified in a certain day and the number of tape drives that are available for the DFSMShsm automatic backup function. You control the automatic backup parameters to best meet your needs.

- **Backup method (standard or Concurrent Copy)**: Concurrent Copy is a combined hardware, Licensed Internal Code (LIC), and software system management solution that creates data dumps or copies while user processing continues. When DFSMShsm uses Concurrent Copy to back up the data sets, the time during which system functions are suspended is dramatically reduced. DFSMShsm reserves the data sets only for the time that it takes to initialize a Concurrent Copy session.

  Use of Concurrent Copy for data set backup is justified only if the data set is database-related and significant value exists in serializing the data set for the shortest possible duration.

- **Data backup frequency and number of backup copies**: You need to decide how frequently to back up the data and whether to back it up even if it did not change. How many backup copies do you want of your data sets? How long do you want to keep the backup copies after the data sets are deleted? The correct answers to these questions can help ensure that data sets can be recovered.

Space saving with space management and backup

Part of DFSMShsm space management and backup management allows DFSMShsm to consolidate data while it migrates and backs up the data sets. Both functions perform cleanup activities so that space is available for migration and backup data.

Consider implementing the following functions to save space:

- Partitioned data set (PDS) compression: For PDS data sets, the migrate and recall result in freeing embedded free space (compression), and the user information in the data-set-directory user field is retained.
- Release of over-allocated space: DFSMShsm can release unused over-allocated space during its space management process.
- Extent reduction: The migrate and recall result in reducing the number of extents for that data set.
- Reblocking: During recall and recovery, the process of reblocking user data sets changes the number of records in a physical block and therefore uses the space on the DASD user volume more efficiently. Data movement that uses DFSMShsm reblocks physical sequential data sets. Data movement that uses DFSMSdss reblocks physical sequential and PDSs.

Use these factors to help you to determine the SMS attributes to use as you create your automatic class selection (ACS) routines.

2.3 Tape duplexing

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

If you intend to use duplex migration or to back up tape volumes, you need twice as many available tape drives for the use of DFSMShsm. DFSMShsm writes to two output devices in parallel for each function. This function is specified in your ARCCMDxx startup PARMLIB member.

2.4 Data encryption

Data encryption is an important tool for protecting against the possible misuse of confidential information that can occur if tapes are lost or stolen. Unless the possessor of the tape has the required key, any encrypted data on the tape remains confidential and is unreadable. Therefore, consider securing tape data through encryption in your overall security plan.

You can secure your tape data with either *tape device encryption* (with an encryption-capable tape drive) or through *host-based encryption,* that is, by requesting that DFSMSdss encrypt the data before writing it to the tape volume. In general, you use one method, not both.

When you choose a method, consider the following factors:

- Use tape device encryption if your installation includes one or more encryption-capable tape drives. Here, you specify by data class the data that is to be encrypted when stored on the tape drives.
- Use host-based encryption if you do not have an encryption-capable tape drive. You can encrypt tape backups through the host-based encryption method.

2.4.1 zEDC hardware compression

IBM z™ Systems Enterprise Data Compression (zEDC) for z/OS hardware compression was introduced with EC12 and BC12 servers. zEDC hardware compression requires I/O compression adapters and a licensed feature (Feature Code (FC) 0420) on the server side and z/OS V2.1 (plus PTFs) and the zEDC Express for z/OS feature on the operational system side.

zEDC compression is an effective hardware compression feature for queued sequential access method (QSAM) and basic sequential access method (BSAM) that can replace tailored compression. From the DFSMSdss perspective, support for zEDC compression was added for QSAM and BSAM, also. DFSMShsm will support zEDC compression on disk and tape for backup, migration, and dump data if DFSMSdss is the data mover.

Activation of zEDC hardware compression in DFSMShsm, in addition to the prerequisites, requires zEDC compression to be enabled in the IGDSMSxx member in DFSMS and PARMLIB activation in DFSMShsm. For DFSMShsm migration and backup, zEDC compression can be activated by function (backup on disk, backup on tape, migration to disk, or migration to tape), or for all of these functions in one **SETSYS** command. Activation of zEDC compression on the DFSMShsm DUMP function can be activated with a new **ZCOMPRESS** keyword on the dump class.

For more information, see the *IBM zEnterprise BC12 Technical Guide*, SG24-8138, *Reduce Storage Occupancy and Increase Operations Efficiency with IBM zEnterprise Data Compression*, SG24-8259, and *zEDC Compression: DFSMShsm Sample Implementation*, REDP-5158.

2.5 Using small-data-set packing

The small-data-set packing (SDSP) data set facility of DFSMShsm allows DFSMShsm to migrate small user data sets from user volumes and store them as records on an ML1 migration volume that is known as an *SDSP volume*. This ML1 volume is made up of one large VSAM key-sequenced data set (KSDS). The small user data sets are stored in the VSAM data set as a record, which takes up less space on a DASD volume.

SDSP volumes offer the following advantages:

- Reduced fragmentation of an ML1 volume.
- Reduced use of space in the volume table of contents (VTOC). Because the small-user data sets are stored as records, ML1 VTOCs are not filled with an entry for each small user data set that is in an SDSP.
- Better use of space on the ML1 volumes because the small data sets are in the SDSP VSAM KSDS as a record. A single record within a VSAM KSDS takes up less space than a data set that is stored directly on a volume.

SDSP data sets might require periodic reorganization just like any other VSAM KSDS. You can reduce the need to reorganize SDSP data sets by enabling the control area (CA) reclaim function for them.

Although one SDSP data set can be used for each concurrent migration task, certain DFSMShsm activities have a higher usage priority for SDSP data sets, such as the following activities:

- Recall processing
- Aggregate backup and recovery support (ABARS) processing
- FREEVOL processing
- AUDIT MEDIACONTROLS processing
- Automatic secondary space management processing

It is important to plan the number of SDSP data sets in relation to the number of concurrent migration tasks and the amount of processing that is performed by functions with a higher usage priority for the SDSP data sets.

Because of their higher usage priority, any of these activities can gain control of your SDSP data sets and leave you with fewer than the expected number of SDSP data sets for migration. When an activity with a higher usage priority for SDSP data sets has or requests an SDSP data set, that SDSP data set is no longer a candidate for migration.

To begin using the SDSP facility, you first need to perform the following tasks:

- Define the size of a small user data set.
- Allocate SDSP data sets.
- Specify the **`SDSP`** parameter on the `ADDVOL` statement.

For more information, see the *DFSMShsm Implementation and Customization Guide,* SC35-0418.

2.6 Expired data sets and tape recycle

The following considerations relate to expired data sets and tape recycling:

- Expiring backup versions: This expiration is accomplished through the **EXPIREBV** command. The **DFSMShsm** command deletes unwanted backup and expired ABARS versions of SMS-managed and non-SMS-managed data sets from storage that is owned by DFSMShsm. This command is a long-running command so most clients are selective when and how often they run this command.
- Recycle process: The RECYCLE function is a DFSMShsm method of reclaiming tape media capacity. As time passes, data on your backup and ML2 tape volumes becomes invalid. Migrated data sets expire or are marked for deletion. Backup versions roll off or are marked for deletion. To consolidate the valid data on to fewer tapes, you can use the DFSMShsm recycle process. The recycle function is considered by DFSMShsm to be a long-running task and therefore is limited to one command in execution per logical partition (LPAR). Because this function processes offline media, run this process outside of normal batch windows, either during the day or on weekends.

2.7 SMS classes and ACS routines

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

The goals for system-managed storage are listed:

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

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

2.8 Security aspects in the DFSMShsm environment

How do you plan on securing DFSMShsm data and the DFSMShsm commands? The use of a security product, such as RACF or an equivalent, simplifies security for the storage administrator in protecting DFSMShsm resources.

Consider the following factors about security:

- Protect all CDSs with resource protection, such as RACF, from being updated by unauthorized programs and unauthorized personnel.
- DFSMShsm and the DFSMShsm cross-memory mode started tasks must be added to the started procedures table in RACF (or the equivalent table if another security product is used).
- Consider protecting the DFSMShsm commands in the same manner as any other resources that are restricted to only storage administrators.
- Because DFSMShsm manages its own data set resources on DASD and tape, protection from unauthorized access to DFSMShsm resources is an important consideration.

  The following DFSMShsm resources must be protected from unauthorized programs and unauthorized personnel:

  – DFSMShsm data sets:
    - CDSs
    - Journal
    - Problem determination aid (PDA) trace data sets
    - Logs
    - CDS backup versions
    - SDSP data sets
    - Migrated data sets
    - Backed-up data sets
    - ABARS SYSIN data sets
    - ABARS FILTERDD data sets
    - ABARS RESTART data sets
    - ABARS access method services (IDCAMS) data sets

  – DFSMShsm tapes:
    - ML2 migration tapes
    - Incremental backup tapes
    - Dump tapes
    - TAPECOPY tapes
    - ABARS tapes

2.9 Invoking DSS to move data

In order to maximize throughput for backup, DFSMShsm CDS backup, dump, migration, and recovery function, you can use the cross-memory mode feature of DFSMSdss. The **SETSYS DSSXMMODE** command allows users to specify the functions that invoke DFSMSdss in cross-memory mode as opposed to having DFSMSdss loaded in the DFSMShsm address space.

The **SETSYS DSSXMMODE** command can be issued only from the ARCCMDxx PARMLIB member. It does not affect the FRBACKUP and FRRECOV use of the DFSMSdss cross-memory interface.

Note: If you plan to use FRBACKUP DUMP to tape and FRRECOV from tape operations, the recommendation is that the DUMP and RECOVER functions use the DFSMSdss cross-memory mode to help prevent out-of-storage abends.

To implement this feature, you need to set up all security aspects for all started tasks that are involved in the same way to set up DFSMShsm STC.

2.10 Interactions with object access method

The OAM is a component of DFSMSdfp, the base of the SMS of DFSMS. The OAM uses the concepts of system-managed storage, which are introduced by SMS, to manage, maintain, and verify tape volumes and tape libraries within a tape storage environment.

You need to consider the setup that is needed for DFSMShsm process tape for migration and backup. The focus is on automatic tape management.

In general, a *tape library* is a set of tape volumes and the set of tape drives where those volumes can be mounted. The relationship between tape drives and tape volumes is exclusive. A tape volume that resides in a library (*library-resident tape volume*) can be mounted only on a tape drive that is contained in that library (*library-resident tape drive*). A library-resident tape drive can be used only to mount a tape volume that resides in the same library. A tape library can consist of one or more tape systems.

When a volume is entered into a tape library, it is assigned to a tape storage group. A tape library can contain volumes from multiple storage groups, and a storage group can reside in up to eight libraries.

Tape automation provides satisfactory solutions for many of the problems that occur when tape library storage requires human intervention. Mount times are reduced from minutes to seconds. The number of lost, misfiled, or damaged tapes decreases. Security is enhanced because the tape library hardware and tape cartridges can be kept in a secure area.

The following considerations relate to implementing DFSMShsm functions in an SMS-managed tape library:

- Determine the tape functions that you want to process in a tape library: You must decide which DFSMShsm functions to process in a tape library. Each DFSMShsm function uses a unique tape data set name. An ACS routine can recognize the functions that you want to process in an SMS-managed tape library by the data set names.

  When you consider which functions to process in a tape library, think in terms of space-management processing, availability-management processing, or CDS backup processing as indicated in the following implementation scenarios. The migration and backup conversions are implemented differently. The implementations contrast each other because, for migration, existing tapes are inserted into the library and for backup, scratch tapes are inserted into the library and the existing backup tapes are left in shelf storage.

- Set up a global scratch pool: When a DFSMShsm output function fills a tape, it requests another tape to continue output processing. Tapes are obtained from a scratch pool. The scratch pool can be either a global scratch pool or a specific scratch pool:

  – A *global scratch pool* is a repository of empty tapes for use by anyone. The tape volumes are not individually known by DFSMShsm while they are members of the scratch pool. When a scratch tape is mounted and written to by DFSMShsm, it becomes a private tape and is removed from the scratch pool. When tapes that are used by DFSMShsm no longer contain valid data, they are returned to the global scratch pool for use by anyone and DFSMShsm removes all knowledge of the existence of them.

  – A *specific scratch pool* is a repository of empty tapes that are restricted for use by a specific user or set of users. When DFSMShsm is in a specific scratch pool environment, each empty tape and each used tape is known to DFSMShsm as a result of being added to the scratch pool, generally by the **ADDVOL** command. These tapes can be used by DFSMShsm only. The key characteristic of a specific scratch pool is that when a DFSMShsm tape becomes void of data, the tape is not returned to the global scratch pool but it is retained by DFSMShsm in the specific scratch pool for reuse by DFSMShsm.

  Global scratch pools are recommended because mount requests can be responded to more quickly and more often than when tapes reside in a specific scratch pool. The use of a global scratch pool enables the easy exploitation of cartridge loaders (including cartridge loaders in tape-library-resident devices) and works well with tape management systems, such as DFSMSrmm.

- Protecting tapes: Tape protection helps ensure that only DFSMShsm is allowed to access DFSMShsm tapes. The implementation method that is used depends on the security product that is installed.
- Tape duplexing: The DFSMShsm duplex function allows a primary tape and copy tape to be created concurrently. The primary tape can be kept onsite for recovery and the copy tape can be kept at a secure offsite location. This method is the preferred method of creating copy tapes and needs to be used instead of the **TAPECOPY** command.

2.11 Interaction with the DFSMSrmm tape management system

DFSMSrmm provides enhanced management for tape volumes that DFSMShsm uses for each of its tape functions. Precisely how the two products work together depends on how you use each of them in your installation.

DFSMShsm can tell DFSMSrmm when it no longer requires a tape volume or when a tape volume changes status. When DFSMShsm is using DFSMSrmm, it cannot mistakenly overwrite one of its own tape volumes if an operator mounts a tape volume in response to a request for a non-specific tape volume. DFSMShsm uses the DFSMSrmm EDGTVEXT interface to maintain the correct volume status. DFSMSrmm provides a programming interface so you can use the DFSMShsm exit.

2.12 Preparation for problem diagnostic determination

Often, when problems occur, sufficient data for problem determination is not readily available. IBM technical support requires different data, depending on the type of problem that is encountered. The following items are often used for debugging:

- DFSMShsm activity logs
- DFSMShsm joglog
- System log
- DFSMShsm PDA traces
- DFSMShsm journal records
- System Measurement Facility (SMF) records
- Any possible system dumps that were captured at the time

Consider the following factors when you prepare for problem diagnostic determination:

- Ensure that the PDA log is turned on and being collected.

  Capture and archive the contents of PDA files when they are swapped. For more information, see the following link:

  http://www.ibm.com/support/docview.wss?uid=isg3T1012687

- Ensure that you configured DFSMShsm for logging and log files are allocated.

  If you are concerned about DASD space and you must decide between having DFSMShsm log files or DFSMShsm PDA files, it is recommended to choose capturing the DFSMShsm PDA. Information that is written to the log files is also captured in the DFSMShsm PDA trace. Therefore, the PDA trace provides the logging information.

- Ensure that DFSMShsm is configured correctly to capture SAN Volume Controller dumps if an abend occurs.
- Ensure that your z/OS system is able to capture a dump that is generated by DFSMShsm.

3

Getting started

In this chapter, we describe how to set up your DFSMShsm environment by using the samples that are provided in `SYS1.SAMPLIB` as a starting point. These samples are referred to in the DFSMShsm manuals as the “*DFSMShsm starter set*”. Detailed information about defining DFSMShsm to z/OS and defining and maintaining its control data sets (CDSs) and accompanying data is in the *DFSMShsm Implementation and Customization Guide,* SC35-0418.

The complete details of all of the possible parameters that can be used to tailor DFSMShsm are in *z/OS DFSMSdfp Storage Administration Reference,* SC26-7402.

We describe several of the decisions that you need to make before you begin a DFSMShsm configuration. In this chapter, we describe setting up the required procedures and PARMLIB members for the started task. We then explain allocating the required data sets to enable DFSMShsm to start.

We also describe managing interactions with your DFSMShsm address spaces. For example, we describe starting and stopping the DFSMShsm address spaces and where DFSMShsm fits in your automation hierarchy. We also describe ways of passing commands to the DFSMShsm address space and mention the built-in DFSMShsm recovery ability.

3.1 DFSMShsm components

A DFSMShsm environment consists of one or more DFSMShsm address spaces. DFSMShsm can be run only as a started task and only one DFSMShsm can exist.

Each address space shares access to a set of up to three CDSs and a journal:

- Migration control data set (MCDS)
- Backup control data set (BCDS)
- Offline control data set (OCDS)

In addition to the CDSs, each DFSMShsm address space can have two pairs of allocated log data sets:

- ARCLOGX and ARCLOGY

  This pair is used to log activity (as it occurs) in the DFSMShsm address space, such as the DFSMShsm actions. Each data set of this pair must be allocated on the same volume. DFSMShsm uses these data sets as a circular log. DFSMShsm always writes to the ARCLOGX data set. When the ARCLOGX data set fills, DFSMShsm automatically renames the ARCLOGX to ARCLOGY and the ARCLOGY to ARCLOGX and continues writing to the ARCLOGX data set.

- ARCPPOX and ARCPDOY

  This pair is used to log information that is required only for problem determination. These data sets are used in the same way as the ARCLOG data sets as a circular log.

DFSMShsm can also manage several types of media:

- Migration level one (ML1): ML1 is always direct access storage device (DASD). The DASD must not be managed by the storage management subsystem (SMS). The DASD must be dedicated to DFSMShsm use.
- Migration level two (ML2): ML2 is almost always tape but can be DASD. ML2 tape volumes can reside in SMS-managed tape libraries or as non-SMS volumes. If ML2 is DASD, the DASD must not be SMS-managed.
- Backup volumes: DFSMShsm can use DASD or tape backup volumes or a combination. As with migration volumes, DASD backup volumes must not be SMS-managed.
- Dump volumes: Dump volumes are always tape volumes.

3.2 DFSMShsm starter set

In the following chapters of this book, we describe in detail the parameters that are required to implement functions and housekeeping tasks that are automated by DFSMShsm. This section concentrates on the basic building blocks of a DFSMShsm system.

We describe the use of the DFSMShsm starter set, which is a provided set of sample job control language (JCL) and code. If you are installing DFSMShsm for the first time, the starter set provides useful samples. However, like all samples, the data that is supplied needs to be tailored for your site requirements. If you are reviewing an existing configuration, the starter set might contain information about new defaults and changed functions.

DFSMShsm provides two members with sample code and JCL. These members are delivered in SAMPLIB.

The first member, ARCSTRST. ARCSTRST, is JCL to create and populate a data set called *HSM.SAMPLE.CNTL*.

The second member, ARCTOOLS. ARCTOOLS, is JCL to create and populate a number of data sets:

- The first data set is HSM.SAMPLE.TOOL. This data set contains samples that you can use for both capacity planning and creating multicluster DFSMShsm CDSs.
- The second data set that is allocated by ARCTOOLS is HSM.ABARUTIL.JCL. This data set contains samples that can be used when you implement aggregate backup and recovery support (ABARs) processing.
- The next data set that is allocated by ARCTOOLS is HSM.ABARUTIL.DOCS.
- The last set is HSM.ABARUTIL.PROCLIB, which contains JCL for a sample ABARS address space.

In this chapter, we concentrate on the contents that are created by the ARCSTRST member in HSM.SAMPLE.CNTL and how the contents can be used to create a new DFSMShsm environment or upgrade a current DFSMShsm environment.

3.2.1 Contents of HSM.SAMPLE.CNTL

When you run the ARCTOOLS member, you create the data set HSM.SAMPLE.CNTL and populate it with several members:

- ALLOCBK1
- ALLOSDSP
- ARCCMD01
- ARCCMD90
- ARCCMD91
- HSMEDIT
- HSMHELP
- HSMLOG
- HSMPRESS
- STARTER

These members are reviewed in the following topics in a suggested running order. It is not necessary to run all of the members before you start DFSMShsm. If you choose not to run specific jobs, consider the impact of this selection. In certain cases, you might need to implement equivalent functions.

STARTER

This member provides the basis of your DFSMShsm environment. STARTER contains steps that will allocate the following objects:

- A user catalog and alias
- All DFSMShsm CDSs
- The journal data set
- Member DFSMShsm in SYS1.PROCLIB
- Member ARCCMD00 in SYS1.PARMLIB
- Member ARCSTR00 in SYS1.PARMLIB

You must decide which DFSMShsm functions you plan to use before you run the STARTER member. You must determine the following information before you run the STARTER member:

- A naming convention for DFSMShsm CDSs.
- A naming convention for data that is owned by DFSMShsm.
- The DFSMShsm functions that are implemented.
- The CDSs that are needed.
- In which volumes the DFSMShsm CDSs exist.
- Estimate the initial size of the CDSs.
- Determine how access to the CDSs is serialized. The starter set assumes that Virtual Storage Access Method (VSAM) record-level sharing (RLS) will not be used for serialization.
- Decide how many backup copies of each control data set are required.
- Determine a user ID to use for the DFSMShsm task and define it in advance to your security system. If you are using small-data-set packing (SDSP), the DFSMShsm user ID is used as the high-level qualifier (HLQ) of the SDSP data sets.
- If you are not using the DFSMShsm user ID as the HLQ for DFSMShsm data sets, ensure that you establish security rules in advance for the names that will be used.

DFSMShsm CDSs

The STARTER JCL allocates four single volume CDSs:

- MCDS contains information about migrated data sets, DFSMShsm migration volumes, users, non-SMS DASD volumes, and statistics. MCDS is a key-sequenced data set (KSDS) and a required data set. A multicluster MCDS can consist of more than one physical data set.
- BCDS contains information about backup copies of data sets, backup, and dump volumes. This data set is not required if DFSMShsm availability functions will not be used. The BCDS is one or more KSDSs. A multicluster BCDS can consist of more than one physical data set.
- OCDS contains information about the contents of tape volumes. The OCDS is required if you intend to use tape volumes. The OCDS cannot be multicluster; it must be a single data set. The OCDS is a KSDS.
- JRNL records changes to the MCDS, BCDS, and OCDS. JRNL is only used for recovery. JRNL is a single volume sequential data set.

We include the DEFINE CLUSTER for the single cluster MCDS from the starter set as an example (see Figure 3-1 on page 33). Because the requirements of the individual data sets change over time, always check with the correct level of the *z/OS DFSMShsm Implementation and Customization Guide,* SC23-6869, for the required characteristics of each of the CDSs.

```
/****************************************************************/
/*  THIS PROCEDURE ASSUMES A SINGLE CLUSTER MCDS.  IF MORE THAN */
/*  ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER  */
/*  CDS.                                                        */
/****************************************************************/
/*                                                              */
IF MAXCC = 0 THEN DO
  DEFINE CLUSTER (NAME(?UID.MCDS) VOLUMES(?MCDSVOL) -
       CYLINDERS(?CDSSIZE) FILE(HSMMCDS) -
       STORCLAS(?SCCDSNM) -
       MGMTCLAS(?MCDFHSM) -
       RECORDSIZE(435 2040) FREESPACE(0 0) -
       INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
       SPEED BUFFERSPACE(530432) -
       UNIQUE NOWRITECHECK) -
       DATA(NAME(?UID.MCDS.DATA) -
       CONTROLINTERVALSIZE(12288)) -
       INDEX(NAME(?UID.MCDS.INDEX) -
       CONTROLINTERVALSIZE(2048))
END
/*                                                              */
```

Figure 3-1 DEFINE for the MCDS from the starter set

DFSMShsm procedure

We describe the parameters in the DFSMShsm procedure that is provided by the starter set.

Figure 3-2 contains a copy of the sample DFSMShsm procedure that is provided by the DFSMShsm starter set. Several comment lines were removed.

```
//*****************************************************************/
//*                    DFSMSHSM START PROCEDURE                   */
//*****************************************************************/
//* IF ALL OF THE DFSMSHSM STARTUP PROCEDURE KEYWORDS ARE NEEDED,   */
//* TOTAL LENGTH WILL EXCEED THE 100-BYTE LIMIT, IN WHICH CASE      */
//* YOU SHOULD USE THE KEYWORD STR=XX IN PARM= TO IDENTIFY THE      */
//* PARMLIB MEMBER CONTAINING THE ADDITIONAL KEYWORDS AND PARMS.    */
//*****************************************************************/
//DFSMSHSM   PROC CMD=00,     USE PARMLIB MEMBER ARCCMD00 FOR CMDS
//           STR=00,          PARMLIB MEMBER FOR STARTUP PARMS
//           EMERG=NO,        SETS HSM INTO NON-EMERGENCY MODE
//           CDSQ=YES,        CDSs SERIALIZED WITH ENQUEUES
//           PDA=YES,         PROBLEM DETERMINATION AID
//           SIZE=0M,         REGION SIZE FOR DFSMSHSM
//           DDD=50,          MAX DYNAMICALLY ALLOCATED DATASETS
//           HOST=?HOSTID,    PROC.UNIT ID AND LEVEL FUNCTIONS
//           PRIMARY=?PRIMARY LEVEL FUNCTIONS
//*****************************************************************/
//DFSMSHSM  EXEC PGM=ARCCTL,DYNAMNBR=&DDD,REGION=&SIZE,TIME=1440,
//          PARM=('EMERG=&EMERG','CMD=&CMD','CDSQ=&CDSQ',
//          'UID=?UID','PDA=&PDA','HOST=&HOST','STR=&STR',
//          'PRIMARY=&PRIMARY')
//HSMPARM  DD DSN=SYS1.PARMLIB,DISP=SHR
//MSYSOUT  DD SYSOUT=A
//MSYSIN   DD DUMMY
//SYSPRINT DD SYSOUT=A,FREE=CLOSE
//SYSUDUMP DD SYSOUT=A
//MIGCAT   DD DSN=?UID.MCDS,DISP=SHR
//JOURNAL  DD DSN=?UID.JRNL,DISP=SHR
//ARCLOGX  DD DSN=?UID.HSMLOGX1,DISP=OLD
//ARCLOGY  DD DSN=?UID.HSMLOGY1,DISP=OLD
//ARCPDOX  DD DSN=?UID.HSMPDOX,DISP=OLD
//ARCPDOY  DD DSN=?UID.HSMPDOY,DISP=OLD
//BAKCAT   DD DSN=?UID.BCDS,DISP=SHR
//OFFCAT   DD DSN=?UID.OCDS,DISP=SHR
```

Figure 3-2 DFSMShsm procedure from the DFSMShsm starter set

Do not include unnecessary keywords.

The EXEC statement of the sample that is shown in Figure 3-2 contains the following parameters:

SIZE Is the DFSMShsm region size. Allocate the largest region possible both above and below the line. Use REGION=`0M`, if possible.

DDD Is used to calculate the maximum number of data set allocations that DFSMShsm can hold in anticipation of reuse. The supplied value is `50`.

PARM Is used to pass variable information to the processing program. The length of the PARM subparameters on the EXEC statement must not exceed 100 characters (including commas and excluding parentheses; this rule is a JCL restriction).

The `PARM` parameter of the sample that is shown in Figure 3-2 on page 34 contains the following subparameters:

CMD Specifies the two-character suffix of the ARCCMDxx member of the library that is pointed to by the HSMPARM DD statement. It contains the parameters that define the DFSMShsm working options. The default value is `00`.

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

UID Is the user ID to be assigned to the DFSMShsm started task.

HOST Represents a unique host identifier, *HOSTID*, for each host. The *HOSTID* that you assign to a DFSMShsm address space must be unique to that address space within the sysplex. *HOSTID* allows an optional second character in the value. The function of that second character is now specified by the PRIMARY= keyword. The second character, if specified, is considered only if the PRIMARY= keyword is not specified.

PRIMARY Specify whether this system is a primary system (use `Y`) or a secondary system (use `N`). The backup and dump level functions are performed on the primary processing unit. The automatic secondary space management functions can be performed on any host. The default value is `Y`.

HOSTMODE Specifies how this instance of DFSMShsm relates to various functions of DFSMShsm. HOSTMODE=`MAIN` specifies that this DFSMShsm processes implicit requests, such as recalls and deleting migrated data sets, from user address spaces; processes explicit commands from Time Sharing Option (TSO), such as **`HSENDCMD`** and **`HBACKDS`**; manages ABARS secondary address spaces; allows MODIFY commands from a console; and can run an automatic backup, a dump, and space management. HOSTMODE=`AUX` specifies that this DFSMShsm allows **`MODIFY`** commands from a console and can run automatic backup, dump, or space management. The default value is `MAIN`.

STR Specifies a PARMLIB member that contains DFSMShsm startup parameters, which are logically concatenated with any remaining parameters that are specified on the EXEC statement. The value for the **`STR`** keyword must be two characters.

PDA Specifies that the problem determination aid (PDA) tracing begins before the **`SETSYS PDA`** command is processed or the DFSMShsm initialization is complete.

CDSQ Specifies that DFSMShsm serializes its CDSs with a global enqueue product (global resource serialization (GRS), for example) instead of serializing with volume reserves. When you specify `YES` for this parameter, DFSMShsm serializes the use of the CDSs (between multiple z/OS images) with a global (SYSTEMS) exclusive enqueue and still allows multiple tasks within a single z/OS image to access the CDSs concurrently.

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

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

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

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

SYSPRINT Is the standard DFSMShsm message data set.

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

MIGCAT Points to the MCDS, which is a required data set. Even if you will not use DFSMShsm for migration, you must define an MCDS. The starter set assumes that only a single cluster MCDS will be used.

JOURNAL Points to the journal data set. This data set is not required, but we strongly recommend that you use the JRNL data set.

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

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

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

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

BAKCAT Points to the BCDS. The BCDS is used only if you are using backup processing. Again, the assumption is that only a single cluster BCDS will be used.

OFFCAT Points to the OCDS. The OCDS is required only if you will be use DFSMShsm with tapes.

The starter set provides an empty ARCSTR00 member that is placed into SYS1.PARMLIB. This member is used to contain unique startup parameters. ARCSTR00 can also contain additional parameters from the EXEC statement if you exceed the JCL limit of 100 characters for an EXEC statement.

ARCCMD00 member

Tailor your DFSMShsm started tasks through `SETSYS` commands in the DFSMShsm PARMLIB member ARCCMDxx to ensure that the settings are defined to DFSMShsm at startup initialization. The general format for the `SETSYS` command is shown:

```
SETSYS parameter(option)
```

These parameters are contained in the PARMLIB member that is pointed to by the HSMPARM DD statement in the ARCCMDxx started procedure. The ARCCMD00 that is created by the STARTER job is not intended to be sufficient for a production DFSMShsm environment. It does not contain definitions for ML2 and does not create entries for DASD volumes. It provides a starting place for you to continue tailoring your environment.

These parameters can be divided into four groups that define the following areas:

- Base options
- Space management characteristics
- Availability management characteristics
- ABARS support

This section provides details and suggestions for coding values for scenarios where the STARTER set does not provide a value or where we are changing the value that is specified by the starter set that is described in this topic. For more information, see *V1R10.0 DFSMS Storage Administration Reference (for DFSMSdfp, DFSMSdss, DFSMShsm)*, SC26-7402, which contains a detailed explanation of each possible **SETSYS** keyword. Topics in this section are not covered in the same order that the **SETSYS** commands appear in the starter set ARCCMD00 member.

The base parameters set the basic working options of DFSMShsm. In general, these parameters do not directly relate to a particular function. Instead, they establish several defaults, such as tape management, that define the way DFSMShsm implements the available options.

Define the following base parameters:

- What is the TSO user ID of the DFSMShsm administrator or system programmer?

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

  ```
  AUTH uid DATABASEAUTHORITY(USER)
  AUTH uid DBA(CONTROL)
  ```

  Even if you will use RACF to control access to DFSMShsm commands, define at least one user that can issue commands if a problem occurs with the security system.

- What is the job entry subsystem (JES)?

  DFSMShsm defaults to JES2. If you want to use JES3, you must specify the **JES3** parameter before you specify the first **ADDVOL** command. Specify the JES3 parameter in the following format:

  ```
  SETSYS JES2
  SETSYS JES3
  ```

- Do you want secondary host promotion for this DFSMShsm?

  If a primary DFSMShsm host fails, a host that is started as other than PRIMARY=Y is allowed to take over selected primary host functions. This capability is host promotion. The promoted host can assume responsibility for all backup functions that are limited to the primary host and responsibilities for secondary space management (SSM).

  If you want this hierarchical storage management (HSM) image to be promoted as primary host only, use statement (a) in Example 3-1 on page 38. If you want this HSM image to be promoted as SSM only, use statement (b) in Example 3-1 on page 38. Use statement (c) for this HSM image to be promoted as both primary and SSM.

Example 3-1 Secondary host promotion options

```
(a) SETSYS PROMOTE(PRIMARYHOST(YES) SSM(NO))
(b) SETSYS PROMOTE(PRIMARYHOST(NO) SSM(YES))
(c) SETSYS PROMOTE(PRIMARYHOST(YES) SSM(YES))
```

DFSMShsm host promotion is described in more depth in 3.4.7, “Host promotion” on page 57.

- Do you want only some hosts to perform certain functions?
- Do you want DFSMShsm to reblock eligible data sets during data set recall or recovery?

  All system reblockable data sets are reblocked during recall and recovery if they require reblocking. If data sets are not system reblockable but are of selected categories, this parameter permits DFSMShsm to reblock them. If you do not want these data sets to be reblocked, use statement format (a) in Example 3-2. The format (b) statement in Example 3-2 allows reblocking during recall or in recover to any device type that is supported by DFSMShsm, including target volumes of the same type as the source volume.

Example 3-2 Reblocking options

```
(a) SETSYS NOCONVERSION
(b) SETSYS CONVERSION(REBLOCKTOANY)
```

- How do I specify data set serialization for data sets that are being backed up or migrated?

  To prevent a data set from being changed during backup or migration, access to the data set is controlled by using serialization. The DFSMShsm serialization mechanism is determined by specifying one of the following **SETSYS** parameters.

  Use statement (a) in Example 3-3 when you are sharing volumes and a serialization facility, such as GRS, is not provided. In this case, a reserve is placed on the volume. An example of sharing volumes without GRS is with a VM or VSE system.

  Use statement (b) in Example 3-3 when you are in a single-system environment or are sharing data between z/OS systems and are using GRS to serialize at the data set level.

  We strongly recommend that you specify SETSYS USERDATASETSERIALIZATION where you can.

Example 3-3 Data set serialization specification

```
(a) SETSYS DFHSMDATASETSERIALIZATION
(b) SETSYS USERDATASETSERIALIZATION
```

The following considerations apply to data set serialization:

– The Fast Subsequent Migration function supports data set reconnection in a USERDATASETSERIALIZATION environment only.

– Certain data sets, such as multivolume physical sequential data sets, are processed with SETSYS USERDATASETSERIALIZATION only.

– In a multiple-system environment, do not specify USERDATASETSERIALIZATION unless you installed and enabled a data set serialization facility on your systems. Otherwise, serious data integrity problems can occur.

If you use SETSYS DFHSMDATATSERIALIZATION, DFSMShsm will not select data sets for processing where either the creation date or the date last altered is today.

- Do you want to activate a DFSMShsm exit?

  You can use DFSMShsm installation exits to customize DFSMShsm processing according to your installation requirements. These exits are described in *z/OS V2R10.0 DFSMS Installation Exits,* SC26-7396-14, where you can obtain additional information about the DFSMShsm exit points.

  If you are not planning to use a DFSMShsm exit, specify statement (a) in Example 3-4.

  Statement (b) in Example 3-4 shows how to activate an exit. In this example, we activate the DFSMShsm tape volume exit ARCTVEXT. This exit is called when a tape that is owned by DFSMShsm no longer contains valid data and therefore becomes empty. The ARTVEXT exit is used to tell the OEM tape management systems that DFSMShsm released the ownership of a DFSMShsm tape. If you use DFSMSrmm, ARCTVEXT is not required. However, for an independent software vendor (ISV) tape management system (TMS) product, use statement (b).

  Example 3-4 DFSMShsm exit options

  ```
  (a) SETSYS EXITOFF(TV)
  (b) SETSYS EXITON(TV)
  ```

  Note: If you installed an ISV TMS, specify statement (b) in Example 3-4 and ask the tape management system vendor for its version of ARCTVEXT.

- Do you want to write updated CDS records in the journal data set?

  Journaling is necessary because it is the only way to provide point-in-time forward recovery if a CDS is damaged. The **JOURNAL** parameter specifies that DFSMShsm write the BCDS, MCDS, and OCDS records in the journal data set when it updates them. The following statement guarantees that control is not returned until each CDS change is written to both the CDS and JRNL data set:

  `SETSYS JOURNAL(RECOVERY)`

  The JRNL data set is performance-critical and needs to be allocated on a high-performance device with write caching enabled.

  We strongly recommend both using the JRNL data set and specifying SETSYS JOURNAL(RECOVERY). Without this setting, data that is required for DFSMShsm CDS recovery will not be guaranteed to be recorded.

- Do you want System Measurement Facility (SMF) to contain DFSMShsm statistics?

  If yes, specify statement (a) in Example 3-5 with an SMF user code SMFID to assign it to records that are generated by DFSMShsm. DFSMShsm uses two consecutive user codes (SMFID and smfid+1). The fields are required by the **REPORT** command. If you do not want DFSMShsm to write SMF records, use statement (b) in Example 3-5.

  Example 3-5 Options to set SMF to contain DFSMShsm statistics

  ```
  (a) SETSYS SMF(smfid)
  (b) SETSYS NOSMF
  ```

- Which SYSOUT class do you want to assign to the DFSMShsm output?

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

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

- Do you want the DFSMShsm dump to be written in a system dump data set?

  With statement (a) in Example 3-6, which we recommend, dump is written in a system dump data set every time that an error occurs in the primary or secondary address space. This format is required if you are using Interactive Problem Control System (IPCS). With statement (b) in Example 3-6, DFSMShsm dumps are written where indicated with a SYSABEND, SYSUDUMP, or SYSMDUMP DD statement in the DFSMShsm start procedure.

  Example 3-6 Writing the DFSMShsm dump to a system dump data set options

  ```
  (a) SETSYS SYS1DUMP
  (b) SETSYS NOSYS1DUMP
  ```

- Do you want DFSMShsm to write all activity log messages?

  ACTLOGMSGLVL determines which ARC0734I data set movement messages will be written to the activity log. Statement (a) in Example 3-7, which we recommend, specifies that messages will be generated and logged for all activities. Statement (b) in Example 3-7 specifies that messages will be generated only for activities with a nonzero return code. Statement (c) specifies that only the original space management message is generated.

  Example 3-7 Options for writing activity log messages

  ```
  (a) SETSYS ACTLOGMSGLVL(FULL)
  (b) SETSYS ACTLOGMSGLVL(EXCEPTIONLY)
  (c) SETSYS ACTLOGMSGLVL(REDUCED)
  ```

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

  With ACTLOGTYPE, you can choose whether to create the activity logs as DASD data sets (statement (a) in Example 3-8) or SYSOUT data sets (statement (b) in Example 3-8). For class, substitute an alphanumeric character for the class DFSMShsm is to use for output. DASD specifies that DFSMShsm dynamically allocate DASD data sets with the HLQ HSMACT.

  If the activity logs are written to DASD, the activity logs can be managed by DFSMShsm according to the SMS management class specifications. If the activity logs are written to SYSOUT, you will need to arrange for this data to be archived, for example, by being sent to an external writer.

  Example 3-8 Writing the activity log messages to DASD or SYSOUT

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

- Do you want DFSMShsm to print information about the system console and in the activity logs?

  `MONITOR` is an optional parameter that specifies the informational messages that DFSMShsm will print at the system console and the values that DFSMShsm will use for monitoring space in the journal and CDSs. Do not code this parameter if you do not want DFSMShsm monitor messages on the system console and SYSLOG or OPERLOG.

Otherwise, you can select the information that you want by coding **STARTUP** for messages at DFSMShsm initialization. The **SPACE** subparameter is used to print the volume space-use messages at the system console, during space management. The **VOLUME** subparameter specifies that data set processing messages (ARC0734I) are printed on the system console. You can suppress messages from the console by using PARMLIB MPFLSTxx or your automation product to keep the messages available in SYSLOG or OPERLOG for later review. Specify where to direct DFSMShsm informational messages:

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

- Do you want warning messages on space utilization inside the CDSs and the journal?

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

  ```
  SETSYS MONITOR(JOURNAL(70) BCDS(70) MCDS(70) OCDS(70))
  ```

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

  CSA storage is used to save WAIT Management Work Elements (MWEs). Reasonable starting values, which are also the default values, are shown in statement (a) in Example 3-9. If you do not want to limit the amount of space that is allocated in the CSA, choose statement (b).

  Example 3-9 Options to limit the amount of space that DFSMShsm allocates in the CSA

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

- How many tasks do you want to assign to the RECYCLE process?

  Although a single DFSMShsm host can process only one RECYCLE request at a time, RECYCLE can initiate up to 15 tape processing tasks. Each RECYCLE task allocates two tape drives. Through the following statement, you can limit the number of tape drives that RECYCLE will request:

  ```
  SETSYS MAXRECYCLETASKS(nn)
  ```

- Do you want to use the hardware compaction feature on 3480X tape devices?

  The improved data recording capability (IDRC) feature on 3480s, if installed, can be used on DFSMShsm 3480X output tapes if you code statement (a) in Example 3-10. If you do not want to use this feature on 3480X devices, code statement (b). This parameter is ignored for 3490 and 3590 devices.

  Example 3-10 Hardware compaction feature on the 3480X tape drives

  ```
  (a) SETSYS TAPEHARDWARECOMPACT
  (b) SETSYS NOTAPEHARDWARECOMPACT
  ```

- Do you want to reuse partially full tapes?

  To enable each task, except for data set migration, to initially request a scratch tape, use statement (a) in Example 3-11 on page 42. Data set migration is excluded to avoid marking a tape volume full after a command migration of a single data set to ML2. This option is valuable when you use automatic cartridge loaders. However, this option is not useful if the DFSMShsm tape output is directed to high-capacity tapes, such as the IBM Magstar® media tape, because they will not be used well.

  If you want DFSMShsm to append to tapes that were not marked full when they were last written to, use statement (b) Example 3-11 on page 42. With statement (b), DFSMShsm marks tapes full only when the maximum block count that is specified in the SETSYS **TAPEUTILIZATION** parameter is reached.

DFSMShsm automatically processes TAPECOPY requests for tape volumes that are marked as full, and partial tapes without alternates are not selected for copy.

If you specify the global PARTIALTAPE parameter, it applies to both migration and backup. If you specify a specific function, the parameter applies only to that function. If you specify a global value and a specific function in a single command, the command fails. If you want to specify this parameter differently for migration and backup, do not code the sample that is shown here. Use statement (c) or (d) instead.

Example 3-11 Handling partial tapes

```
(a) SETSYS PARTIALTAPE(MARKFULL)
(b) SETSYS PARTIALTAPE(REUSE)
(c) SETSYS PARTIALTAPE(MIGRATION(MARKFULL | REUSE)
(d) SETSYS PARTIALTAPE(BACKUP(MARKFULL | REUSE)
```

When you use a Virtual Tape Server (VTS) subsystem for migration or backup output, the **MARKFULL** parameter can improve performance by reducing the need to remount yesterday's partial volume to extend it today. Additionally, this usage can reduce the reclamation process of the VTS subsystem because the physical tape, which contains the initial partial tape that was extended, was not invalidated on a true tape.

- Do you want to reduce the occurrences of data sets that span tape volumes?

  In the following statement, you specify the maximum number of megabytes of tape that DFSMShsm can leave unused on tape while trying to eliminate spanning data sets:

  `SETSYS TAPESPANSIZE(nnn)`

  When space is insufficient to contain the entire next data set on the current tape without exceeding the requested maximum utilization, the next data set begins on an empty tape. The default value for this parameter is 500 MB. However, IBM recommends a value of 4000 MB for all device types from 3480 through 3592 tape cartridges.

- Do you want DFSMShsm to wait when a tape allocation is requested?

  If you select statement (a) in Example 3-12, DFSMShsm continues other processing when no device is immediately available and reissues the allocation request every 10 seconds for one minute. After seven attempts, DFSMShsm asks the operator whether to try again or fail the allocation and therefore the function. If the operator prompt is issued, DFSMShsm waits for the write to operator with reply (WTOR) to be answered even if the drives that were requested are now available. We recommend that you use the **NOWAIT** parameter.

  If you want DFSMShsm to wait until a tape request is satisfied, use statement (b) in Example 3-12. In statement (b), all DFSMShsm functions that request allocations, opens, or closes are stopped until this request is satisfied. JES3 installations are forced to use **WAIT** because JES3 schedules tape drives.

Example 3-12 Setting DFSMShsm wait option when a tape allocation is requested

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

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

- Do you want DFSMShsm to use its own tape pool?

  With statement (a) in Example 3-13, DFSMShsm requests scratch tapes every time that a new volume is needed for output and every time that a continuation volume is needed at end of volume (EOV). The SELECTVOLUME specification is typically made with the TAPEDELETION specification. So, according to the TAPEDELETION specification in statement (a), when DFSMShsm no longer needs a tape volume, it tells the TMS that it released the ownership of that tape volume. The TMS is then responsible for returning tape volumes to the global scratch pool.

  With statement (b) in Example 3-13, DFSMShsm selects the tape within a defined tape pool if any are available. It is expected that the installation supplied a set of standard label tapes and identified them to DFSMShsm, by using the **ADDVOL** command. When a tape volume no longer contains valid data, the TAPEDELETION specification in statement (b) tells DFSMShsm to keep it for reuse. If you specify the global parameter, it applies to all functions (migration, backup, and dump). If you specify the parameter for a specific function, it applies only to that type of function.

  We recommend that you not use specific scratch pools. With global scratch pools, you can reduce the overhead of needing to manage specific pools of tapes that are dedicated to DFSMShsm. Global scratch pools are more easily integrated with tape automation solutions.

  Example 3-13 Setting whether DFSMShsm uses its own tape pool

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

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

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

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

  To identify esoteric tape unit names to DFSMShsm, use statement (a) in Example 3-14 on page 44. If an esoteric name represents a set of units with automatic cartridge loaders, and the esoteric name is used to allocate a device for output, the output esoteric is translated to the generic tape unit equivalent for mounting the tape for input. If you do not want that translation, you can specify an alternate unit name translation, such as the example that is shown in statement (b) in Example 3-14 on page 44. Substitute `eso` with the esoteric name that you want to use for output allocations, and `esi` with the esoteric name that you want to use for input allocations. In statement (b), `eso` designates tape drives with automatic cartridge loaders, and `esi` designates tape drives without automatic cartridge loaders.

  You can use statement (c) in Example 3-14 on page 44 if you do not want the translation to occur for certain esoteric names. In this example, a tape that was written by DFSMShsm to a tape that is associated with the esoteric name `eso1` will be allocated for input in a tape unit that is also associated with `eso1`.

If no esoteric tape unit names are identified to DFSMShsm, use statement (d) in Example 3-14. Any previously defined esoteric names are no longer known to DFSMShsm. Statement (d) is the default.

Example 3-14 Using installation-defined esoteric unit names for tape allocations

```
(a) SETSYS USERUNITTABLE(eso1,eso2,eso3,..)
(b) SETSYS USERUNITTABLE(eso1:esi1,eso2:esi2...)
(c) SETSYS USERUNITTABLE(eso1:eso1,eso2:eso2...)
(d) SETSYS NOUSERUNITTABLE
```

- How long do you want DFSMShsm to wait for a tape mount?

  Indicate the number of minutes (maximum `120`) that DFSMShsm can wait for a tape mount before it asks the operator about volume availability. The DFSMShsm default is `15` minutes. The following statement shows that DFSMShsm waits 10 minutes for a tape mount:

  `SETSYS MOUNTWAITTIME(10)`

- Do you want DFSMShsm to ask the operator about the availability of tapes for input before it uses them?

  If you want DFSMShsm to ask the operator whether the requested input tapes are available, use statement (a) in Example 3-15. If you do not want DFSMShsm to issue this operator request, use statement (b) in Example 3-15. If all of the requested input tapes are in an IBM automated tape library, these action messages are not issued, even if (...TAPES(YES)) is specified.

Example 3-15 Options for DFSMShsm to ask the operator about input tape availability

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

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

- Do you want to create a duplex duplicate of your migration and backup tapes?

  The **DUPLEX** parameter allows DFSMShsm to create two tapes concurrently. The two copies of data can be written to the same or different locations. By using statement (a) in Example 3-16, you can duplex both BACKUP and MIGRATION tapes. If you do not want to duplex your tapes, use statement (b) in Example 3-16, or do not specify the **DUPLEX** parameter on the **SETSYS** command. Use statement (c) if you want to duplex migration tapes only and statement (d) for duplexing backup tapes only.

Example 3-16 Options for creating a DUPLEX duplicate of migration and backup tapes

```
(a) SETSYS DUPLEX(MIGRATION(Y) BACKUP(Y))
(b) SETSYS DUPLEX(MIGRATIOM(N) BACKUP(N))
(c) SETSYS DUPLEX(MIGRATIOM(Y) BACKUP(N))
(d) SETSYS DUPLEX(MIGRATIOM(N) BACKUP(Y))
```

- Do you want to limit the maximum tape utilization?

  DFSMShsm provides the **TAPECOPY** command and the **DUPLEX** tape option to create alternate migration and backup copies of cartridge-type tapes. The usefulness of these copies depends on a one-to-one cartridge copy. Because not all cartridges of a certain type have the same capacity, DFSMShsm writes only 97% of the capacity of a cartridge, by default. If you want to change the amount of data that is written to migration and backup tapes, use statement (a) in Example 3-17. You might need to issue the **SETSYS TAPEUTILIZATION** command for each tape unit type that you use if you want to change a particular default.

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

  In a duplex tape environment, if you specify `NOLIMIT`, DFSMShsm uses the default PERCENTFULL value of `97%`. This value is not likely to be a problem unless you are targeting a 3590-Bx or vendor drive in 3490 emulation mode. These drives need a percent value of a few hundred to fully use the tape's capacity. The value of `97%` is fine if you are targeting a 3590-Ex drive, even if it is in 3490 emulation mode.

  IBM recommends that you use `97` (default) in all cases. If you are using the IBM 3590-Bx drives in emulation mode, the recommended value for PERCENTFULL is `2200`. (For drives with a microcode level before 1.9.19.0, use a value of `1100` instead.)

  IBM provides TAPEUTILIZATION recommendations for new drives and media as they become available, usually with the program temporary fixes (PTFs) that provide new function support.

  For OEM drives and media, request this information from your vendor.

  Example 3-17 Limiting the maximum tape utilization

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

(b) SETSYS TAPEUTILIZATION(UNITTYPE(unit) PERCENTFULL(100))
SETSYS TAPEUTILIZATION(LIBRARYMIGRATION PERCENTFULL(100))
SETSYS TAPEUTILIZATION(LIBRARYBACKUP PERCENTFULL(95))

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

- Do you want all migrated and backup copies on DASD to be RACF indicated?

  If the RACF environment is set up with always-call support and generic profiles that are defined for migration and backup qualifiers, we recommend that you use statement (a) in Example 3-18. Migration copies and backup versions will not be RACF indicated.

  Use statement (b) in Example 3-18 to indicate RACF for migration copies and backup versions.

  Example 3-18 RACF indication options for migrated and backup copies

```
(a) SETSYS NORACFIND
(b) SETSYS RACFIND
```

- How do you want to protect DFSMShsm tapes?

  DFSMShsm requires that you select one of three ways to protect tapes. If you want to protect them with RACF, use statement (a) in Example 3-19 on page 46. In this case, RACF must be updated as explained in 5.1, “Providing security for DFSMShsm resources” on page 80. This method is recommended if RACF is installed. If RACF is not installed, you can use statement (b) in Example 3-19. This method protects tapes with the 99365 expiration date. If you choose statement (c) in Example 3-19, DFSMShsm keeps tapes as password-indicated so that only security-authorized programs can access them. If you choose statement (d), you are using RACF and also expiration dates for protecting DFSMShsm tapes. These options are fully supported by other security software products, such as CA-TOPSECRET or CA-ACF2. Security options are not mutually exclusive; they can be used in combination.

  Example 3-19 Options to protect DFSMShsm tapes

```
(a) SETSYS TAPESECURITY(RACF | RACFINCLUDE)
(b) SETSYS TAPESECURITY(EXPIRATION | EXPIRATIONINCLUDE)
(c) SETSYS TAPESECURITY(PASSWORD)
(d) SETSYS TAPESECURITY(RACF EXPIRATION)
```

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

  Statement (a) in Example 3-20 applies only if RACF is installed. It controls the overwriting after a migration or backup copy is deleted from a DASD volume that is owned by DFSMShsm, only if the erase-on-scratch option is associated with the RACF profile of the user’s data set. Specify statement (b) in Example 3-20 if you do not need this overwriting to occur.

  Example 3-20 Options to overwrite deleted migration and backup copies of data sets

```
(a) SETSYS ERASEONSCRATCH
(b) SETSYS NOERASEONSCRATCH
```

  Specifying SETSYS ERASEONSCRATCH can cause a performance impact.

- Do you want to control whether individual backup requests go to DASD or tape?

  With the data set backup commands, a user can target a specific output device type. However, if an output device type is not specified, the **SETSYS DSBACKUP** command controls whether individual backup requests go to DASD or tape. With this command, you can also define the number of tasks that are available, up to 64 tasks, to the data set backup function. The **SETSYS DSBACKUP** command has three parameters: **DASDSELECTIONSIZE**, **DASD**, and **TAPE**.

  The **DASDSELECTIONSIZE (*maximum standard*)** parameter on the **SETSYS DSBACKUP** command allows DFSMShsm to balance the workload between DASD and tape tasks for all WAIT requests that do not target a device type:

  `SETSYS DSBACKUP DASDSELECTIONSIZE(maximum standard)`

  DFSMShsm uses a selection process that is based on data set sizes and the availability of tasks. If the data set is greater than a specified threshold maximum, the data set is directed to tape. Any data set that is less than or equal to a specified threshold standard is directed to the first available task, either DASD or tape.

  To direct the data set to back up to tape, use statement (b) in Example 3-21. Use statement (a) in Example 3-21 for DASD only. This SETSYS directs all backups to one or the other. You can also direct individual requests with the **TARGET** keyword.

Example 3-21 Directing the data set to back up to tape or DASD

```
(a) SETSYS DSBACKUP(DASD(TASK(0)))
(b) SETSYS DSBACKUP(TAPE(TASK(0)))
```

- Do you want RACF discrete profiles to be copied during the data set backup?

  This parameter applies only if RACF is installed. If you use generic profiles with always-call support, select statement (a) in Example 3-22. However, if you do not have an always-call environment, use only discrete profiles and specify statement (b).

Example 3-22 Options to copy RACF discrete profiles during the data set backup

```
(a) SETSYS NOPROFILEBACKUP
(b) SETSYS PROFILEBACKUP
```

- Do you want to issue authorized DFSMShsm commands in a TSO batch environment without RACF?

  To issue authorized DFSMShsm commands in a TSO batch environment without RACF, the user ID that is associated with the job must match the user ID that was entered with the **AUTH** command. If RACF is installed, RACF passes this information to DFSMShsm. In this case, use statement (a) in Example 3-23. If RACF is not installed, use statement (b). An installation without RACF has DFSMShsm retrieve the user ID for TSO batch requests from the protected step control block (PSCB). An installation must ensure that the user ID is placed in the PSCB.

Example 3-23 Issuing authorized DFSMS commands in a TSO batch environment

```
(a) SETSYS NOACCEPTPSCBUSERID
(b) SETSYS ACCEPTPSCBUSERID
```

- Do you want DFSMShsm to compact data sets during migration or backup?

  If you want DFSMShsm to compact data sets during backup and migration to DASD and during backup and migration to tape, specify statement (a) in Example 3-24. Optionally, you can choose the functions that use DFSMShsm compaction. For example, if you want compaction to occur during migration to DASD only, specify statement (c). We recommend that you specify statement (d). If you do not want compaction to occur, do not specify the **SETSYS COMPACT** command.

  If most of your migration and backup data goes to IDRC-capable tape devices or RAMAC Virtual Array (RVA) DASD, our recommendation is to not specify the **SETSYS COMPACT** command.

Example 3-24 SETSYS COMPACT command options

```
(a) SETSYS COMPACT(ALL)
(b) SETSYS COMPACT(DASDBACKUP)
(c) SETSYS COMPACT(DASDMIGRATE)
(d) SETSYS COMPACT(DASDBACKUP DASDMIGRATE)
(e) SETSYS COMPACT(TAPEBACKUP)
(f) SETSYS COMPACT(TAPEMIGRATE)
```

- Do you want to specify the percentage of minimum space savings for software compaction?

If you request software compaction, DFSMShsm compares any history that it has of the number of bytes that are written to the total bytes of the original data set and computes the percentage of bytes that are saved by compaction. If the saved percentage is not greater than or equal to the percentage you specified, DFSMShsm will not compact the data set during subsequent migrations or backups. The DFSMShsm percentage saved default is `40`. Use the following statement to specify the percentage of minimum space savings for software compaction:

`SETSYS COMPACTPERCENT(`*`pct`*`)`

- Do you want to optimize DFSMShsm compaction?

  If you use the DFSMShsm data compaction feature, you can optimize the algorithm that associates the source and object compaction tables with the data set types. The algorithm passes the last-level qualifier of the data set names that belong to these two groups to DFSMShsm with the following statements.

  Example 3-25 Optimizing DFSMShsm compaction

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

- Do you want DFSMShsm to use IBM z Systems™ Enterprise Data Compression (zEDC) for z/OS hardware compression?

  zEDC hardware compression is I/O adapter-based hardware compression that was introduced on IBM zEnterprise® EC12 and IBM BC12 servers. The server requires the zEDC Express feature for the PCI Express (PCIe) I/O drawer (licensed feature number FC0420).

  For the operating system, you need the PTFs for zEDC Express for z/OS feature and for queued sequential access method (QSAM) and basic sequential access method (BSAM) zEDC exploitation (see authorized program analysis report (APAR) OA42195) with coexistence support on z/OS V1.12 and z/OS V1.13.

  For DFSMShsm, you must activate compression in the IGDSMSxx member and set up the zEDC compression in the DFSMShsm startup member. You can choose different levels to implement the zEDC compression as shown in Example 3-26.

  Example 3-26 zEDC hardware compression options in DFSMShsm migration and backup

```
SETSYS ZCOMPRESS(ALL NONE)
SETSYS ZCOMPRESS(DASDBACKUP(YES))
SETSYS ZCOMPRESS(DASDMIGRATE(YES))
SETSYS ZCOMPRESS(TAPEBACKUP(YES))
SETSYS ZCOMPRESS(TAPEMIGRATE(YES))
```

  For DFSMShsm DUMP processing, you can specify zCOMPRESS(YES) on the DUMPCLASS if you want to activate zEDC compression for DFSMShsm dumps, also.

  For more information about the server setup for zEDC compression, see the *IBM zEnterprise EC12 Technical Guide,* SG24-8049, and the *IBM zEnterprise BC12 Technical Guide*, SG24-8138. For more information about the specific DFSMShsm setup, see *zEDC Compression: DFSMShsm Sample Implementation*, REDP-5158.

- Do you want DFSMShsm to load DFSMSdss in separate address spaces?

  Enabling DSSXMMODE causes workload that requires DFSMSdss to be offloaded to secondary address spaces. This offloading reduces contention for resources in the primary DFSMShsm address space. However, you need to perform additional setup steps beyond the **SETSYS** command. For more information, see 10.1.5, "Using DSSXMMODE" on page 262.

Example 3-27 Options for loading DFSMSdss in separate address spaces

```
SETSYS DSSXMMODE(Y)
SETSYS DSSXMMODE(N)
```

- Do you want DFSMShsm to automatically generate a recycle command when recall takes a tape away from recycle?

  When a tape takeaway occurs, a new recycle command for the original tape is generated automatically by using the following command:

  `SETSYS RECYCLETAKEAWAYRETRY(YES MAXRETRYATTEMPTS(nn) DELAY(mmmm | no)`

  This function is helpful because the storage administrator no longer needs to perform this task manually.

ARCSTR00 member

This member contains additional startup parameters as specified in the **STR=00** parameter of the DFSMShsm startup procedure. The sample member that is provided by the STARTER job is an empty member, which is shown in Figure 3-3.

```
/****************************************************************/
/*              DFSMSHSM ADDITIONAL STARTUP COMMAND MEMBER      */
/****************************************************************/
EMERG=NO,STARTUP=YES,CDSSHR=YES,CDSQ=YES,CDSR=NO
```

Figure 3-3 Additional startup parameters

The ARCSTR00 member can contain the following additional parameters:

STARTUP Specifies whether startup messages are printed at the operator console. The default value is NO.

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

CDSR Specifies that DFSMShsm serializes its CDSs with volume reserves. If you use a cross-system serialization product, you can specify CDSQ=YES instead. CDSQ=YES serializes the CDSs (among multiple MVS images) with a global (SYSTEMS) exclusive enqueue but allows other tasks within a single MVS image to access the CDSs concurrently. When a serialization technique is not specified, the default serialization technique depends on the following specified HOSTMODE. If HOSTMODE=MAIN, DFSMShsm assumes CDSR=YES. If HOSTMODE=AUX, DFSMShsm indicates an error with message ARC0006I.

RNAMEDSN Specifies whether to use a new serialization method so that no interference occurs between HSMplexes that are contained within a single GRSplex. When you specify `YES` for this parameter, you are invoking the new method of serialization, which uses the data set name of the CDSs and the journal. The default value is `NO`.

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

LOGSW Specifies whether to automatically swap the log data sets at startup. The default value is `NO`.

CELLS DFSMShsm uses the cell-pool (CPOOL) function of MVS to obtain and manage virtual storage in its address space for the dynamically obtained storage for certain high-usage modules, and for data areas DFSMShsm frequently gets and frees. The **`CELLS`** parameter provides the cell sizes for five cell pools. The default value is (`200,100,100,50,50`).

3.2.2 ALLOCBK1

ALLOCBK1 allocates CDS backups: four copies of each of the MCDS, BCDS, OCDS, and JRNL. This function assumes that you are writing backups of the CDS to DASD. DFSMShsm backups can be written to either tape or DASD. We recommend writing CDS backups to DASD where possible.

We also strongly recommend increasing the number of backups to provide a week's worth of data at a minimum. This function is described further in Chapter 12, "Control data set considerations" on page 341. The number of backup copy data sets that DFSMShsm uses is determined by the **`SETSYS CDSVERSION BACKUPCOPIES`** parameter. You must have at least as many data set backup copies as are allocated because the specification of **`BACKUPCOPIES`** or **`CDS`** backup fails at a certain point.

3.2.3 ALLOSDSP

You need to run this job only if you use SDSP data sets. This job allocates and initializes a single SDSP data set. If you defined ML1 volumes with ADDVOL SDSP, you need to allocate and initialize one SDSD for each ML1 volume. See 11.3, "Maintaining SDSP data sets: Reorganization" on page 338.

3.2.4 ARCCMD01

DFSMShsm SETSYS specifies whether to use tape for ML2. This member needs to be copied into the DFSMShsm startup member that is created by the STARTER job. Figure 3-4 on page 51 shows the contents of the ARCCMD01 member.

```
/***************************************************************/
/*  DFSMSHSM STARTUP COMMAND MEMBER FOR LEVEL 2 TAPE MIGRATION  */
/*                                                              */
/*  APPEND THIS COMMAND STREAM TO ARCCMD00 TO PROVIDE LEVEL 2   */
/*  TAPE MIGRATION                                              */
/***************************************************************/
/***************************************************************/
/*          DFSMSHSM LEVEL 2 TAPE MIGRATION PARAMETERS          */
/***************************************************************/
/*                                                              */

SETSYS -
  TAPEMIGRATION(ML2TAPE)    /* MIGRATE TO LEVEL 2 TAPE.         */

SETSYS -
  MIGUNITNAME(3590-1)       /* START WITH 3590-1 ML2 TAPE       */
                            /* UNIT.                            */
SETSYS -
  ML2RECYCLEPERCENT(20)     /* LOG MESSAGE WHEN VALID DATA      */
                            /* ON AN ML2 TAPE FALLS BELOW       */
                            /* 20%.                             */
                            /* 20%.                             */
SETSYS -
  TAPEMAXRECALLTASKS(1)     /* ONE TAPE RECALL TASK AT A TIME   */

/*                                                              */
/***************************************************************/
/* SEE MEMBER ARCCMD91 IN HSM.SAMPLE.CNTL FOR AN EXAMPLE        */
/* OF ADDVOL COMMANDS TO BE USED IN CONJUNCTION WITH LEVEL      */
/* 2 TAPE MIGRATION.                                            */
/***************************************************************/
```

Figure 3-4 ARCCMD01 member from the DFSMShsm starter set

3.2.5 ARCCMD90

This member provides DFSMShsm **ADDVOL** sample commands. Place these commands at the end of the ARCCMD00 member that is created by the STARTER job.

SETSYS commands are required for any non-SMS primary volumes and ML1 volumes.

3.2.6 ARCCMD91

The starter set provides a sample **ADDVOL** command for ML2 tape volumes. This command is shown in the Figure 3-5 on page 52.

```
/*                                                              */
/****************************************************************/
/* THE FOLLOWING EXAMPLE ADDVOL COMMANDS CAN BE USED WITH THE   */
/* ARCCDOO MEMBER OF THE STARTER SET TO IDENTIFY LEVEL 2 TAPE   */
/* MIGRATION VOLUMES.  AFTER YOU HAVE ADDED A VOLUME SERIAL     */
/* NUMBER AND A UNIT TYPE IN THE SPACE PROVIDED, APPEND THIS    */
/* COMMAND STREAM TO YOUR ARCCMD00 MEMBER.                      */
/*                                                              */
/****************************************************************/
/*                                                              */
ADDVOL ______                /* ADD A VOLUME (PROVIDE SERIAL)   */ -
       MIGRATION             /* AS A MIGRATION LEVEL 2 TAPE     */ -
        (MIGRATIONLEVEL2)    /* VOLUME.                         */ -
       UNIT(______)          /* PROVIDE PROPER UNIT TYPE.       */
/*                                                              */
```

Figure 3-5 Starter set ARCCMD91 member

These commands do not need to be added to your ARCCMD00 member. ADDVOL for ML2 volumes needs to be issued only if you use a dedicated tape pool for DFSMShsm. If you plan to use scratch volumes for DFSMShsm input, you do not need to pre-define volumes by using **ADDVOL** commands. For more information about ML2, see Chapter 9, “Space management” on page 211.

3.2.7 HSMHELP

Add HSMHELP to a data set in your ISPF HELP concatenation, for example, SYS1.HELP. HSMHELP provides **HELP** command support for authorized DFSMShsm commands.

3.2.8 HSMLOG

Use the ARCPRLOG program to print the contents of the ARCLOGX and ARCLOGY data sets. This member is not required for the initial setup of DFSMShsm, but it is required if you intend to use this data for later performance reporting. The sample JCL that is provided is shown in Figure 3-6 on page 53.

```
//*****************************************************************/
//*         THIS SAMPLE JOB PRINTS THE DFSMSHSM LOG              */
//*                                                              */
//* REPLACE THE ?UID VARIABLE IN THE FOLLOWING SAMPLE JOB WITH   */
//* THE NAME OF THE DFSMSHSM -AUTHORIZED USERID (1 TO 7 CHARS).  */
//*                                                              */
//* (NOTE: UID AUTHORIZATION IS VALID IN A NON-FACILITY CLASS    */
//* ENVIRONMENT ONLY, OTHERWISE, FACILITY CLASS PROFILES         */
//* WILL BE USED FOR AUTHORIZATION CHECKING.)                    */
//*****************************************************************/
//*
//PRINTLOG EXEC PGM=ARCPRLOG
//ARCPRINT DD SYSOUT=*
//ARCLOG DD DSN=?UID.HSMLOGY1,DISP=OLD
//ARCEDIT DD DSN=?UID.EDITLOG,DISP=OLD
//*
//EMPTYLOG EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=?UID.HSMLOGY1,DISP=OLD
//SYSUT1 DD DUMMY,DCB=(?UID.HSMLOGY1)
/*
```

Figure 3-6 Sample JCL from HSMLOG member

If your DFSMShsm ARCLOG data sets are filling rapidly, this sample needs to be updated, perhaps to use a daily generation data group (GDG) for processing. We describe managing ARCLOG data in more detail in 3.1, "DFSMShsm components" on page 30.

3.2.9 HSMEDIT

This sample job runs the ARCPEDIT program and formats copies of the ARCLOGX and ARCLOGY data that is produced by ARCPRLOG. Figure 3-7 shows the sample JCL.

```
//EDITLOG  JOB ?JOBPARM
//*
//*****************************************************************/
//*           THIS JOB PRINTS THE EDIT-LOG DATA SET              */
//*                                                              */
//* REPLACE THE FOLLOWING ?UID VARIABLE WITH THE NAME OF THE     */
//* DFSMSHSM-AUTHORIZED USER (1 TO 7 CHARS).                     */
//* (NOTE: UID AUTHORIZATION IS VALID IN A NON-FACILITY CLASS    */
//* ENVIRONMENT ONLY, OTHERWISE, FACILITY CLASS PROFILES WILL BE */
//* USED FOR AUTHORIZATION CHECKING.)                            */
//*****************************************************************/
//*
//EDITLOG EXEC PGM=ARCPEDIT
//ARCPRINT DD SYSOUT=*
//ARCLOG DD DSN=?UID.EDITLOG,DISP=SHR
/*
```

Figure 3-7 Sample JCL from HSMEDIT member

3.2.10 HSMPRESS

This sample job reorganizes DFSMShsm CDSs. It is not required during the initial DFSMShsm setup, but it is required later. Due to the pattern of updates in DFSMShsm CDSs, we recommend regular reorganization of the DFSMShsm CDSs. We describe DFSMShsm control data set maintenance in Chapter 12, "Control data set considerations" on page 341.

3.3 Media that is owned by DFSMShsm

Media that is owned by DFSMShsm consists of both tape volumes and DASD volumes. No special preparation is required for tape volumes. They require a standard tape label. DFSMShsm DASD volumes must be initialized as non-SMS managed. Generally, these volumes contain large numbers of non-VSAM data sets. A VSAM volume data set (VVDS) is required only when an ML1 volume contains an SDSP data set. However, consider increasing volume table of contents (VTOC) and index VTOC sizes for volumes that are owned by DFSMShsm.

3.4 DFSMShsm operations

Stopping and starting DFSMShsm in a single system environment and a multiple system environment are described. We also describe automating DFSMShsm and where it fits with automating started tasks.

3.4.1 Stopping DFSMShsm

DFSMShsm stops in response to either an **MVS STOP** command or a **MODIFY STOP** command that is issued to the address space. It also shuts down cleanly in response to an authorized user that issues the **HSEND STOP** command. Figure 3-8 shows SYSLOG information from a DFSMShsm shutdown.

```
P HSM
ARC0016I DFSMSHSM SHUTDOWN HAS BEEN REQUESTED
ARC1502I DISCONNECTION FROM STRUCTURE SYSARC_MVSLS_RCL 313
ARC1502I (CONT.) WAS SUCCESSFUL, RC=00, REASON=00000000
ERB425I III: UNABLE TO GATHER RESOURCE HSM
ARC0002I DFSMSHSM SHUTDOWN HAS COMPLETED
GSDMV20I -JOBNAME  STEPNAME PROCSTEP CCODE ELAPSED-TIME  CPU-TIME STEPNO
GSDMV21I -HSM      HSMS     HSM          0     46:06:12     3.96S     1
IEF404I HSM - ENDED - TIME=11.54.40
GSDMV22I -HSM      ENDED.  NAME-* NO NAME PROVIDED * TOTAL CPU TIME=
.02  TOTAL ELAPSED TIME=2766.2
$HASP395 HSM      ENDED
```

Figure 3-8 Sample DFSMShsm shutdown

If DFSMShsm does not respond to a stop command, you can use the **MVS CANCEL** command to stop the address space.

In certain circumstances, DFSMShsm takes longer to complete a stop command. In particular, if you specified, or defaulted to, PARMLIB SMFPRMxx DDCONS(`YES`) for started tasks, DFSMShsm must perform potentially significant processing to consolidate EXCP sections for SMF30 records. We recommend that you specify DDCONS(`NO`) for PARMLIB SMFPRMxx. For more information about factors that can affect DFSMShsm operation, see *z/OS MVS System Management Facilities,* SA22-7630.

3.4.2 Starting DFSMShsm

Start the DFSMShsm address space by using the **`MVS START`** command. In a single system environment, start DFSMShsm as the primary address space. See Figure 3-9.

As DFSMShsm starts, it provides a series of status informational messages as the various initialization phases complete. Full DFSMShsm functions are not available until DFSMShsm issues message ARC0008I.

```
S HSM.HSMS,HOST=SN
$HASP100 HSM      ON STCINRDR
IEF695I START HSM      WITH JOBNAME HSM      IS ASSIGNED TO
 , GROUP SYS1
$HASP373 HSM      STARTED
IEF403I HSM - STARTED - TIME=12.02.22
ARC0037I DFSMSHSM PROBLEM DETERMINATION OUTPUT DATA 330
ARC0037I (CONT.) SETS SWITCHED, ARCPDOX=CCTS.HSM.AAIS.PDOX,
ARC0037I (CONT.) ARCPDOY=CCTS.HSM.AAIS.PDOY
ARC0001I DFSMSHSM 1.13.0 STARTING HOST=S IN 331
ARC0001I (CONT.) HOSTMODE=MAIN
ARC1700I DFSMSHSM COMMANDS ARE RACF PROTECTED
ARC0041I MEMBER ARCCMD00 USED IN SYS1.PARMLIB
ARC0100I SETSYS COMMAND COMPLETED
ARC0100I SETSYS COMMAND COMPLETED
...
ARC0270I DUMP CYCLE DEFINITION SUCCESSFUL
ARC0216I DUMPCLASS DEFINITION MODIFIED, CLASS=TEST1
ARC0216I (CONT.) RC=0
ARC0120I MIGRATION VOLUME LSHM01 ADDED, RC=0000,
ARC0120I (CONT.) REAS=0000
...
ARC0171I SETMIG LEVEL CATALOG PROCESSED
ARC0120I MIGRATION VOLUME LSHM01 ADDED, RC=0000,
ARC0120I (CONT.) REAS=0000
ARC0171I SETMIG LEVEL MASTER PROCESSED
ARC0171I SETMIG LEVEL ACDN PROCESSED
ARC0171I SETMIG LEVEL CATALOG PROCESSED
...
ARC0270I BACKUP CYCLE DEFINITION SUCCESSFUL
ARC0180I USER MHLRESBAUTHORIZATION IS NOT CHANGED , 4
ARC0180I (CONT.) RC= 4
ARC0038I RESOURCE MANAGER SUCCESSFULLY ADDED.  RETURN
ARC0038I (CONT.) CODE=00
ARC0008I DFSMSHSM INITIALIZATION SUCCESSFUL
```

Figure 3-9 DFSMShsm startup

Starting DFSMShsm in a multisystem environment

In a multisystem environment, start only a single DFSMShsm address space identifier (ASID) as a primary host. All DFSMShsm address spaces must share MCDS, BCDS, OCDS, and JRNL data sets.

Multiple hosts can share HSMPARM data by using either host-specific ARCCMDxx members or a single ARCCMDxx member with ONLYIF statements. More than one command can be specified for a specific host and ONLYIF command by using the BEGIN and END that are shown in Example 3-28.

Example 3-28 ONLYIF BEGIN and END

```
ONLYIF HSMHOST(1,2,3)
BEGIN
SETSYS AUTODUMPSTART(0300 0400 0500)
SETSYS BACKUP(TAPE)
SETSYS MAXDUMPTASKS(3)
DEFINE BACKUP(Y)
DEFINE DUMPCYCLE(Y)
END
```

You must ensure that other data sets that are allocated in the DFSMShsm startup are unique to each address space. If you use separate procedure libraries, maintaining individual DFSMShsm start members is simple; however, remember that you might need to propagate changes to multiple locations.

If you use a single shared procedure library, you can use system symbols to ensure that unique data sets are used. You can also create unique symbols for other startup variables.

You need to perform an initial program load (IPL) to create the symbols before they can be used.

You need to ensure that each DFSMShsm host has a unique ID. Host IDs are specified by using the HOST=x override on the DFSMShsm startup procedure. You also need to ensure that only a single primary DFSMShsm is started within a sysplex. You establish a host as primary host either by using the **PRIMARY=YES** startup parameter or the older form of startup parameter, **HOST=xY**.

If you do not start any host as a primary host, the functions that can be performed only by a primary host will not be run by any DFSMShsm address space:

- Backing up CDSs as the first phase of automatic backup
- Backing up data sets that were migrated before they were backed up
- Moving backup versions of data sets from ML1 volumes to backup volumes
- Deleting expired dump copies automatically
- Deleting excess dump VTOC copy data sets

It is possible to define more than one primary host. However, these functions might run more frequently than expected.

3.4.3 MASH implications

The requirements for running multiple DFSMShsm address spaces on a single system are the same as the requirements in the multisystem description in the topic "Starting DFSMShsm in a multisystem environment" on page 56. Additionally, you must create a unique name for each DFSMShsm address space within a system.

It is possible to issue commands to other than the primary address space by using console support. However, if you issue **HSEND** commands from TSO, they are always processed by the primary DFSMShsm address space on the systems where they were issued, including commands that are issued through batch unless an EMCS console environment is established in which to direct your command to a specific DFSMShsm address space.

3.4.4 Sending commands to DFSMShsm

Commands can be issued by using either modify commands from a console or Extended Multiple Console Support (EMCS) console, or through a TSO background job. If using a background job, the user that issues the commands needs to be authorized to the command that is being issued.

If you use modify commands to communicate to DFSMShsm address spaces, the capability to issue a single command from one system that affects all DFSMShsm address spaces that are running on this system and other systems in the sysplex is beneficial. The z/OS route command and the z/OS modify command support the use of wildcards to identify address spaces.

3.4.5 Secondary address spaces

Secondary address spaces that are created by either ABARs or as the result of SETSYS DSSXMMODE do not respond directly to commands. Commands to influence these address spaces need to be issued to the DFSMShsm address space that created the secondary address space.

All commands must be issued to the primary DFSMShsm system not the secondary address spaces.

3.4.6 Automation

For DFSMShsm startup, we recommend that you not start DFSMShsm until your JES and TMSs are initialized. DFSMShsm has no dependencies on subsystems other than JES being available; however, if your TMS is not initialized, tape requests are failed. DFSMShsm must be started before batch work is permitted to start to prevent jobs from failing or waiting for data set recalls.

Similarly, Data Facility Storage Management Subsystem (DFSMS) needs to be shut down after batch and TSO users finish, but before your TMS and JES are terminated.

3.4.7 Host promotion

If a primary DFSMShsm host fails, another DFSMShsm host in the sysplex, on the same or another z/OS image, can take over a subset of the functions of the primary host. Two levels of promotion are available. The level of promotion is determined by the **SETSYS PROMOTE** command. We described this command in “ARCCMD00 member” on page 36.

In a sysplex that remains DFSMShsm, hosts are notified through a cross-system coupling facility (XCF) group when either a primary host or an SSM host terminates.

If a secondary host is promoted to primary host, it issues message ARC1522I to indicate that the promotion is complete. Information about backup and dump windows from the failing primary host is copied to the host that is being promoted with the status of the exit ARCCBEXT.

If the promoted host does not normally perform autobackup, it performs unique sections of autobackup, for example, control data set backup, but it will not start volume backup. Similarly, if you limit DFSMShsm functions by system or system group at the storage group level, a promoted host cannot overcome these restrictions.

When you implement any automation, consider that host promotion might occur as a result of an unscheduled DFSMShsm host shutdown.

4

SMS and its interactions with DFSMShsm

Implementing DFSMShsm, whether in an existing storage management subsystem (SMS) environment or from scratch, requires you to change your SMS configuration. In this chapter, we describe the impact of settings in your SMS environment (classes, groups, and automatic class selection (ACS) routines) on how data is managed by DFSMShsm and on how data that is owned by DFSMShsm is managed.

4.1 DFSMShsm in an SMS environment

To implement DFSMShsm into an existing SMS environment or from scratch, you must change your SMS configuration. If you plan to implement DFSMShsm in a non-SMS environment, consider also converting the data that DFSMShsm manages to SMS control as part of the DFSMShsm implementation. You have more options for and granularity in enforcing those options when data is managed by SMS.

In a DFSMShsm environment, data can be divided into the following categories:

- Data that is managed by DFSMShsm

  This group includes all of the user and application data that DFSMShsm manages according to the parameters that are coded in the SMS management classes. Certain DFSMShsm system data sets are also included in this group.

- Data that is owned by DFSMShsm

  User data that is now on DFSMShsm-owned volumes, for example, migration level 2 (ML2) data or backup copies. The data in this group belongs to DFSMShsm. SMS must be updated so that it recognizes the DFSMShsm data and excludes it from SMS management.

- DFSMShsm control data

  Data, such as control data sets (CDSs) that are required for the operation of DFSMShsm.

The description in this chapter assumes that the data your DFSMShsm is managing as well as potentially a subset of DFSMShsm control data and DFSMShsm-owned data can be SMS-managed.

DFSMShsm migration level 1 (ML1) volumes and direct access storage device (DASD) backup volumes *cannot* be SMS-managed.

DFSMShsm CDSs, their backups, log data sets and ML2, tape backup volumes, and DUMP volumes can be SMS-managed if they reside on DASD or reside in SMS-managed tape libraries.

We start by describing how the SMS values (groups and classes) that are assigned to data sets determine how DFSMShsm manages the data sets. These topics are not intended to provide an in-depth description about converting data to SMS management, but instead focus on the attributes of SMS classes and groups that interact with DFSMShsm.

Next, we describe how you can use SMS attributes to manage the DFSMShsm resources. This description includes how you can use access control list (ACS) routines to ensure that the correct classes and groups are assigned to data that is both managed and owned by DFSMShsm.

This topic is restricted to SMS-managed data sets unless otherwise stated. Several of the same attributes can be assigned to non-SMS-managed data sets by **SETSYS** commands.

We start with the SMS attributes that can be assigned to a data set, describing them in the reverse order to the order they are assigned:

- Pool storage groups
- Tape storage groups
- Management classes
- Storage classes
- Data classes

Then, we describe assigning these constructs by using ACS routines for data that is both DFSMShsm-managed and DFSMShsm-owned.

All of these groups and classes are defined to your SMS configuration by using Interactive Storage Management Facility (ISMF). For detailed instructions about defining and maintaining an SMS configuration, see *z/OS V1R10.0 DFSMS Storage Administration Reference (for DFSMSdfp, DFSMSdss, DFSMShsm),* SC26-7402.

4.2 POOL storage group

A POOL storage group describes a grouping of DASD on which data sets reside. Figure 4-1 shows the ISMF Pool Storage Group Alter panel for the DB8B storage group in our environment.

```
DGTDCSG2                   POOL STORAGE GROUP ALTER         Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : DB8B
To ALTER Storage Group, Specify:
 Description ==> DASD FOR DB2 V8 DB8B
             ==>
 Auto Migrate . . N  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . N  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . N  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . . . .
                                      Copy Pool Backup SG Name . . . SGCPB1

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

 ALTER     SMS Storage Group Status . . . N   (Y or N)
 ALTER     SMA Attributes . . . . . . . . N   (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Page;
```

Figure 4-1 Storage group attributes

Several DFSMShsm functions are managed at a storage group level:

- Space management
- Availability management:
  - Automatic backup processing
  - Automatic dump processing

You can also control which system performs what functions against the volumes of a storage group at the system or system group level.

The storage group specifications determine only how DFSMShsm automated processing processes the volume. They have no impact on how commands that are directed to the volume or against data sets that are allocated on the volume are processed.

4.2.1 Space management

Both primary space management and interval migration are determined by the storage group that is assigned to the first extent of the data set. This determination occurs whether or not a data set is eligible for processing by space management. If a data set extends over more than one storage group (for example, a primary storage group and an extend storage group), the settings of the storage group where the first extent of the data set resides are critical.

Two separate parameters determine whether DFSMShsm processes data from the volumes in a storage group. Both of these parameters are set at the storage group level (not the volume level).

The first parameter that you define is the migration attribute of a storage group with the "Auto Migrate" attribute. The second attribute consists of a high and a low allocation threshold that determines whether any particular volume is targeted by a particular run of space management.

Auto Migrate attribute

The following options are available when you determine whether DFSMShsm space management processes data:

N DFSMShsm will not process this volume during space management.

Y This setting is the default setting for a pool storage group. DFSMShsm processes data sets from volumes in this storage group during primary space management. Volumes are targeted by primary space management until either the volume falls below its "MIGR LOW" threshold or no more data sets that are eligible to be migrated are on the volume.

If SETSYS INTERVALMIGRATION was issued, volumes in a storage group are also eligible for interval migration processing.

I Data sets on volumes in a storage group that are defined as AM=`I` are eligible for processing by both primary space management and interval migration regardless of the setting of SETSYS INTERVALMIGRATION.

P Data sets are eligible for primary space management but not interval space management regardless of the setting of SETSYS INTERVALMIGRATION.

Thresholds

Storage groups are assigned both a low and a high allocation threshold. Thresholds apply at the volume level. Thresholds are specified as a percentage of the space in the storage group.

Thresholds are applied as a percentage of the space for each volume so where storage groups consist of volumes of different capacities, thresholds are applied according to each volume's capacity.

A volume that is over its high allocation threshold can still be targeted for new allocations if no other volume is available in the eligible storage groups with sufficient space to accommodate the data set.

Low allocation threshold

The default low allocation threshold is `1`.

During primary space management, data sets are processed until either the allocated space on the volume drops below the storage group low allocation threshold or until DFSMShsm can find no more data sets on the volume that are eligible for processing.

High allocation threshold

The default high allocation threshold is `85`. During interval migration, DFSMShsm moves eligible data sets from each volume with allocation at or above the MIGR HIGH value until the allocation reaches the MIGR LOW value. When Auto Migrate is `I`, migration is performed when the space that is used exceeds the halfway mark between the HIGH and LOW thresholds.

Migrate system/system group name

The default value is blank. By specifying a system or system group, you can restrict DFSMShsm space management to only process volumes in this storage group on the system or system group that is specified.

4.2.2 Availability management

When you define a pool storage group, you define settings for both automatic backup and automatic dump processing.

Automatic backup processing

Enabling automatic backup processing for a storage group is a binary choice. By default, the following Auto Backup attributes are enabled for pool storage groups:

N DFSMShsm availability management does not target data sets whose primary extent is on volumes in this storage group during automatic backup processing.

Y DFSMShsm availability management targets data sets whose primary extent is on volumes in this storage group during automatic backup processing.

You can also limit processing to a single system or system group by specifying a system or system group name, which is similar to the configuration for space management.

Automatic dump processing

Although the automatic dump process is part of DFSMShsm availability management, you can specify the Auto Dump attribute of a storage group independently of the Auto Backup attribute.

Auto Dump attribute

Two settings are possible for the Auto Dump attribute: `Y` to enable automatic dump processing and `N` to disable it. The default setting for the Auto Dump attribute for a pool storage group is `N`.

If you use overflow and extend storage groups for automatic dump processing, these groups must be managed in the same way as the primary storage group. This approach ensures that all data that was targeted to this storage group is managed in the same way.

Dump classes

Each pool storage group can be assigned 1 - 5 DFSMShsm dump classes. As with automatic migration and automatic backup, you can also limit automatic dump processing of a storage group to a specific system or system group.

4.3 Tape storage group

Tape storage groups are used to logically group automated or manual tape libraries. If you plan to use SMS-managed tape and direct DFSMShsm tape allocations to an SMS-managed library, you need to define at least one tape storage group.

If you plan to create off-site copies of DFSMShsm media, or to separate the location of migrated data from backup or dump copies, more than one tape storage group might be required. The only entries that are required in a tape storage group are 1 - 8 tape library names. Figure 4-2 shows an example of defining a tape storage group.

```
DGTDCSGA                 TAPE STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : SGLIB1

To ALTER Storage Group, Specify:

  Description ==>
              ==>

  Library Names  (1 to 8 characters each):
    ===> LIB1      ===>          ===>          ===>
    ===>           ===>          ===>          ===>



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


Use ENTER to Perform Verification and Selection;
```

Figure 4-2 Define tape storage group

Tape libraries are defined by using ISMF Option 10, Library Management, where library names are registered. For more information about defining tape libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*, SC35-0427.

4.4 Management class

The *management class* that is assigned to a data set is the primary factor that determines how DFSMShsm treats a data set. Only SMS-managed DASD data sets are assigned management classes. Management classes define the following attributes for a data set:

- Expiration
- Partial release
- Migration
- Special processing for generation data sets (GDSs)
- Backup
- Aggregate backup and recovery support (ABARs)

4.4.1 Expiration

Three attributes together determine whether a data set is eligible to be expired. These values apply to DASD data sets based on the management class that is assigned to the data set. These values can also influence the life of tape data sets. For DASD data sets, expiration processing is performed as part of primary space management. For tape data sets, expiration is performed by your tape management system.

The following attributes define whether a data set expires:

- Expire after days non-usage
- Expire after date/days
- Retention limit

The default value for each of these settings is `NOLIMIT`.

Figure 4-3 shows the first page of the Management Class Alter panel.

```
DGTDCMC1                 MANAGEMENT CLASS ALTER              Page 1 of 5
Command ===>

SCDS Name . . . . . . : SYS1.SMS.MHLRES3.SCDS
Management Class Name : MC365

To ALTER Management Class, Specify:

  Description ==>
              ==>

  Expiration Attributes
    Expire after Days Non-usage  . . 365        (1 to 93000 or NOLIMIT)
    Expire after Date/Days . . . . . NOLIMIT    (0 to 93000, yyyy/mm/dd or
                                                NOLIMIT)

  Retention Limit  . . . . . . . . . NOLIMIT    (0 to 93000 or NOLIMIT)



Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

Figure 4-3 ISMF Management Class Alter panel

DFSMShsm uses a combination of the three values to determine whether or when a DASD data set is eligible for expiration.

Expire after days non-usage

This attribute determines how many days will elapse after the last use of the data set before it is eligible to be expired. The data set's last use is determined by the last reference date in the volume table of contents (VTOC) (DS1REFD in the format1 or format8 data set control block (DSCB)).

Expire after date/days

This attribute defines either the actual date on which a data set becomes eligible for expiration, or the absolute number of days after its initial allocation that a data set becomes eligible for expiration.

Retention limit

If you specify a value for the retention limit attribute other than `NOLIMIT`, you can restrict the values that users can specify for data sets at allocation time. A user-specified value for retention period (RETPD) or expiration date (EXPDT) can be accepted, reduced, or removed entirely. If `NOLIMIT` is specified for the retention period, any RETPD or EXPDT that is assigned to a data set is accepted. If the user-specified value for RETPD or EXPDT is less than the value that is specified in the retention limit, it is accepted and assigned to the data set. If the retention limit is set to `zero`, any RETPD or EXPDT value that the user assigns to a data set is overridden.

Interaction between expiration and retention

The following rules are followed for both tape and DASD data sets:

- If both expiration attributes are `NOLIMIT`, the data set never expires. The setting of RETPD has no impact.
- If the retention period is zero, neither of the settings for expiration has any impact and the data set will not expire.
- If the retention period is not zero and one of the expiration attributes is `NOLIMIT` and the other is specified, the specified value is used and the data set expires.
- If the date that is determined by the expiration values is sooner than the retention period, both expiration attributes must be satisfied. The data set expires on the latter of the dates that are determined by the expiration attributes.

Recommendations

In general, we recommend not using expiration dates for DASD data sets and setting RETPD to `zero` in management classes. This approach ensures that data sets are managed by DFSMShsm while they are cataloged, and that any EXPDT or RETPD values that are coded at data set allocation are ignored.

For DFSMShsm tape data sets, we recommend not specifying a retention period or expiration date unless your tape management system treats this date as a non-expiration flag. This topic is described more in Chapter 7, “Tape considerations” on page 117.

4.4.2 Partial release

This attribute determines whether DFSMShsm releases allocated but unused space for data sets that are assigned to this management class. This attribute is only relevant to non-Virtual Storage Access Method (VSAM) data sets and to extended format VSAM data sets.

The following values are valid for the partial release attribute:

N Partial release is not performed. `N` is the default value.

Y Space is released during primary space management.

C Partial release is performed only if a secondary allocation exists for the data set.

YI Partial release is performed either during primary space management or whenever the data set is closed.

CI Partial release is performed either during primary space management or whenever the data set is closed, but only if a secondary allocation exists for the data set.

If you plan to use partial release, consider using the options that occur only if a secondary allocation is present. If a data set with only a primary allocation is processed by partial release, any attempt to later extend the data set fails.

4.4.3 Migration

Your management classes define how DFSMShsm targets data sets during space management, determining what data sets are migrated and to which level they are migrated. You can control DFSMShsm migration processing by using management class attributes. Figure 4-4 shows the ISMF Management Class Alter panel where these attributes are defined.

```
DGTDCMC2                 MANAGEMENT CLASS ALTER               Page 2 of 5
Command ===>

SCDS Name . . . . . . : SYS1.SMS.MHLRES3.SCDS
Management Class Name : MC365

To ALTER Management Class, Specify:

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

  Migration Attributes
    Primary Days Non-usage  . . . . 10          (0 to 9999 or blank)
    Level 1 Days Non-usage  . . . . 30          (0 to 9999, NOLIMIT or blank)
    Command or Auto Migrate . . . . BOTH        (BOTH, COMMAND or NONE)


  GDG Management Attributes
    # GDG Elements on Primary . . . 2           (0 to 255 or blank)
    Rolled-off GDS Action . . . . . EXPIRE      (MIGRATE, EXPIRE or blank)

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
```

Figure 4-4 ISMF Management Class Alter panel

Primary days non-usage

This attribute determines how many days after a data set is not accessed, based on DS1LREFD, until a data set is eligible to be migrated. You can specify a value `0 - 9999`. A value for this field is required if the Command or Auto Migrate attribute is set to `Yes`. The default value is `2`.

Level 1 days non-usage

This setting determines whether data sets are migrated to DFSMShsm ML1 media, and if so, how long they remain on level 1 media. The default value is `60`. The following values are valid:

Zero
A `0` indicates no migration to DFSMShsm ML1. If migration occurs, data sets are migrated directly to DFSMShsm ML2.

A number of days 1 - 9999
The total number of days since the last access to the data set before the data set is eligible to be moved to ML2. This period includes any time that is spent on primary DASD before the data set was moved to ML1.

NOLIMIT Data sets will not move from ML1 to ML2 during automated DFSMShsm processing. They remain on ML1. Data sets can be moved with storage administrator commands, such as a `FREEVOL` command.

For more information about using the `FREEVOL` command, see “Converting level 1 DASD volumes” on page 334.

Command or Auto Migrate attribute

This attribute determines whether data sets are eligible to be processed by DFSMShsm automated functions or by command only. Commands can be generated either by storage administrators or users unless migration is prohibited for this data set. The possible values are `BOTH`, `COMMAND`, or `NONE`. If `NONE` is specified, data sets will not migrate and you can leave the “primary days non-usage” and “level 1 days non-usage” fields blank.

4.4.4 GDG management

The following generation data group (GDG) management criteria apply only for GDS data sets. They apply in addition to the normal migration attributes. You can specify the following attributes in ISMF. Figure 4-4 on page 67 shows the ISMF Management Class Alter panel where these options are defined.

GDG Elements on Primary attribute

This attribute determines the number of GDS data sets for the GDG base on the primary DASD. Additional GDS data sets are targeted by DFSMShsm for migration. Possible values are `0 - 255`, or `blank`. The default is `blank`. DFSMShsm targets older GDS data sets for migration. Older generations are migrated even if they do not meet the normal migration criteria for the management class.

Rolled-off GDS Action attribute

If a GDG is defined with the NOSCRATCH option, older GDS data sets will not be scratched as they are rolled out of the GDG base. Rolled-off GDS action applies only to those data sets. The following values are valid:

MIGRATE Rolled-off GDS data sets are targeted for migration by using the management class Migration attributes.

EXPIRE Rolled-off GDS data sets are expired.

None No action is taken.

If a GDG is defined with the SCRATCH attribute, these settings have no meaning.

4.4.5 Backup attributes

The following attributes of a management class are used by DFSMShsm backup processing:

- Backup Frequency
- Number of Backup Vers (Data Set Exists)
- Number of Backup Vers (Data Set Deleted)
- Retain days only Backup Ver (Data Set Deleted)
- Retain days extra Backup Vers
- Admin or User Command Backup
- Auto Backup
- Backup Copy Technique

Figure 4-5 shows the portion of the ISMF Management Class Alter panel where these values are specified.

```
DGTDCMC3                 MANAGEMENT CLASS ALTER               Page 3 of 5
Command ===>

SCDS Name . . . . . . : SYS1.SMS.MHLRES3.SCDS
Management Class Name : MC365

To ALTER Management Class, Specify:
  Backup Attributes
    Backup Frequency  . . . . . . . . 1        (0 to 9999 or blank)
    Number of Backup Vers . . . . . . 10       (1 to 100 or blank)
      (Data Set Exists)
    Number of Backup Vers . . . . . . 2        (0 to 100 or blank)
      (Data Set Deleted)
    Retain days only Backup Ver . . . 60       (1 to 9999, NOLIMIT or blank)
      (Data Set Deleted)
    Retain days extra Backup Vers . . 20       (1 to 9999, NOLIMIT or blank)
    Admin or User command Backup  . . BOTH     (BOTH, ADMIN or NONE)
    Auto Backup . . . . . . . . . . . Y        (Y or N)
    Backup Copy Technique . . . . . . S        (P, R, S, VP, VR, CP or CR)
```

Figure 4-5 ISMF Management Class Alter panel

Backup Frequency attribute

This attribute determines how many days can elapse before DFSMShsm backs up a changed data set. The default value is `1`. To back up a changed data set every backup cycle, specify the backup frequency as `zero`.

Number of Backup Vers (Data Set Exists) attribute

This attribute specifies the maximum number of backup copies for a data set that can exist while the source data set exists. The default value is `2`.

Number of Backup Vers (Data Set Deleted) attribute

This attribute specifies the maximum number of backup copies that can be kept when EXPIREBV processing determines that the source data set was deleted. For more information about EXPIREBV processing, see 11.1.6, “Expiring backup versions” on page 328. The default value is `1`.

Retain days only Backup Ver (Data Set Deleted) attribute

This attribute determines how long DFSMShsm keeps the most recent backup version of a data set when EXPIREBV processing determines that the original data set was deleted. If the value that is specified for “retain days extra backup versions” is `NOLIMIT`, all backup versions are kept for this period. The possible values for this field are a number of days (`1` - `9999`) or `NOLIMIT`. If `NOLIMIT` is set, backups are kept until they are manually deleted. The default value is `60`.

Retain days extra Backup Vers attribute

This field applies to other than the most recent backup copy of a data set. It determines how long to keep backup copies other than the most recent backup copy. The value is specified as a number of days (`1 - 9999`). Or, `NOLIMIT` can be specified, in which case no backup copies are expired until the number of backups exceeds the specified number of backup versions. The default value is `30`.

If a value other than `NOLIMIT` is specified, fewer than the expected number of backup versions for a fairly static data set are possible because older versions are expired and no new backup is created if the data set is not changed.

Admin or User command Backup attribute

This setting determines whether users, DFSMShsm administrators, or both, can explicitly issue data set level backups against the data sets. The following values are valid for this setting:

NONE Data set backup commands that are issued against data sets that are assigned to this management class are not processed.

ADMIN Data set backup commands that are issued against data sets by DFSMShsm storage administrators are processed. Commands from users are failed.

BOTH Both storage administrators and users can issue DFSMShsm backup commands against data sets that are assigned to this management class.

The default value is `BOTH`.

Auto Backup attribute

This parameter specifies whether or not DFSMShsm Autobackup processing targets data sets with this management class. The possible values are `Y` or `N`. `Y` is the default value.

Backup Copy Technique attribute

This parameter works together with the storage class accessibility value and the physical DASD where the data set exists. These values are used for both automatic and user-generated backup copies. Nine values are valid:

R Concurrent Required. Concurrent Copy must be used for data set backup. Backups fail if Concurrent Copy cannot be used.

P Concurrent Preferred. DFSMShsm first attempts to perform the backup by using Concurrent Copy. If this attempt fails, it then selects a non-Concurrent Copy technique to back up the data set.

S Standard. Data sets are backed up without using Concurrent Copy.

VP Virtual Preferred. Virtual Concurrent Copy is preferred. If virtual Concurrent Copy is not available, standard copy techniques are used.

VR Virtual Required. Virtual Concurrent Copy is to be used. If virtual Concurrent Copy fails, the data set backup is failed.

CP Cache Preferred. Data is processed with cache-based Concurrent Copy if available; otherwise, standard copy techniques are used.

CR Cache Required. Data is processed with cache-based Concurrent Copy if available; otherwise, the data set is not backed up.

The default value is `Standard`.

4.4.6 ABARS attributes

The following management class attributes apply only if you plan to use the DFSMShsm ABARs facility:

- # Versions
- Retain Only Version
- Retain Extra Versions
- Copy Serialization
- Abackup Copy Technique

Figure 4-6 is a sample ISMF Management Class Alter panel with specified ABARS attributes.

```
DGTDCMC7                  MANAGEMENT CLASS ALTER                 Page 5 of 5
Command ===>

SCDS Name . . . . . . : SYS1.SMS.MHLRES3.SCDS
Management Class Name : MC365

To ALTER Management Class, Specify:
  AGGREGATE Backup Attributes:
    # Versions . . . . . . . .          (1 to 9999, NOLIMIT or blank)
    Retain Only Version  . . .          (1 to 9999, NOLIMIT or blank)
      Unit . . . . . . . . . .          (D=days, W=weeks, M=months, Y=years or
                                        blank)
    Retain Extra Versions  . .          (1 to 9999, NOLIMIT or blank)
      Unit . . . . . . . . . .          (D=days, W=weeks, M=months, Y=years or
                                        blank)
    Copy Serialization . . . .          (C=continue, F=fail or blank)
    Abackup Copy Technique . . S        (P, R, S, VP, VR, CP or CR)



Use ENTER to Perform Verification; Use UP Command to View previous Panel;
```

Figure 4-6 ISMF Management Class Alter panel

You can define the following attributes. These attributes apply to the aggregate group, not to the data sets that are part of the group.

Versions attribute

This attribute determines the number of versions of an aggregate group that DFSMShsm retains. The field is not required. The specified value can be a number (1 - 9999) or NOLIMIT. By default, no value is specified.

Retain Only Version attribute

This attribute defines how long the only version of an Abackup is retained. The value is a number (1 - 9999) or NOLIMIT, which means that the most recent version of an aggregate is retained for an unlimited time. The default value is a blank, which results in the most recent or only version of an aggregate being expired. If you specify a numeric value, you can also specify a unit: days, weeks, months, or years that are applied to the numeric value. If a non-blank value is specified, “Retain Only Version, Unit” becomes a required field.

Retain Extra Versions attribute

This attribute defines how long aggregate versions other than the most recent version are retained. The value is a number (`1 - 9999`) or `NOLIMIT`, which means that the most recent version of an aggregate is retained for an unlimited time. The default value is a blank, which results in the most recent, or only version of an aggregate being expired. If you specify a numeric value, you can also specify a unit: days, weeks, months, or years that are applied to the numeric value. If a non-blank value is specified, “Retain Extra Versions, Unit” becomes a required field.

Copy Serialization attribute

The following values can be specified:

C Continue. Allows aggregate backup to continue if a requested shared enqueue cannot be obtained.

F Fail. Aggregate backup fails if a requested enqueue cannot be obtained.

Blank Aggregate backup fails.

Abackup Copy Technique attribute

This parameter specifies the copy technique to use during ABACKUP processing. This value is independent of the copy technique that is used for other DFSMShsm backups. Nine values are valid:

R Concurrent Required. Concurrent Copy must be used for data set backup. Backups fail if Concurrent Copy cannot be used.

P Concurrent Preferred. DFSMShsm first attempts to perform the backup by using Concurrent Copy. If Concurrent Copy fails, it selects a non-Concurrent Copy technique to back up the data set.

S Standard. Data sets are backed up without using Concurrent Copy.

VP Virtual Preferred. Virtual Concurrent Copy is preferred. If virtual Concurrent Copy is not available, standard copy techniques are used.

VR Virtual Required. Virtual Concurrent Copy is to be used. If virtual Concurrent Copy fails, the data set backup is failed.

CP Cache Preferred. Data is processed with cache-based Concurrent Copy if available; otherwise, standard copy techniques are used.

CR Cache Required. Data is processed with cache-based Concurrent Copy. If cache-based Concurrent Copy fails, the data set is not backed up.

The default value is `Standard`.

4.5 Storage class

Assigning a storage class to a DASD data set is the trigger for a data set to be SMS-managed. So, the act of assigning the storage class not only assigns the attributes of the storage class to the data set, but also affects how DFSMShsm processes the data set. When a data set is SMS-managed, the attributes that are assigned by the storage class, management class, and storage group data sets determine how DFSMShsm processes the data set rather than the DFSMShsm options that are established by using **`SETSYS`** commands.

Storage classes also specify several attributes that can directly affect how DFSMShsm processes a data set, for example:

- Guaranteed space
- Accessibility

Storage class attributes are defined in ISMF option 5. See Figure 4-7.

```
DGTDCSC1                     STORAGE CLASS ALTER             Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Class Name  : GSSTD
To ALTER Storage Class, Specify:
  Description ==> STORAGE CLASS FOR AVERAGE RESPONSE TIME
              ==>
  Performance Objectives
   Direct Millisecond Response . . . .             (1 to 999 or blank)
   Direct Bias . . . . . . . . . . . .             (R, W or blank)
   Sequential Millisecond Response . .             (1 to 999 or blank)
   Sequential Bias . . . . . . . . . .             (R, W or blank)
   Initial Access Response Seconds . .             (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .             (0 to 999 or blank)
   OAM Sublevel  . . . . . . . . . . .             (1, 2 or blank)
  Availability . . . . . . . . . . . . S           (C, P ,S or N)
  Accessibility  . . . . . . . . . . . S           (C, P ,S or N)
   Backup  . . . . . . . . . . . . . .             (Y, N or Blank)
   Versioning  . . . . . . . . . . . .             (Y, N or Blank)
```

Figure 4-7 ISMF Storage Class Alter panel

Guaranteed space

Guaranteed space allows hand placement of data sets. It also changes the way that primary and secondary space allocations are distributed on multiple volume data sets. Guaranteed space also permits pre-allocation of space for data sets. Primary space is allocated on each volume. After primary space on a specific volume is used, as many secondary extents as possible are added to the volume before space on a different volume is used.

If you are allocating data sets with the guaranteed space attribute, you might want to limit DFSMShsm space management of this data.

Accessibility

This attribute defines whether Concurrent Copy can be used for backups of data sets. The following values are valid:

C Continuous data sets must be allocated on volumes that support Concurrent Copy.

CP Continuous preferred data sets are to be allocated on volumes that support Concurrent Copy, if possible. If not, allocation is allowed on other volumes.

S Volumes that do not support Concurrent Copy are preferred for allocation.

N No preference. Concurrent Copy capability is not considered during allocation.

4.6 Values for SMS classes and groups

We do not describe absolute values in this section because requirements vary depending on the system. We describe several factors that you need to consider when you determine how much data to migrate and how many data set backups to take.

Several of these values need to be determined with the owners of the data that you are managing. Certain values are based on how the cost of data on both primary DASD and migrated or backed up data is passed back to users.

In a perfect world, data does not need to be migrated and an almost infinite number of backups can be kept. In the real world, the trick is to minimize the impact to users who have to wait while data sets are recalled while also minimizing the work that DFSMShsm performs. The number of backups created is a trade-off between users wanting to keep many backups and the space and cost of maintaining backups.

You might also be able to identify classes of data where backups or dumps taken by using DFSMShsm availability management might not be the best way to ensure that data is recoverable. For example, almost any database where multiple data sets are required to be backed up at a specific point in time are in this class.

If you run a multisite environment or if your disaster recovery depends on a second site, you need to consider whether both DFSMShsm migrated and backup data needs to be replicated to all sites.

4.6.1 Naming classes and groups

Where possible, we recommend meaningful names for classes and groups. Names are restricted to eight alphanumeric characters with the first character alphabetic. Consider that names for data classes, storage classes, and management classes can be specified by users. Therefore, select names that are meaningful to them. You are not required to indicate the type of class in the name of the class. Also, management, storage, and data class names can be identical.

SMS class names

Because users see the names that are assigned to their data sets, choose names that tell the user how their data will look or will be managed. Also, try to choose names that do not disparage the service that is provided. No one wants to see a class that is called “slow” assigned to their data.

If you allow users to hardcode class names in their JCL, the suggested data class names include the following names:

FB80S For RECFM=FB, LRECL=80 DSORG=PS

VB255PE For RECFM=VB, LRECL=255, DSORG=PO, DSNTYPE=LIBRARY

KS For a key-sequenced data set (KSDS) with no other specified attributes

TAPE3592 For tape data sets

You can also create certain data classes for specific requirements, for example:

LOGGER To ensure that IXGLOGR data sets get the required share options

DEFAULT To ensure that all allocated data sets get a complete data set control block (DSCB) and can therefore be managed by DFSMShsm

For storage classes, use names that describe the service that is provided, for example:

STANDARD A class for most data sets

GSSTD The same as standard but with the guaranteed space attribute

CDSRLS Used for record-level sharing (RLS)

NONSMS As a flag in the ACS routines to ensure that data is not SMS-managed

SMSTAPE To direct tape data sets to SMS

For management classes, also use names that describe how the data set is managed, for example:

STANDARD Most data sets

NOMIG Data sets that must not be processed by DFSMShsm space management

NOMGMT Data sets that must not be processed at all by DFSMShsm

PRODGDS Production GDS

Storage group names

Unlike SMS classes, users cannot directly specify the storage group name in the JCL, and storage group names are not externalized in allocation messages, for example, IGD101I. Therefore, storage group names must be chosen so that they are the most meaningful to storage administrators. Usually, storage group names are selected so they reflect the pool of DASD that is maintained in the storage group, for example:

PRIMARY The primary storage group

TAPE3592 A tape storage group

4.6.2 How many classes and groups

The number of classes and groups depends on the interaction between the different types of data that you might need to manage and the different DFSMShsm management criteria that you use for that data.

We recommend that you define as few classes and groups as possible, by using criteria for most data, and managing the rest by exception. SMS makes this process easier by enabling data with different management criteria to reside on the same volume. For example, data sets that must not be migrated can reside on volumes that are processed by DFSMShsm space management, that is, if the correct management class is assigned to them.

Depending on the number of DASD volumes that are managed, separating data into separate storage groups that are based on the data's management requirements might be beneficial. For example, separate a group of volumes that are not processed by availability management. However, you then need to ensure that no data sets are allocated on volumes in this storage group with management classes where backup is expected.

4.7 Automatic class selection

In addition to updating your automatic class selection (ACS) routines to ensure that the correct classes and groups are assigned to user data, you also need to ensure that DFSMShsm-owned data is managed correctly. For a description of the data sets that constitute DFSMShsm data, see 3.1, "DFSMShsm components" on page 30.

4.7.1 Coding for data that is owned by DFSMShsm and control data

Most data that is managed by DFSMShsm has names in fixed formats. These names can be used as filters in ACS routines to control the allocation of data sets.

The following filters are a starting point. The filters assume that the default high-level qualifier (HLQ) of hierarchical storage management (HSM) is used to manage DFSMShsm data sets. For more information and examples, see the *DFSMShsm Implementation and Customization Guide*, SC35-0418.

The first group of data sets is data that must be directed to non-SMS DASD. This data includes DFSMShsm backup and migration copies, VTOC copies, DUMP VTOC copies, and small-data-set packing (SDSP) data sets. See Example 4-1.

Example 4-1 Sample ACS filter list for non-SMS data

```
FILTLIST &HSM_NONSMS_DATA_DSN
INCLUDE(HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,HSM.MDB.**,
HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)
```

The next set of data sets consists of the HSM CDSs. These CDSs must be DASD resident but can be SMS-managed or non-SMS-managed. See Example 4-2.

Example 4-2 Sample ACS filter for DFSMShsm CDSs

```
FILTLIST &HSM_CONTROL_DATA INCLUDE(HSM.%CDS.*, HSM.JRNL.*
INCLUDE(HSM.%CDS.*,HSM.JRNL.*)
```

HSM logs are next. HSM logs include both the ARCLOG data sets and the HSM problem determination aid (PDA) data sets, and potentially data that is offloaded from these data sets. Again, these data sets can be SMS-managed or non-SMS-managed. If you decide that these HSM logs are not managed by SMS, they can be written to DFSMShsm ML1 volumes. Use the following sample ACS filter for DFSMShsm log data sets:

```
FILTLIST &HSM_LOG_DATA INCLUDE(HSM.EDITLOG?, HSM.HSMLOG*,HSM.HSMPDO.*)
```

If you are writing HSM activity logs to DASD by using `SETSYS ACTLOGTYPE(DASD)`, the following filter manages the activity log output data sets. If these data sets are used, they can be either managed by SMS or not. See Example 4-3.

Example 4-3 Sample ACS filter for DFSMShsm activity logs

```
FILTLIST &HSM_ACT_DATA_DSN INCLUDE
   (HSMACT.H%.*.D*.T*)
```

The next filter is for CDS backups. See Example 4-4.

Example 4-4 Sample ACS filter for DFSMShsm CDS backup copies

```
FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
(HSM.%CDS.BACKUP.*,
HSM.JRNL.BACKUP.*)
```

The last filter is for ABARs control files. See Example 4-5.

Example 4-5 Sample ACS filter for DFSMShsm ABARS CDSs

```
FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
(ABARS.*.INSTRUCT)
```

4.7.2 ACS execution for data that is managed by DFSMShsm

Coding is reviewed for user data, which is the data that DFSMShsm manages.

ACS environments

Execution of ACS routines in both the order that the routines are executed and the variables that are available for evaluation uses several environments. For a DFSMShsm environment, three ACS environments need to be considered. Other values for *&ACSENVIR* exist and are described in *z/OS DFSMS Storage Administration Reference,* SC35-7402. Each of these values is returned as a different value for the *&ACEENVIR* ACS variable. The values are shown:

ALLOC This value is the environment that is present when a data set is first allocated.

RECALL This environment is present when a data set is being recalled by DFSMShsm.

RECOVER This value exists when a data set is being recovered by DFSMShsm. If DFSMSdss is the data mover, ACSENVIR=`RECOVER` is passed to the storage group ACS routine for a data set recall.

For initial allocations, all of the data class ACS routine is executed, followed by the storage class routine. If no storage class is assigned, no further ACS routines are executed and the data set becomes non-SMS. If a storage class is assigned, the management class ACS routine is executed, followed by the storage group ACS routine. The management class ACS routine might assign a management class to a data set, but the storage group ACS routine must assign a storage group to a data set or the data set allocation fails. The class values that are set in previous ACS routines are available as read-only variables in subsequent routines.

When *ACSENVIR* is `RECOVER`, the storage class ACS routine is executed first, followed by the management class, and then the storage group routines. The data class ACS routine is not executed.

We do not recommend adding special code for changing ACS execution that is based on *ACSENVIR*. Instead, we recommend that you redrive ACS routines and allow ACS constructs to be redetermined regardless of the ACS environment. This way, even if classes change over time if a data set is recalled or recovered, the data set is treated according to your current policies.

ACS design

Designing your ACS routines is beyond the scope of this book. However, if you have a robust data set naming standard, use it to provide a framework for the decisions in your ACS routines. You can use your data set naming standard as the basis of ACS FILTERLISTs in your ACS routines.

If you have no data set naming standard or an installation with many naming standards, it might be easier to assign default values to profiles by using RACF or an equivalent function and enable these defaults to be used by setting PARMLIB IGDSMSxx ACSDEFAULTS(`YES`).

This method requires additional administration because new RACF profiles need to be provided with an SMS segment with the appropriate STORCLAS and MGMTCLAS values. You use the values that are passed in DEF_MGMTCLAS and DEF_STORCLAS as input to the ACS routines. However, because no default storage group is assigned, you cannot use the assigned values in the storage group ACS routine.

Testing changes

Whenever you change your ACS environment, you must test the changes before you activate them. ISMF provides options to both generate and run test cases. For more information, see *z/OS DFSMS Storage Administration Reference*, SC35-7402, and *Maintaining Your SMS Environment*, SG24-5484.

4.8 SMS and non-SMS configuration considerations

You might not define copypool storage groups for your installation and therefore none are eligible for processing. When DFSMShsm does not find any copypools, message ARC0570I is issued with return code 36, which indicates that the information is unavailable. It appears as though something is wrong with the configuration when nothing is wrong in the environment.

If you are not using copypool storage groups, you can use the **PATCH** command, which will stop this message from being issued. The following **PATCH** command turns message ARC0570I with return code 36 off:

```
F hsm,PATCH .MCVT.+297 BITS(.....1..) VERIFY(.MCVT.+297 BITS(.....0..))
```

Also, you might not define any storage groups that are eligible for processing. In this case, DFSMShsm issues an ARC0570I RC17 message, which can appear to be a problem when it is not. If you do not want to receive the ARC0570I RC17 message, use the following **PATCH** command:

```
F hsm,PATCH .MCVT.+297 BITS(....1...) VERIFY(.MCVT.+297 BITS(....0...))
```

You can reverse the suppression of these messages by using the following patches, as shown in Example 4-6.

Example 4-6 TURN ARC0570I MESSAGES ON

```
F hsm,PATCH .MCVT.+297 BITS(.....0..) VERIFY(.MCVT.+297 BITS(.....1..))
F hsm,PATCH .MCVT.+297 BITS(....0...) VERIFY(.MCVT.+297 BITS(....1...))
```

5

Implementing DFSMShsm security

The purpose of the implementation process is to select and implement options so that the resultant customized product can be used effectively. It is important to consider the modifications that must be applied to the existing system (for example, Resource Access Control Facility (RACF) and Data Facility Storage Management Subsystem (DFSMS)) as prerequisites before you start any implementation activity.

5.1 Providing security for DFSMShsm resources

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

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

5.1.1 Defining a DFSMShsm RACF user ID

For a started procedure to access any of the system resources, a user ID must be associated with that task. The user ID must have all of the correct authorizations for accessing the system resources. Therefore, it must be assigned the RACF OPERATIONS attribute. We recommend defining a new user ID for each started procedure rather than using the default started task user ID (for example, STCUSER). In our system, we associated user ID *HSM1* with the DFSMShsm started procedure, and user ID *ABR1* with the ABARS started procedure. We used the following RACF commands to define the DFSMShsm user IDs:

- `ADDUSER hsm1 OPERATIONS DFLTGRP(SYS1) NAME('ITSO DFSMShsm Userid')`
- `ADDUSER abr1 OPERATIONS DFLTGRP(SYS1) NAME('ITSO ABARS Userid')`

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

You can also create RACF user IDs by submitting a batch job by using the JCL that is shown in Figure 5-1.

```
//RACFADDU JOB ,'ADD STC RACF USERS',
//         NOTIFY=&SYSUID,
//         MSGCLASS=X,
//         CLASS=A
//*****************************************************************
//***  THIS JOB WILL CREATE THE DFSMSHSM AND ABARS STC USER IDS **
//*****************************************************************
//S0000010 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 ADDUSER HSM1 DFLTGRP(SYS1) NAME('DFSMSHSM STC USERID') OPERATIONS
 ADDUSER ABR1 DFLTGRP(SYS1) NAME('ABARS STC USERID') OPERATIONS
/*
```

Figure 5-1 Sample JCL to create a RACF user ID

5.1.2 Identifying started procedures to RACF

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

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

RACF STARTED class

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

- `RDEFINE STARTED (HSM1.*) UACC(NONE) STDATA(USER(HSM54) GROUP(SYS1))`
- `RDEFINE STARTED (ABR1.*) UACC(NONE) STDATA(USER(ABR54) GROUP(SYS1))`
- `SETROPTS RACLIST(STARTED) REFRESH`
- `SETROPTS GENERIC(STARTED) REFRESH`

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

Listing the STARTED class profiles

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

- `RLIST STARTED (HSM1.*)`
- `RLIST STARTED (ABR1.*)`

Figure 5-2 on page 82 shows the output of the first command.

```
READY
rlist started (hsm1.*)
CLASS NAME
----- ----
STARTED HSM1.* (G)
LEVEL OWNER UNIVERSAL ACCESS YOUR ACCESS WARNING
----- -------- ---------------- ----------- -------
00 MHLRES3 NONE NONE NO
INSTALLATION DATA
-----------------
NONE
APPLICATION DATA
----------------
NONE
AUDITING
--------
FAILURES(READ)
NOTIFY
------
NO USER TO BE NOTIFIED
```

Figure 5-2 Output of the RLIST command

ICHRIN03 started procedures table

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

The sample JCL in Figure 5-3 on page 83 shows how to identify the DFSMShsm and aggregate backup and recovery support (ABARS) started procedures to RACF by assembling ICHRIN03. You can enter a generic profile name for the started task table with an asterisk (*), this approach has limited flexibility. These limitations might affect how useful it can be in certain installations. For more information, see the *z/OS Security Server RACF System Programmer's Guide*, SA22-7681. Started class profiles provide the capability to define specific and appropriate authority flexibly and dynamically.

```
/ICHRIN03 JOB 'STARTED PROC TAB',MSGLEVEL=(1,1),REGION=4096K
//*****************************************************************//
//* ICHRIN03:                                                     *//
//*                                                               *//
//*   RACF INSTALLATION PROCEDURE STEP: CREATE/UPDATE          @03C*//
//*   STARTED PROCEDURES TABLE.                                @P2C*//
//*                                                               *//
//* NOTE: Please read the section titled "Coding the Started      *//
//*       Procedures Module" in the System Programming            *//
//*       Library: RACF.  It is important that you are familiar   *//
//*       with this section before attempting to use or modify    *//
//*       this sample.                                            *//
//*                                                               *//
//*****************************************************************//
//STEP1     EXEC HLASMCL PARM.L=(RENT,XREF,LIST,LET,NCAL)
//*                                                           @04C
//C.SYSIN DD *
ICHRIN03 CSECT
         TITLE 'ICHRIN03 - STARTED PROCEDURES TABLE'
         EJECT
         DC    XL2'80xx'          CHANGE 'xx' TO THE HEX VALUE FOR
                                  THE NUMBER OF ENTRIES IN THE TABLE
*
         DC    CL8'HSM1   '      PROCNAME - SPECIFY YOUR DFSMShsm
         DC    CL8'HSM1   '      STARTED TASK USER ID
         DC    CL8'SYS1    '      GROUP
         DC    XL1'00'            NOT PRIVELEDGED OR TRUSTED
         DC    XL7'00'            RESERVED
*
         DC    CL8'ABR1    '      PROCNAME - SPECIFY YOUR ABARS
         DC    CL8'ABR1    '      STARTED TASK USER ID
         DC    CL8'SYS1    '      GROUP
         DC    XL1'00'            NOT PRIVELEDGED OR TRUSTED
         DC    XL7'00'            RESERVED
*
                  .                   .
                  .                   .
                  .                   .
                  .                   .
         END
/*
//L.SYSLMOD DD DSN=SYS1.LPALIB,            ** MUST BE LIBRARY WITH**
//             DISP=SHR,UNIT=YYYY,         ** NEW RELEASE OF RACF **
//             VOL=SER=XXXXXX
//L.SYSIN   DD *
       ENTRY ICHRIN03
       NAME ICHRIN03(R)
/*
//*
```

Figure 5-3   ICHRIN03 sample source

5.1.3 Protecting DFSMShsm data sets

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

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

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

You can give user ID HSMSTOR ALTER access to the DFSMShsm CDSs. You can use a different qualifier for CDSs, such as HSMCDS, so that you can provide access to only those data sets with the following command:

```
PERMIT 'HSMCDS.**' ID(HSMSTOR) ACC(ALTER)
```

Listing the generic data set profile

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

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

Figure 5-4 shows the output of the command.

```
READY
INFORMATION FOR DATASET HSM.** (G)
LEVEL OWNER UNIVERSAL ACCESS WARNING ERASE
----- -------- ---------------- ------- -----
00 SYS1 NONE NO NO
AUDITING
--------
FAILURES(READ)
NOTIFY
--------
NO USER TO BE NOTIFIED
YOUR ACCESS CREATION GROUP DATASET TYPE
----------- -------------- ------------
ALTER SYS1 NON-VSAM
NO INSTALLATION DATA
```

Figure 5-4 LISTDSD command output

Protecting DFSMShsm activity logs

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

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

After you create the RACF generic profile for protecting the DFSMShsm activity logs, you must permit users or groups access to the RACF profile based on their requirements. Consider that someone might use a batch job to access data in the activity logs.

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

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

Protecting DFSMShsm tapes

To protect DFSMShsm-managed tapes with RACF-protected data sets on them, follow these steps:

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

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

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

Protecting up to 10,000 tapes

If you are protecting up to a maximum of 10,000 tapes, you define two RACF resource names in the TAPEVOL resource class:

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

Issue the following RACF commands:

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

You can add tapes to RACF before DFSMShsm uses them. If you choose this approach, you must use the appropriate TAPEVOL resource class. Use the following command:

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

IBM suggests that clients *not add tapes to RACF before DFSMShsm uses them*. Instead, let DFSMShsm add to the TAPEVOL for you automatically as tapes are encountered.

Protecting more than 10,000 tapes

To RACF-protect more than 10,000 tapes, you define multiple RACF resource names for DFSMShsm tape volume sets in the TAPEVOL resource class. Use the following resource names:

- HSMHSM (must be defined)
- HSMABR for ABARS tapes
- DFHSM*x*

In DFHSM*x*, *x* is a non-blank character (alphanumeric, national, or the hyphen) that corresponds to the last non-blank character of the tape volume serial number. You need to define a DFHSM*x* resource name for each *x* value that exists, based on your installation naming standards.

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

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

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

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

You can add RACF protection to the DFSMShsm tape volumes before DFSMShsm uses them, except for the HSMABR tapes. You must add the tape volume serial number to the appropriate DFHSMx tape volume set, which is based on the last non-blank character of the tape volume serial number. To protect a tape with a volume serial of `POK33H`, you use the following RACF command:

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

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

5.1.4 Controlling the use of DFSMShsm operator-entered commands

No RACF authorization checking is performed for DFSMShsm commands that are entered at the system console if LOGON(OPTIONAL) is specified for consoles in the CONSOLxx PARMLIB member. In this case, the authority of the AUTH attribute is used. Consoles with master authority can enter any command.

To protect DFSMShsm commands that are entered from a console, two levels of authorization processing are required. When commands are entered at the console, they are entered in the form of a **`MODIFY`** command against the DFSMShsm task or job name. The console address space can check whether the operator has authority to enter this command against an OPERCMDS profile. However, this check requires that the console has a RACF user ID that is associated with it. In CONSOLxx member, LOGON(AUTO) or LOGON(REQUIRED) must be specified.

For the first check, the RACF OPERCMDS class is used by the console address space COMTASK to verify that the user ID of the console has authority to issue the **`MODIFY`** command against DFSMShsm. The second check is then performed by DFSMShsm to see whether the user ID that is passed is authorized for the relevant FACILITY class profile for the command. Alternatively, this second check can be performed by using the AUTH table constructed from the **`AUTH`** commands in the ARCCMDxx member, although this approach is not recommended. RACF is preferred.

Associating a user ID with a console

You can use a RACF profile in the CONSOLE class to determine which user IDs are authorized to log on to a particular console. The commands in Example 5-1 on page 87 define a RACF profile for console CON1 (CON1 is defined in CONSOLxx), and authorize user ID `CONSID1` to log on to that console.

Example 5-1 Defining consoles and authorizing users

```
RDEF CONSOLE CON1 UACC(NONE)
PERMIT CON1 CLASS(CONSOLE) ID(CONSID1) ACCESS(READ)
PERMIT CON1 CLASS(CONSOLE) ID(OPERGRP) ACCESS(READ)
SETROPTS CLASSACT(CONSOLE)
```

OPERCMDS profile for the MODIFY command

Table 5-1 lists the profile names to use to authorize the **MVS MODIFY** command, depending on whether the target is a job, started task, or user. DFSMShsm can be run as a job or a started task. The first row is for a job. The second row is for a Time Sharing Option (TSO) user (not relevant in this instance for DFSMShsm). The last two rows are for a started task, depending on whether it is started with an identifier, for example, `START DFHSM.HSM`, or just `START DFHSM`.

Table 5-1 OPERCMDS MODIFY command resource names

Command	Authority	Resource
MODIFY *jobname*	UPDATE	MVS.MODIFY.JOB.*jobname*
MODIFY *userid*	UPDATE	MVS.MODIFY.JOB.*userid*
MODIFY *jobname* **MODIFY *jobname.id*** **MODIFY *id***	UPDATE	MVS.MODIFY.STC.*member.id*
MODIFY *jobname*	UPDATE	MVS.MODIFY.STC.*member.* *jobname*

For the example that is shown in Example 5-2, assume that DFSMShsm is run as a started task named *DFHSM*, and its JCL is in a member called *DFHSM*. Our console, CON1, is defined with LOGON(AUTO) and so is given authority to issue the **MODIFY** command against DFHSM. In addition, an operator with the user ID of CONSID1 and members of the OPERGRP are also permitted access.

Example 5-2 Creating profiles to allow the DFSMShsm MODIFY command

```
SETROPTS GENERIC(OPERCMDS) REFRESH
REDFINE OPERCMDS MVS.MODIFY.STC.DFHSM.DFHSM UACC(NONE)
PERMIT MVS.MODIFY.STC.DFHSM.DFHSM CLASS(OPERCMDS) ID(CON1) ACCESS(UPDATE)
PERMIT MVS.MODIFY.STC.DFHSM.DFHSM CLASS(OPERCMDS) ID(OPERGRP) ACCESS(UPDATE)
PERMIT MVS.MODIFY.STC.DFHSM.DFHSM CLASS(OPERCMDS) ID(CONSID1) ACCESS(UPDATE)
SETROPTS CLASSACT(OPERCMDS)
SETROPTS RACLIST(OPERCMDS) REFRESH
SETROPTS GENERIC(OPERCMDS) REFRESH
```

When the OPERCMDS class is activated

If RACF is active, MVS uses the z/OS CMDAUTH function, that is, OPERCMDS profiles, to verify that the operator console that is used to issue the **DFSMShsm** command is authorized to issue that **DFSMShsm** command.

If RACF is not active at DFSMShsm startup, MVS does not verify any commands that are issued from an operator console. In this case, authorization falls back to the user IDs that are given authority by using the **AUTH** command in ARCCMDxx.

If you stop RACF while DFSMShsm is active, MVS fails all DFSMShsm commands that are issued from an operator console.

5.1.5 Authorizing and protecting commands by using AUTH

We recommend that you use RACF to protect DFSMShsm commands and resources. However, we include information about the DFSMShsm **AUTH** command for completeness. If DFSMShsm is initialized before RACF, or RACF or another external security software product is not present, authorities that are granted by the **AUTH** command in ARCCMDxx will be used.

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

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

CONTROL authority

The following command allows user ID ITSOHSM to add, delete, or change the DFSMShsm authorization of other users. This command can be placed in the ARCCMDxx member of PARMLIB or issued by a user with the database control attribute. The following **AUTH** command grants `CONTROL` authority:

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

USER authority

The following command allows user ID ITSOHSM to issue authorized commands except for the **AUTH** command. This command can be placed in the ARCCMDxx member of PARMLIB or issued by a user with the database control attribute. This **AUTH** command grants `USER` authority:

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

Revoking authority

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

```
AUTH ITSOHSM REVOKE
```

5.1.6 Authorizing and protecting commands by using RACF

DFSMShsm uses the following set of the RACF FACILITY class profiles to protect commands:

- `STGADMIN.ARC.command`: The DFSMShsm storage administrator command protection is a profile for a specific DFSMShsm storage administrator command.
- `STGADMIN.ARC.command.parameter`: This profile protects a specific DFSMShsm administrator command with a specific parameter.
- `STGADMIN.ARC.ENDUSER.h_command`: This profile is for DFSMShsm user command protection. This profile protects a specific DFSMShsm user command.
- `STGADMIN.ARC.ENDUSER.h_command.parameter`: This profile protects a specific DFSMShsm user command with a specific parameter.

For a complete list of commands and parameters that you can protect, see "Protecting DFSMShsm storage administrator commands with RACF FACILITY class profiles" in the *z/OS DFSMShsm Implementation and Customization Guide*, SC35-0418.

The RACF commands in Example 5-3 show how to grant broad access to the storage administrators group to issue all commands by using a generic profile.

Example 5-3 Granting access to all commands

```
SETROPTS CLASSACT(FACILITY)
SETROPTS GENERIC(FACILITY)
SETROPTS RACLIST(FACILITY)
REDFINE FACILITY STGADMIN.ARC.* UACC(NONE)
PERMIT STGADMIN.ARC.* CLASS(FACILITY) ID(STGADGRP) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
SETROPTS GENERIC(FACILITY) REFRESH
```

After you define the generic profile and permitted access, you can display it by using the **RLIST** RACF command. Figure 5-5 on page 90 shows a sample listing.

Other groups and users will require various accesses and each command can be specifically permitted and restricted. Section 5.1.7, "RACF protection for ABARS" on page 91 illustrates the use of more specific profiles to protect the ABARS commands.

Figure 5-5 shows how to display the STGADMIN.ARC.* profile.

```
RL FACILITY (STGADMIN.ARC.*) ALL
CLASS      NAME
-----      ----
FACILITY   STGADMIN.ARC.* (G)

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

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

APPLICATION DATA
----------------
NONE
SECLEVEL
--------
NO SECLEVEL

CATEGORIES
----------
NO CATEGORIES

SECLABEL
--------
NO SECLABEL

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

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

CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
 (DAY) (YEAR)       (DAY) (YEAR)      (DAY) (YEAR)
-------------  -------------------  ----------------
  205    03          205    03         205    03

ALTER COUNT   CONTROL COUNT   UPDATE COUNT   READ COUNT
-----------   -------------   ------------   ----------
NOT APPLICABLE FOR GENERIC PROFILE

USER      ACCESS
----      ------
MHLRES3   READ
MHLRES1   READ
MARY      ALTER
```

Figure 5-5 Listing the STGADMIN.ARC. profile*

5.1.7 RACF protection for ABARS

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

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

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

The RACF FACILITY class profiles have names that begin with `STGADMIN` (storage administration). These FACILITY profiles are used to protect ABARS functions and many other storage management subsystem (SMS) functions.

Define the profiles for comprehensive command authority with the following RACF commands, as shown in Example 5-4.

Example 5-4 Defining FACILITY profiles to protect ABARS commands

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP UACC(NONE)
RDEFINE FACILITY STGADMIN.ARC.ARECOVER UACC(NONE)
```

The following command authorizes a user (MHLRES5) to issue the **ABACKUP** command for all aggregate groups:

```
PERMIT STGADMIN.ARC.ABACKUP CLASS(FACILITY)
   - ID(MHLRES5) ACCESS(READ)
```

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

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP.ITSOU001 UACC(NONE)
```

Authority to issue an **ABACKUP** command for aggregate ITSOU001 is given to user MHLRES5. The following command permits access to the **ABACKUP** command for a specific aggregate:

```
PERMIT STGADMIN.ARC.ABACKUP.ITSOU001 CLASS(FACILITY)
   - ID(MHLRES5) ACCESS(READ)
```

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

As with the **ABACKUP** commands, **ARECOVER** commands can also be restricted with a profile for each aggregate, STGADMIN.ARC.ARECOVER.*agname*. The use of DSCONFLICT(REPLACE), REPLACE as a conflict resolution data set action, or REPLACE when it is specified by ARCCREXT, can also be restricted by using the RACF FACILITY class profile STGADMIN.ARC.ARECOVER.*agname*.REPLACE. The following command shows how to define the profile to protect **ARECOVER** for an aggregate and the use of **REPLACE**:

```
RDEFINE FACILITY STGADMIN.ARC.ARECOVER.ITSOU001.REPLACE UACC(NONE)
```

5.1.8 RACF protection for Concurrent Copy

DFSMShsm uses the STGADMIN.ADR.DUMP.CNCURRNT FACILITY class to authorize the use of the Concurrent Copy options on the data set **backup** commands. Checking for authorization occurs before DFSMSdss is invoked. If RACF indicates a lack of authority, DFSMShsm will only fail the data set backup request if the Concurrent Copy request was `REQUIRED`. If `REQUIRED` was not specified and RACF indicates a lack of authority, DFSMShsm continues to back up the data set as though the Concurrent Copy keyword was not specified on the **backup** command.

Define the profiles for comprehensive command authority with the following RACF command. This command protects Concurrent Copy:

```
RDEFINE FACILITY STGADMIN.ADR.DUMP.CNCURRNT UACC(NONE)
```

The following command authorizes a user to issue the **backup** or **dump** command with Concurrent Copy options:

```
PERMIT STGADMIN.ADR.DUMP.CNCURRNT CLASS(FACILITY)
   - ID(MHLRES5) ACCESS(READ)
```

5.1.9 ARCCATGP group

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

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

   ```
   ADDGROUP (ARCCATGP)
   ```

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

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

3. Each user who needs to perform catalog operations without automatic recall must log on to TSO and specify the **GROUP(ARCCATGP)** parameter on the TSO logon window (Figure 5-6) or the **GROUP=ARCCATGP** parameter on the JOB statement of a batch job (Figure 5-7 on page 93).

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

   Figure 5-6 JCL specifying RACF group ARCCATGP

```
------------------------------ TSO/E LOGON -------------------------------


   Enter LOGON parameters below:                  RACF LOGON parameters:

   Userid    ===> MHLRES5

   Password  ===>                                 New Password ===>

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

   Acct Nmbr ===> ACCNT#

   Size      ===> 6072

   Perform   ===>

   Command   ===>

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

PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==>
Reshow
You may request specific help information by entering a '?' in any entry
field
```

Figure 5-7 TSO/E LOGON panel

5.1.10 Protecting migration and backup data sets

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

The migration copy of a data set has the following name:

prefix.HMIG.Tssmmhh.user1.user2.Xydd

In this name, *prefix* is the prefix that you specify on the **SETSYS MIGRATEPREFIX** command.

The backup version of a data set has the following name:

prefix.BACK.Tssmmhh.user1.user2.Xydd

In this name, *prefix* is the prefix that you specify on the **SETSYS BACKUPPREFIX** command.

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

In our system, migrated data sets have the prefix *HSM*. Use the following RACF command to define a generic data set profile to protect all migrated data sets with a universal access of `NONE`. The following command defines generic profile protection of migration and backup data sets:

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

After you create the RACF generic profile to protect all migrated data sets, you must permit users access to the RACF profile based on their requirements.

6

Data set format support, record-level sharing, and extended address volumes

In this chapter, we describe DFSMShsm and storage management subsystem (SMS) considerations for *record-level sharing* (RLS) and *extended address volumes* (EAVs).

6.1 Record-level sharing

Record-level sharing (RLS) offers an advantage because it allows sharing with a finer granularity, which reduces contention as the number of systems sharing resources in sysplexes increases.

6.1.1 VSAM record-level sharing

DFSMShsm supports Virtual Storage Access Method (VSAM) RLS for accessing the control data sets (CDSs). RLS enables DFSMShsm to take advantage of the features of the coupling facility for CDS access.

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

Requirements for CDS RLS serialization

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

- Global resource serialization (GRS) or an equivalent function is implemented.

  Note: This function is required only if you are sharing CDSs between multiple systems. If you have one or more hosts on a single system, it is not required.

- Your CDSs must be SMS-managed.
- All processors in the installation must access the CDSs in RLS mode.
- All DFSMShsm hosts must specify CDSSHR=RLS in the DFSMShsm startup procedure.
- The CDS storage class must indicate which coupling facility to use.
- The CDSs must not be key range key-sequenced data sets (KSDSs).
- You must know how to implement recovery for RLS data sets.
- You must specify DFSMSdss as the data mover for CDSVERSIONBACKUP.
- If CDS backup is directed to tape, the **PARALLEL** parameter must be used.

Making your CDSs RLS eligible

Before CDSs can be accessed in RLS mode, you must define or alter them to be RLS eligible, by using the LOG(NONE) attribute. You must define or alter all of your CDSs: migration control data sets (MCDSs), backup control data sets (BCDSs), and offline control data sets (OCDSs).

The following example shows how to use the **IDCAMS ALTER** command to make the CDSs that we defined previously RLS eligible:

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

Figure 6-1 is an example of how we might use the **DEFINE** command when we initially set up our DFSMShsm environment. We show the definition for the MCDS only, but the same method is used for the other CDSs.

```
DEFINE CLUSTER (NAME(HSM.MCDS) VOLUMES(HG6622) -
        CYLINDERS(100) FILE(HSMMCDS) -
        RECORDSIZE(435 2040) FREESPACE(0 0) -
        INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
        SPEED BUFFERSPACE(530432) -
        UNIQUE NOWRITECHECK LOG(NONE)) -
        DATA(NAME(HSM.MCDS.DATA) -
        CONTROLINTERVALSIZE(12288)) -
        INDEX(NAME(HSM.MCDS.INDEX) -
CONTROLINTERVALSIZE(4096))
```

Figure 6-1 Sample DEFINE command for a DFSMShsm environment

You must never use the **ALL** or **UNDO** parameters of the **LOG** keyword. If you ever need to change the CDSs back to non-RLS-eligible, use the following command:

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

Removing key range CDSs

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

Determining the CDS serialization technique

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

If you use RLS, the following RLS messages are returned from the **QUERY CONTROLDATASETS** command:

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

6.2 RLS implementation

The steps that we took to implement VSAM RLS for our CDSs are shown. In addition to the requirements that we detailed in “Requirements for CDS RLS serialization” on page 96, other considerations must be met and assumptions are made about your system knowledge. You need to be familiar with SMS for VSAM RLS, SMS constructs, SMS classes, SMS configuration, and the coupling facility cache and lock structures. We do not recommend that you undertake these steps until you consider how RLS implementation might affect your system.

It is recommended that any pre-existing VSAM RLS structures be used for accessing the DFSMShsm CDSs in RLS mode. Assigning the DFSMShsm CDSs to unique structures offers no benefit.

6.2.1 Defining the SHCDSs

Sharing Control Data Sets (SHCDSs) are linear data sets that contain information to allow processing if a system failure might affect RLS. They also act as logs for sharing support. You must consider SHCDS size and adhere to a naming convention for the SHCDSs. For comprehensive information about defining these data sets, see *OS/390® DFSMSdfp Storage Administration Reference,* SC26-7331. We used the JCL in Figure 6-2 to allocate the SHCDS.

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

Figure 6-2 Sample SHCDS allocation JCL

6.2.2 Coupling facility cache and lock structures

If you need to allocate cache and lock structures specifically for DFSMShsm, use the following recommendations:

- Cache structure: The size of the cache structure must be a minimum of 1 MB per DFSMShsm host in the HSMplex. For example, if 10 hosts are in the HSMplex, the cache structure needs to be a minimum of 10 MB.
- Lock structure: Use Table 6-1 to determine the minimum size for the lock structure.

Table 6-1 Minimum size for lock structure

Number of DFSMShsm hosts	Minimum lock structure size
Fewer than 8	1 MB
At least 8, but not more than 23	2 MB
At least 24, but not more than 32	3 MB
More than 32	4 MB

For example, if 10 DFSMShsm hosts are in the HSMplex, the lock structure must be a minimum of 2 MB.

Certain changes to your "define coupling facility resource management (CFRM)" policies are necessary. The DFSMShsm policy needs to be defined. We added the structures for cache and locking to our current CFRM policy definitions, by using the administrative data utility IXCMIAPU. See Figure 6-3 on page 99.

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

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

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

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

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

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

Figure 6-3 CFRM policy definition for RLS access of the DFSMShsm CDSs

Note: The code in Figure 6-3 does not represent the entire policy data for the CFRM data set. It represents the CFRM policy that specifies the requirements for the DFSMShsm RLS structures.

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

6.2.3 Altering the SMS configuration

You must update the SMS configuration with the coupling facility cache structure names that you defined earlier. Use the Interactive Storage Management Facility (ISMF) panels. Enter option `8` from the ISMF Primary Option menu for Storage Administrators to display the CDS Application Selection panel (Figure 6-4).

```
                        CDS APPLICATION SELECTION
Command ===>

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

Select one of the following Options:

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

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

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

Figure 6-4 CDS Application Selection panel for cache

We entered option `7` to define the cache sets that relate to our coupling facility cache structure names (Figure 6-5 on page 101).

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

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

 Cache Set                      CF Cache Structure Names
 HSM1      HSMCACHE1

 HSM2      HSMCACHE2









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

Figure 6-5   CF Cache Set Update panel

The coupling facility cache structure names must be the same names that we previously defined. (See 6.2.2, “Coupling facility cache and lock structures” on page 98.)

6.2.4 Storage class changes

We opted to alter the storage class where our CDSs were already defined. The storage class is SC54GRT, and we altered it by selecting option **4** from the Storage Class Application Selection panel (Figure 6-6 on page 102).

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

To perform Storage Class Operations, Specify:
  CDS Name  . . . . . . . 'SYS1.SMS.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Storage Class Name  . . SC54GRT (For Storage Class List, fully or
                                    partially specified or * for all)
Select one of the following options  :
  4  1. List          - Generate a list of Storage Classes
     2. Display       - Display a Storage Class
     3. Define        - Define a Storage Class
     4. Alter         - Alter a Storage Class
     5. Cache Display - Display Storage Classes/Cache Sets
If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

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

Figure 6-6 Altering a storage class

We added the coupling facility cache set information (Figure 6-7).

```
 Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
-
                            STORAGE CLASS ALTER               Page 2 of 2
 Command ===>

 SCDS Name . . . . . : SYS1.SMS.SCDS
 Storage Class Name  : SC54GRT

 To ALTER Storage Class, Specify:

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

Figure 6-7 Storage Class Alter panel

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

Because the CDSs were already in this storage class, we validated and then activated our SMS Source Control Data Set (SCDS) of SYS1.SMS.SCDS1.

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

6.2.5 Altering IGDSMSxx PARMLIB member

To specify to SMS that the RLS address space, SMSVSAM, starts at initial program load (IPL) and to include other information, we added the following statements that are shown in Figure 6-8 to our PARMLIB data set, member IGDSMS54.

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

Figure 6-8 Sample IGDSMSxx PARMLIB to implement RLS

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

6.2.6 Activating the SMSVSAM address space

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

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

The information that is returned states whether your system is active and the status of the SMS complex.

6.2.7 Activating the SHCDS

If your SHCDS is already active, you do not need to activate it. However, if you are implementing RLS for the first time, you must issue the commands that are shown in Figure 6-9, which are modified to represent the names that we chose for our SHCDSs.

```
VARY SMS,SHCDS(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS1),NEW
VARY SMS,SHCDS(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS2),NEW
VARY SMS,SHCDS(SYS1.DFPSHCDS.WTSCPLX1.VSHCDS3),NEWSPARE
```

Figure 6-9 Activating the SHCDSs

The commands that are shown in Figure 6-9 perform the following functions:

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

6.2.8 DFSMShsm procedure changes

To your DFSMShsm started procedure member in SYS1.PROCLIB, you must add **`CDSSHR=RLS`** after the PROC statement.

You must also add **`'CDSSHR=&CDSSHR`** to your EXEC statement.

6.3 RLS implementation checklist

To review, follow these steps to implement RLS access for the DFSMShsm CDSs:

1. Ensure that all sharing systems are at the prerequisite levels of software.
2. If SHCDSs are not already defined, use IDCAMS to define SHCDSs and ensure that SHAREOPTION(3,3) is specified.
3. Define your coupling facility cache and lock structures.
4. Alter your SMS configuration to include the cache set information.
5. Create or alter a storage class that contains the cache set information.
6. Assign the CDSs to the new or altered storage class.
7. Alter the SYS1.PARMLIB member IGDSMSxx with the RLS parameters.
8. Define the new RLS profiles in the RACF FACILITY class and authorize users.
9. Schedule an IPL of the MVS systems and ensure that the SMSVSAM address space becomes active.

6.3.1 DFSMShsm support for RLS-managed ICF catalogs

z/OS V 2.1 introduced RLS support for integrated catalog facility (ICF) catalogs. ICF catalogs are now basically enqueued at a record level, instead of enqueue happening at the entire catalog (SYSIGGV2), which will serialize catalog access. RLS support gives better accessibility and performance through record-level access.

When DFSMShsm invokes DFSMSdss for recovering an ICF catalog, a new **BCSRECOVER(LOCK)** keyword is used on the **RESTORE** command. This command performs a serialized LOCK on the ICF catalog in process, if it is not already locked or suspended. When restore completes successfully, DFSMSdss issues an UNLOCK of the ICF catalog to enable normal processing.

Note: ICF catalogs are still not supported by migration or recall.

For aggregate backup and recovery support (ABARS), you can specify RLS-managed ICF catalog names in the ALLOCATE list and DFSMShsm will process them. For the ABARS INCLUDE and ACCOMPANY list, ICF catalogs are not supported.

For non-RLS-managed ICF catalogs, DFSMShsm still invokes IDCAMS EXPORT/IMPORT for processing ICF catalogs.

Toleration authorized program analysis report (APAR) OA36414 was created for z/OS V10, z/OS V11, z/OS V12, and z/OS V13. The program temporary fix (PTF) and its prerequisites pointed to in the APAR need to be applied to lower-level systems in the sysplex.

After this support is installed, DFSMShsm can restore RLS-managed ICF catalogs by invoking DFSMSdss on pre-z/OS V2.1 hosts. The RLS subcell will be preserved and the RLS state will be restored as RLS is quiesced.

6.3.2 RLS considerations

Because DFSMShsm uses RLS for serialization of its CDSs, it is important to ensure that SMSVSAM completely initializes before you start DFSMShsm. Likewise, DFSMShsm needs to be brought down before SMSVSAM.

If the SMSVSAM server encounters an unusual error and terminates, DFSMShsm terminates, also. Review the information in “z/OS DFSMS V2.1 VSAM record-level sharing error recovery” on page 417 for a description of how DFSMShsm handles SMSVSAM server failure.

6.4 EAV support, considerations, and coexistence

Extended address volumes (EAVs) were implemented in three releases: z/OS V1.10 supported EAV R1, z/OS V1.11 supported EAV R2, and z/OS V1.12 was enhanced to support EAV R3. z/OS V.13 includes extra support to increase EAV capacity from 223 GB (262,668 cylinders) to 1 TB (1,182,006 cylinders). This 1 TB support was rolled back to z/OS V.12 with an SPE (OA28553).

The support that DFSMShsm currently provides for EAV is described. EAV concepts are introduced, and then the particular support features of DFSMShsm for EAV are described.

6.4.1 DFSMShsm EAV support and considerations

EAVs require hardware and software. Due to the limit of 64 K devices per channel subsystem (CSS), large installations still required more storage space. The introduction of logical channel subsystems (LCSSs) helped to increase the capacity by double and quadruple, but EAVs increase this capacity by potentially 4,000-fold, which is based on a maximum theoretical capacity of 225 TB. The maximum EAV with DFSMS V1.13 is 1 TB.

Extended address volumes

EAVs require IBM DS8000® R4 direct access storage device (DASD) subsystems so that EAVs can be defined, as shown in Figure 6-10 on page 106. For EAVs of 1 TB, DS8800 or DS8700 subsystems and z/OS V1.12 or later are required.

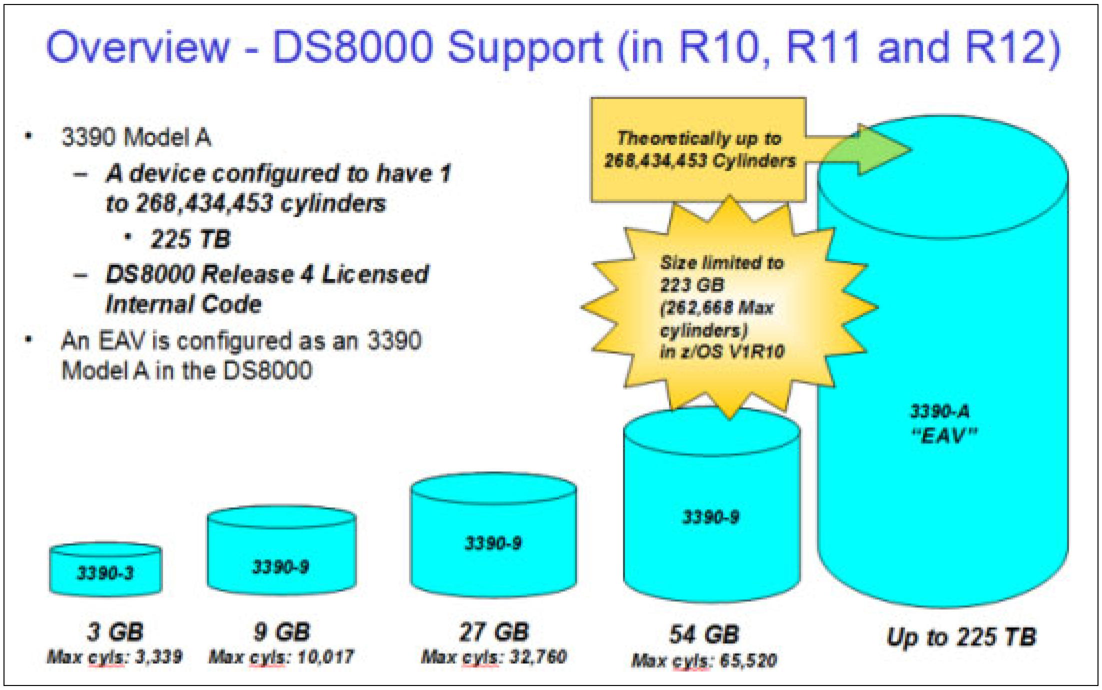

Figure 6-10 EAV DASD model on DS8000 R4

An EAV is divided into two parts: track-managed space for the first 64 K cylinders and cylinder-managed space beyond 64 K cylinders. Cylinder-managed space is in an area that is called the *extended addressing space* (EAS). Cylinder-managed space is allocated in multicylinder units (MCUs), which are 21 cylinders each. Therefore, the smallest data set size that uses cylinder-managed space is 21 cylinders. This size has implications for space usage. It can affect the allocation of data sets, which also applies to DFSMShsm and how it uses EAVs. See Figure 6-11 on page 107.

Important: Only the EAS of the EAV requires any special support. Even data sets that are not EAS-eligible can still be allocated on an EAV in track-managed space that is within the first 54 GB.

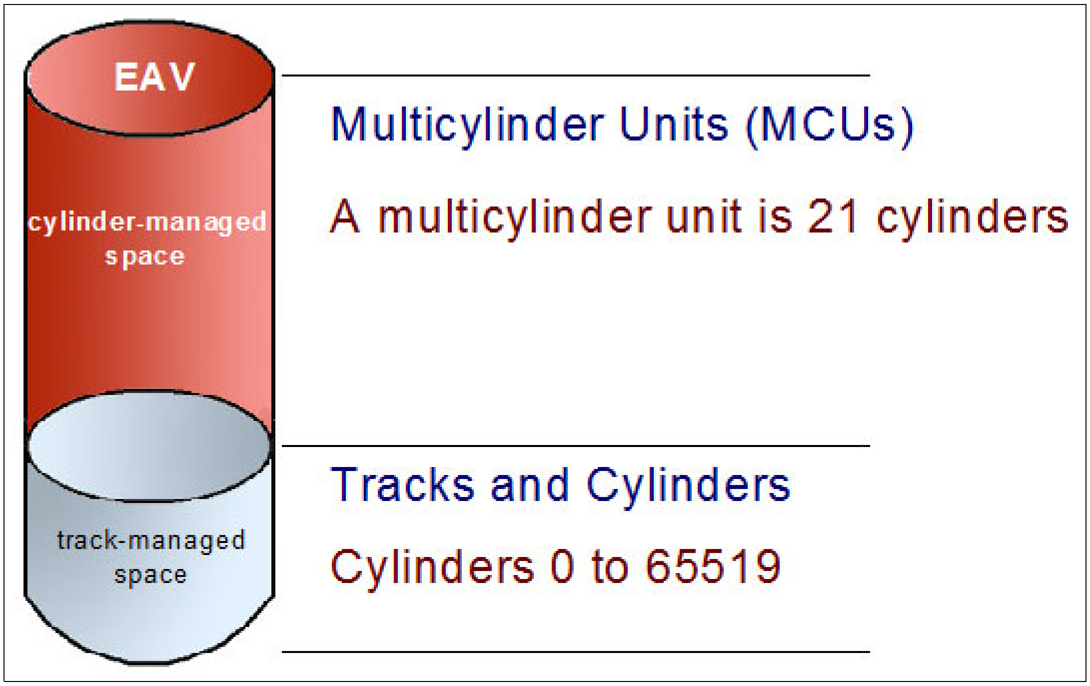

Figure 6-11 The EAS is cylinder-managed space

EAS-eligible data sets

In EAV R1 (z/OS V1.10), only VSAM data sets were eligible for EAS. At EAV R2 (z/OSV.11), extended format sequential data sets were added. Now at EAV R3 (z/OS V1.12+), almost all data set types are eligible and DFSMShsm is enhanced to support almost all data set types.

DFSMShsm EAV support includes almost all data set types:

- Sequential, including extended, basic, and large formats
- Partitioned data set (PDS) and partitioned data set extended (PDSE)
- Basic direct access method (BDAM)
- Undefine data set organization (DSORG=U)
- Catalogs (basic catalog structure (BCS) and VSAM volume data set (VVDS)
- VSAM
- z/OS file system (zFS), which is VSAM

The DFSMShsm logic uses the EATTR attribute (as used in IDCAMS DEFINE and ALLOCATE, JCL, dynamic allocation, and DATACLAS) when it performs volume selection.

The following data set types are not eligible for EAS:

- Page data sets
- Volume table of contents (VTOC) and VTOC indexes
- Hierarchical file system (HFS)
- VSAM with **IMBED** or **KEYRANGE** parameters that were inherited from earlier physical migrations or copies

These data sets must remain allocated in track-managed space. HSM is not used to manage page data sets, VTOCs, or VTOC indexes, in any case.

EATTR parameter

To allow users to control the migration of non-VSAM data sets to EAS (during the earlier releases of EAV), a new data set level attribute was defined. Each data set used a new attribute, **EATTR**, to indicate whether the data set can support extended attributes (that is, format-8 and format-9 data set control blocks (DSCBs)). By definition, a data set with extended attributes can reside in EAS on an EAV. This attribute can be specified for non-VSAM data sets and VSAM data sets.

The following values are valid for this attribute:

EATTR = NO
: No extended attributes. Allocation can select a volume with or without extended addressing with no preference based on this attribute. However, when the data set created, the data set has an F1 DSCB and cannot have extents in the EAS. `NO` is the *default for non-VSAM* data sets.

EATTR = OPT
: Extended attributes are optional. The data set can use extended attributes and reside in EAS. This value is the *default for VSAM* data sets.

EATTR handling

EXIT28 is used for restore processing to pass DFSMSdss the attributes of pre-allocated data sets (in particular the EATTR) that are deleted by DFSMShsm before its reallocation.

DFSMShsm uses an IDCAMS, which was modified to recognize the EATTR value on an import of a catalog data set and also passes the EATTR value to an export.

Support for DFSMShsm-owned data sets

HSM data sets can now also be allocated as EAS-eligible in V1.12. By using the **`SETSYS USECYLINDERMANAGEDSPACE`** command, migration and backup copies of data sets can be EAS-eligible. Dynamic allocation of PDOx/y, LOGx/y, and JRNL as EAS-eligible is supported, too.

DFSMShsm journal data set

Large and extended format sequential, PDSE, and VSAM data sets can occupy an entire EAV with multiple extents. Since EAV R2 (z/OS V1.11), direct-access device space management (DADSM) allows a single extent to span the track-managed and cylinder-managed space. Therefore, a data set can occupy the entire volume with a single extent. Therefore, the DFSMShsm journal can potentially occupy an entire volume.

Command for EAS for migration and backup

With DFSMShsm V1.12 migration data sets (migration level 1 (ML1) and migration level 2 (ML2)) and backup, data sets can be directed to cylinder-managed space on EAVs. All members of an HSMplex need to be at V1.12 before you can use this function. You can use the **`SETSYS USECMS`** command to allow EAS eligibility for migration and backup data sets (Example 6-1).

Example 6-1 MVS SETSYS USECMS command

```
SETSYS USECYLINDERMANAGEDSPACE(Y|N)
SETSYS USECMS(Y|N)
```

The default is to not use cylinder-managed storage. The status of the **`USECYCLINDERMANAGEDSPACE`** parameter can be displayed by using a **`QUERY SETSYS`** command. The output is displayed in the ARC0153I message.

DFSMShsm data mover

Support is added to provide the following functions:

- Migration and backup, and recall and recovery of a PDS
- Backup and recovery of uncataloged and user-labeled sequential data sets
- Backup and recovery of catalogs
- Migration and recall of user-labeled sequential data sets
- Recovery with a specified pre-allocated data set

ABARS support

Enhancements were made to ABARS to support the additional data set types for z/OS DFSMS V1.12. See “EAS-eligible data sets” on page 107. The enhancements are listed:

- If the extended attributes for the EAS-eligible Instruction Data Set are lost during recovery of this data set to non-EAV, message ARC0784I is issued.
- The algorithm of aggregate recovery of migrated data sets from the INCLUDE list is changed. For storage requirements for an aggregate group, the calculation of the size of the user catalog and user-labeled data set was corrected, and the message ARC6369I is now issued with the correct space requirement value.
- A new algorithm is available to assign the JOBNAME and STEPNAME of the source data set (or a pre-allocated data set) from the ALLOCATE list to the target data set in the same way that DFSMSdss is created.
- The EATTR value of a source user catalog from the ALLOCATE list is passed to the target.

Modified messages

The following messages were changed or corrected, or the situations in which they are issued were changed.

ARC0153I

ARC0153I now shows the USECMS status:

```
ARC0153I SCRATCHFREQ=days, SYSOUT(CLASS=class, COPIES=number, SPECIAL
         FORMS={form | NONE}), SWAP={YES | NO},
         PERMISSION={YES | NO}, EXITS={NONE | exits}, UNLOAD={YES | NO},
         DATASETSERIALIZATION= {USER | DFHSM}, USECMS={YES | NO}
```

ARC0784I

This message is now issued when a recall, recover, or ARECOVER to a non-EAV-capable volume causes the loss of format-9 DSCB data:

```
ARC0784I EXTENDED ATTRIBUTES FOR DATA SET dsname WERE NOT RETAINED DURING
         THE RECALL | RECOVER | ARECOVER
```

ARC6369I

This message is displayed for ABARS support when space is recalculated and corrected for EAS allocation for user catalogs and user-labeled data sets. The format though remains the same:

```
ARC6369I STORAGE REQUIREMENTS FOR AGGREGATE GROUP agname, ARE: L0=number
         {K|M|G|T}, ML1=number {K|M|G|T}, ML2=number{K|M|G|T}, TOTAL=number
         {K|M|G|T}
```

SMS volume selection

For SMS volume selection on recall, recovery, or ARECOVER, DFSMShsm will pass the DATACLASS value to DFSMSdss and volume selection will be performed by SMS. The only exception is for PDS. The DFSMShsm volume selection logic for SMS PDS data sets will be consistent with the non-SMS volume selection logic.

Non-SMS volume selection

DFSMShsm checks the data set level attribute, EATTR, when it selects a non-SMS volume. The EATTR data set level attribute specifies whether a data set can use extended attributes (format-8 and format-9 DSCBs) and optionally reside in EAS on an EAV. The EATTR attribute is optional. If you specify it, valid values are `NO` and `OPT`. For more information about the EATTR attribute, see ALLOCATE and DEFINE CLUSTER in *z/OS DFSMS Access Method Services for Catalogs,* SC26-7394.

If an EATTR value is not specified for a data set, the following default data set EATTR values are used:

- The default behavior for VSAM data sets is `OPT`.
- The default behavior for non-VSAM data sets is `NO`.

Preference for volume selection corresponds to the EATTR values for each data set:

- When EATTR = NO, the data set has no preference to any volume. For recalls, recovers, and ARECOVERs, the volume with the most free space is selected, but only the track-managed free space is used to determine the free space calculations of an eligible EAV. The data set is allocated with a format-1 DSCB, which indicates that it is not eligible to have extents in the EAS.
- If an EAV or non-EAV is chosen for allocation, and the value is EATTR=`NO` and the data set used a format-8 or format-9 DSCB with vendor attributes, the data set is allocated with a format-1 DSCB. The system issues message ARC0784I to indicate that extended attributes were not retained for this recall, recovery, or ARECOVER. In this situation, the format-9 information of the data set will be lost on the recall, recovery, or ARECOVER.
- When EATTR = `OPT`, the data set prefers EAV volumes if the size of the allocation is equal to or greater than the breakpoint value (BPV) that is specified. If the requested size of the data set allocation is smaller than the BPV, EAVs and non-EAVs have equal preference:
  - If an EAV is chosen for allocation and the data set type is EAS-eligible, the data set is allocated with format-8 or format-9 DSCBs in either the track-managed space or the EAS.
  - If a non-EAV is chosen for the allocation and the data set used format-8 or format-9 DSCBs with vendor attributes, the data set is allocated with a format 1. Then, DFSMShsm issues the ARC0784I message, which signifies that extended attributes were not retained for this recall, recovery, or ARECOVER. The EATTR value always remains in the format-1 DSCB.

EAV considerations for space management

Because an EAV has two areas, track-managed space and cylinder-managed space (EAS), new selection criteria were added with the **`TRACKMANAGEDTHRESHOLD`** parameter. It is used with the **`THRESHOLD`** parameter. It was necessary because data set size is also considered when data sets are migrated. Also, data sets in the EAS are larger on average than the data sets that are below in track-managed space. Data sets in the EAS will be migrated before the data sets that are in track-managed space, which might mean that the track-managed space is not freed up.

DFSMShsm examines both volume-level thresholds and track-managed thresholds for L0 EAVs. If either threshold is exceeded, migration eligibility is performed for data on the volume. If track-managed threshold values are not specified, the default is to use the volume-level threshold values. Data set eligibility is further qualified by the location of the first three extents of the data set. If any of the first three extents are allocated in track-managed space, the data set is eligible to be processed when the track-managed space or entire volume space is managed. If none of the first three extents are allocated in track-managed space, the data set is eligible to be processed when the entire volume space is managed. Table 6-2 shows how selections are made.

Table 6-2 Migration eligibility

Track-managed threshold exceeded	Volume threshold exceeded	Data set selection
Yes	No	Only data sets with one or more of the first three extents that are allocated in track-managed space
No	Yes	All data sets
Yes	Yes	All data sets
No	No	None selected

For SMS volumes, DFSMShsm uses the track-managed threshold and the volume threshold from the storage group. For non-SMS volumes, the track-managed threshold is specified on the **ADDVOL** command when you add primary volumes to DFSMShsm:

```
ADDVOL PROD01 PRIMARY UNIT(3390) TRACKMANAGEDTHRESHOLD(80 40) THRESHOLD(85 60)
```

If no TRACKMANAGEDTHRESHOLD is specified, the volume threshold values are used, by default.

Support and considerations for 1 TB EAVs

The support for DFSMShsm 1 TB resulted in no external updates. Only internal fields required updating to handle the larger quantities that are dealt with. Although no externals require definition or implementation for DFSMShsm, several considerations are new.

Testing shows a considerable increase in elapsed time for IBM HyperSwap® for larger volume capacities due to the larger amounts of metadata. Spreading 1 TB EAVs over as many LSSs as possible is best.

Additionally, although dynamic volume expansion automatically starts reformatting the VTOC so the larger extra capacity that is added is recognized by the VTOC, you still need to manually expand the size of the VTOC, VTOC index, and VVDS to handle the larger number of expected data set entries. Instructions for dynamic volume expansion and expanding the VTOC and VTOC index are covered in Chapter 10 of *z/OS V1.13 Technical Update,* SG24-7961.

Autodump considerations with EAVs

With the introduction of large capacity volumes, such as EAV, users who use dump stacking need to review DUMP CLASS and its dump stack figure.

A storage group of different capacity volumes (3390 Model 3, Model 9, Model 27, Model 54, and EAV) might require the DUMP window to be extended for DFSMShsm to complete the full volumes dump.

A dump task with all high-capacity volumes to be dumped (such as Model 27, Model 54, or EAV) requires more time to complete than another dump task, which only dumps all Model 3 volumes by using the same dump stacking figure.

We recommend that you group the same capacity volumes into the same dump class and assign the appropriate dump stacking figure (the higher the volume capacity, the smaller the dump stacking figure) to prevent large differences between the dump tasks' elapsed times to complete. This configuration helps minimize the need to extend the dump window.

EAV considerations for recovery functions

You can specify the EATTR value when you allocate a data set to define whether the data set can use extended attributes (format-8 and format-9 DSCBs) and optionally reside in EAS. If a data set is directed to a pre-allocated data set on a RECOVER or ARECOVER function, the system uses the EATTR value of the pre-allocated data rather than the EATTR value that is specified for the data set source recovery data.

New considerations exist for ARECOVER and EATTR. The following methods can be used to allow the EATTR information from the ABACKUP site to be restored, with the data set:

- Use the **`DSCONFLICT(RENAMETARGET)`** parameter on the **`ARECOVER`** command, then manually delete the existing data set. Because the ARECOVER resolved the conflict by renaming the existing data set before it restores the ABARS copy, both copies will have their own EATTR. You need to remove only the renamed existing data set.
- Use the **`DSCONFLICT(BYPASS)`** parameter on the **`ARECOVER`** command, then manually delete the existing data set and existing conflict resolution data set, and then rerun the **`ARECOVER`** command. Only the data sets that were bypassed or failed for another reason on the first ARECOVER will be processed. Because no conflict exists, you will get the ABARS copy, complete with its EATTR.

For more information, see the **`EATTR`** parameter of the **`ALLOCATE`** command in *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394.

Recommendations for using EAVs

Because ML1, ML2, and backup volume allocations can reside in either the track-managed space or the cylinder-managed space on an EAV beginning in V1R12, it is important to understand that allocations in the cylinder-managed space are in multicylinder units (MCUs) and can potentially result in an over-allocation.

You might want to evaluate the SMS and non-SMS BPV settings. If possible, enable the system to avoid over-allocation for smaller disk space requests. For larger disk spaces, use cylinder-managed space and allocate in MCUs for larger disk space requests.

For example, setting the BPV to a high value will result in smaller disk space requests made by DFSMShsm more likely to reside in the track-managed area. Larger disk space requests will prefer to use space that can end up in an over-allocation. However, this over-allocation will be a much smaller percentage of the total disk space request and the over-allocation might be acceptable. For more information about using EAVs, see *z/OS DFSMS Using the New Functions*, SC26-7473.

6.4.2 z/OS DFSMShsm earlier level coexistence

Because of the differences between the EAV releases, particularly the more limited function EAS-eligible data set types, as shown in Table 6-3 on page 113, coexistence issues exist if members in an HSMplex are at earlier levels.

Table 6-3 EAV data set type support

EAV release	z/OS release	Data set types supported
EAV R1	z/OS V1.10	VSAM
EAV R2	z/OS V1.11	VSAM and Extended Format Sequential Access Method (EFSAM)
EAV R3	z/OS V1.12 and later	VSAM, catalogs, sequential access method (SAM), basic direct access method (BDAM), PDS, PDSE, undefined, and zFS

Members with earlier-level releases of DFSMShsm will be restricted in their function.

One TB EAVs

Space calculations for data sets that are migrated, backed up, or backed up with ABARs from z/OS V1.10 to V1.11 will show incorrect values when recalled or recovered if they are greater than 2 TB. See OW30632 for details. Attempts to recall or recover on z/OS V1.10 or V1.11 will be failed.

Only z/OS V1.13 and z/OS V1.12 with OW28553 can bring 1 TB EAVs online.

Toleration

z/OS V1.10 and earlier DFSMShsm releases needed coexistence to handle the additional EAS-eligible data sets and the new EATTR data set attribute.

No toleration maintenance is available for z/OS V1.11, except for OA30632, because the basic toleration logic is carried forward from the two earlier APARS OA27146 and OA28804. The same messages are issued in z/OS V1.11 as in z/OS V1.12 or z/OS V1.13, if for example, a restore to a non-EAV volume results in the loss of DSCB9 data, it results in the issue of an ARC0784I message. This basic toleration logic is from OA28804.

Although OA27146 and OA28804 are old, they are important in the context of coexistence and toleration because they provided the logic in use today for coexistence.

APAR OA27146 provided DFSMShsm toleration for EAV R2 and applied to z/OS DFSMShsm in an HSMplex environment with V1.11 DFSMShsm and V1.10 and V1.9 DFSMShsm levels (z/OS V1.9 is out of support today):

- DFSMShsm toleration for EAV R2. z/OS V1.10 and V1.9 releases of DFSMShsm needed coexistence to handle the additional EAS-eligible data sets and the new EATTR data set attribute.
- Coexistence support for DFSMShsm in z/OS V1.10 to enable DFSMShsm to tolerate the DFSMShsm support that is provided in z/OS V1.11. In z/OS V1.10, DFSMShsm can RECALL, RECOVER, and ARECOVER both VSAM and non-VSAM data sets from a z/OS V1.11 migration or backup copy with format-8 and format-9 DSCBs to non-EAVs. Format-8 and format-9 DSCBs will be converted to format-1 DSCBs, and format-9 DSCB attributes will be lost if such a data set is recalled, recovered, or recovered by using ARECOVER to non-EAV. In this case, an ARC0784I message is issued if DFSMSdss issues an ADR556W message. For z/OS V1.9, support is added for preserving the EATTR value.

- Non-VSAM data sets with format-8 and format-9 DSCBs will be visible to the system, but not eligible for migration, backup, or ABACKUP processing. When one of these data sets is encountered, it is failed by DFSMShsm. Messages ARC1220I for migration and ARC1320I messages for backup with RSN21 or ARC0734I with RC20 and RSN21 are issued.

  We get the following error message when we try to migrate the EAV data set CVERNON.EAVTEST.EFSAM0 on a V1.10 level system:

  ```
  ARC1001I CVERNON.EAVTEST.EFSAMO MIGRATE FAILED, RC=0020, REAS=0021
  ARC1220I DATA SET NOT ELIGIBLE FOR MIGRATION
  ***
  ```

  We get the following error message when we try to back up the EAV data set CVERNON.EAVTEST.EFSAM0 on a V1.10 level system:

  ```
  ARC1001I CVERNON.EAVTEST.EFSAMO  BACKDS FAILED, RC=0020, REAS=0021
  ARC1320I DATA SET NOT ELIGIBLE FOR BACKUP
  COMMAND REQUEST 00000072 SENT TO DFSMSHSM
  ***
  ```

- Recall/recover of non-VSAM data sets that have format-8 and format-9 DSCBs will be completed to the track-managed space so that DFSMSdss will RESTORE them. ADR556W is issued, and DFSMShsm issues the ARC0784I message. However, the RECALL will fail if no track-managed space volume is available to satisfy the space allocation request.

  Data set CVERNON.EAVTEST.EFSAM0 originally resided on EAV volume GKDD65 and it is the only volume in storage group EAVGK. When we try to recall the data set on the V1.10-level system, DFSMShsm tries to allocate a data set on a track-managed space volume but none of them are available to storage group EAVGK. Therefore, DFSMShsm failed the recall, as shown:

  ```
  ARC1001I CVERNON.EAVTEST.EFSAMO  RECALL FAILED, RC=0069, REAS=0709
   ARC1169I RECALL/RECOVER FAILED DUE TO AN ERROR IN DFDSS
    RECALL REQUEST 00000073 SENT TO DFSMSHSM
   ***
  ```

- Recover REPLACE of non-VSAM data sets is failed if a pre-allocated non-VSAM data set is identified with a format-8 or format-9 DSCB. The ARC1158I message is issued. Messages ARC1001I or ARC0734I provide the data set name and the reason code 51. If a pre-allocated non-VSAM data set is identified with a format-1 DSCB, the EATTR value is passed to DFSMSdss to preserve the value.

  We get the error message ARC1158I when we try to recover data set CVERNON.EAVTEST.EFSAM0 to replace the existing data set, which has format-8 and format-9 DSCBs on a V1.10-level system:

  ```
  ARC1001I CVERNON.EAVTEST.EFSAMO RECOVER FAILED, RC=0058, REAS=0051
  ARC1158I RECOVER DATA SET FAILED
  COMMAND REQUEST 00000074 SENT TO DFSMSHSM
  ***
  ```

APAR OA22804 provided DFSMShsm V1.9 to V1.6 with toleration for EAV R1. The pre-V1.10 versions of z/OS DFSMShsm coexisted in a HSMplex with z/OS DFSMShsm V1.10. In this situation, the EAV is inaccessible for the pre-V1.10 systems because they will be offline. Data sets that were migrated or backed up from an EAV to ML1, ML2, and backup volumes (non-EAV) under z/OS DFSMShsm V1.10 are accessible for the pre-V1.10 DFSMShsm systems.

This toleration APAR is required to support recall, recover, and recycle for these data sets in the pre-V1.10 systems. If the toleration APAR is not installed, the recall, recover, and recycle of these data sets fail during pre-V1.10 DFSMShsm processing. AUDIT and RECYCLE are changed to process a data set common data set descriptor records (CDDs) with a format-8 DSCB.

APAR OA22804 provides coexistence support of the pre-V1.10 DFSMShsm release to perform the following functions:

- Allow pre-V1.10 systems to recognize MCV records for EAVs.
- Restrict the selection of extent reduction request from the common recall queue (CRQ), if the request is directed to an EAV.
- Restore EAS-eligible data sets from a backup or migrated copy on an online volume.
- Enable pre-V1.10 systems to process the new VCC keywords that might be specified in the management class backup copy technique field.
- Support recall/recover on V1.9 or lower of an EAS data set with a format-8 DSCB that was migrated/backed up on V1.10:
  - Converts to format-1 DSCBs
  - Issues the new ARC0784I message and update the functional statistics record (FSR) record if format-9 DSCB vendor attributes are lost during the RECALL/RECOVER/ARECOVER function:

    ```
    ARC0784I EXTENDED ATTRIBUTES FOR DATA SET dsname WERE NOT RETAINED DURING
    THE RECALL | RECOVER | ARECOVER
    ```

    Explanation: The data set was recalled, recovered, or recovered by using ARECOVER from a migration copy, backup copy, or an aggregate backup successfully. However, JOBNAME, STEPNAME, creation time attributes, and vendor attributes from the format-9 DSCB of the recalled or recovered data set were not retained because the volume on which it was placed did not support format-8 and format-9 DSCBs. The recall, recovery, or aggregate recovery continues.

    Restriction: If a recall for extent reduction is directed to an EAV, the selection of this request from the common recall queue (CRQ) will be restricted in the pre-V1.10 systems.

Use of EAV volumes in mixed environments where the volume cannot be brought online to all systems can also result in recalling or recovering data sets that belong to a storage class that has a guaranteed space attribute.

Recovery considerations

If you need to recover a data set backup that is made from an SMS EAV volume to a lower-level system, perform the following tasks:

- If the EAV is still available on the higher-level system, delete or rename the data set on the higher-level system.
- If the EAV is no longer available and a catalog entry exists, use IDCAMS DELETE NOSCRATCH to remove the data set from the catalog.

Note: The backup copy must reside on a non-EAV and the volume must be online to the lower-level system.

7

Tape considerations

This chapter describes tape usage and the necessary setup for DFSMShsm to process tape for backup and migration. The focus in this chapter is on automatic tape management by using IBM TotalStorage Virtual Tape Server (Virtual Tape Server) and automatic tape libraries. Manually operated tapes are covered at an overview level because of limited usage. The focus is on IBM hardware and related functions. However, it does not include information about hardware installation. The information that is provided is for customization from a DFSMShsm perspective.

7.1 Basic tape setup that is needed for DFSMShsm

To define a tape environment to DFSMShsm, you have to code **SETSYS** commands and also support these allocations in the automatic class selection (ACS) routines. Storage management subsystem (SMS)-managed environments are the common practice today.

These settings vary depending on your environment in relationship to the following variables:

- Media types that are used
- Tape environment (virtual, automated tape library (ATL), or manually operated)
- Use of a tape management system (IBM or not IBM)
- Security

The ACS routine code ensures that your allocation points you to the correct drive, device type, and media. For the non-SMS managed data, you use the esoteric to obtain the correct device.

This chapter describes the tape environment that is needed to support DFSMShsm in more detail. It also describes how to establish the DFSMShsm functions that relate to tape.

7.1.1 DFSMShsm single file tape operation

DFSMShsm tape processing functions support single file tapes. You cannot create either multiple file reel-type backup or migration tapes. Support of reel-type tapes is limited to the following functions:

- Recall or recovery of data sets that currently reside on reel-type tapes
- Creation of dump tapes
- Creation of control data set (CDS) Version Backup tapes
- Aggregate backup and recovery support (ABARS) functions

Single file format is a potentially multireel tape with only one file on it. Using the single file format reduces I/O and system serialization because only one label is required for each connected set (as opposed to multiple file format tapes that require a label for each data set). The standard-label tape data set that is associated with the connected set can span up to the allocation limit of 255 tapes. This standard-label tape data set is called the *DFSMShsm tape data set*. Each user data set is written, in 16 K logical blocks, to the DFSMShsm tape data set. A single user data set can span up to 254 tapes, previously a single user data set only spanned to 40 tapes.

Tape hardware emulation

Emulation of one tape subsystem by another tape subsystem is commonplace. Virtual tape and new technologies emulate earlier technologies. Because the cartridges are not interchangeable, DFSMShsm added support for several subcategories that use 3490 cartridges:

- 3590 Model E 256 track recording
- 3590 Model B 128 track recording
- 3590 Model H 384 track recording
- Virtual Tape Server
- All others whose device type is 3490

Additionally, DFSMShsm supports the following subcategories of 3590 cartridges:

- 3590 Model E 256 track recording
- 3590 Model B 128 track recording

- 3590 Model H 384 track recording
- 3592 Model J1A 512 track recording (EFMT1)
- 3592 Model E05 drive format 896 track recording (EFMT2)
- 3592 Model E06 drive format 1152 track recording (EFMT3)
- 3592 Model E07 drive format 1664/2560 track recording (EFMT4)
- 3592 Model E08 drive format 5120 track recording (EFMT5)
- All others whose device type is 3590

Restriction: You can use the 3592-E05 and newer tape drives in 3590 emulation mode only, never 3490. The 3592 Model J1A can operate in 3490 emulation mode only when it uses MEDIA5 for output.

This additional support enables the multiple use of these technologies concurrently for the same function within the same HSMplex on condition that their output use is on different DFSMShsm hosts. This support improves the installation of new technologies to improve performance and capacity. When multiple technologies are used concurrently for output outside the Data Facility Storage Management Subsystem (DFSMS) tape umbrella, an esoteric unit name is needed to enable DFSMShsm to distinguish them. An esoteric name is also needed to select existing (partially filled) tapes that are compatible with the allocated drives.

Drive compatibility

Tape cartridges and tape formats that are compatible with the 3592 tape drives are described. The 3592 tape cartridge has external dimensions (form factor) that allow it to be used within existing storage cells of libraries that contain 3590 tapes. However, the 3592 tape drives must be installed in frames separately from any 3590 drives. The 3592 tape drive cartridges are not compatible with 3590 tape drives cartridges, and, likewise, 3590 tapes cannot be used in the 3592 drives. You need to consider the following compatibility topics:

- Supported formats

  The following formats with different recording densities yield different capacities per cartridge type:

  - Enterprise Format 1 (EFMT1) is used by the 3592 J1A Tape Drive and the 3592 E05 Tape Drive in both native and J1A emulation mode. This format records 512 tracks on eight channels. The 3592 E06 tape drive and 3592 E07 tape drive with code level D3I3_5CD or higher read data that is written in EFMT1 format but do not write in EFMT1.
  - Enterprise Format 2 (EFMT2) is used by the 3592 E05 Tape Drive and the 3592 E06 Tape Drive. This format records 896 tracks on 16 channels. When the drive operates on encrypted data at this density, the recording format is Enterprise Encrypted Format 2 (EEFMT2).
  - Enterprise Format 3 (EFMT3) is used by the 3592 E06 Tape Drive and the 3592 E07 Tape Drive. This format records 1,152 tracks on 16 channels. When the drive operates on encrypted data at this density, the recording format is Enterprise Encrypted Format 3 (EEFMT3).
  - Enterprise Format 4 (EFMT4) is used by the 3592 E07 Tape Drive. This format records 1,664 tracks on JB and JX cartridges and 2,560 tracks on JC, JK, and JY cartridges on 32 channels. When the drive operates on encrypted data at this density, the recording format is Enterprise Encrypted Format 4 (EEFMT4).
  - Enterprise Format 5 (EFMT5) is used by the 3592 E08 Tape Drive. This format records 5,120 tracks on JC, JD, JK, JL, JY, and JZ cartridges on 32 channels. When the drive operates on encrypted data at this density, the recording format is Enterprise Encrypted Format 5 (EEFMT5).

- The following cartridge types are supported:
  - The 3592 E08 Tape Drive uses the Advanced Type D read/write (type JD), the Advanced Type D WORM (type JZ), and the Advanced Type D economy (type JL) cartridges. Both the type JD and JZ cartridges have a maximum capacity of 10 TB (9,313.23 GiB). The Advanced Type D economy cartridges have a maximum capacity of 2 TB (1,862.65 GiB).
  - The 3592 E07 and E08 tape drive uses the Advanced Data (type JC) and Advanced WORM (type JY) cartridges. The type JC and JY cartridges have a maximum capacity of 4,000 GB (3,725.3 GiB) for E07 and 7 TB (6,519.26 GiB) for E08.
  - The 3592 tape drive models E07, E06, and E05 use 3592 Extended Data (type JB) and Extended WORM (type JX) with a maximum capacity of 1,600 GB (1,490.12 GiB) with EFMT4 format, 1,000 GB (931.3 GiB) with EFMT3 format, and 700 GB (651.93 GiB) with EFMT2 format. The 3592 E07 Tape Drive with code level D3I3_5CD or higher can read format EFMT2 but does not write EFMT2 format.
  - The 3592 tape drive models E06, E05, and J1A use 3592 Standard Data (type JA) and Standard WORM (type JW) cartridges with a maximum capacity of 640 GB (596.04 GiB) with EFMT3 format, 500 GB (465.66 GiB) with EFMT2 format, and 300 GB (279.39 GiB) with EFMT1 format. The 3592 E07 Tape Drive with code level D3I3_5CD or higher can read only cartridge types JA and JW.
  - The 3592 tape drive models E06, E05, and J1A use 3592 Economy Data (type JJ) and Economy WORM (type JR) cartridges with a maximum capacity of 128 GB (119.21 GiB) with EFMT3 format, 100 GB (93.13 GiB) with EFMT2 format, and 60 GB (55.88 GiB) with EFMT1 format. The 3592 E07 Tape Drive with code level D3I3_5CD or higher can read only cartridge types JJ and JR.
  - The 3592 E07 and E08 tape drives use the 3592 Advanced Economy (type JK) cartridge. The maximum capacity is 500 GB (465.66 GiB) with EFMT4 format, and 900 GB (838.19 GiB) with EFMT5 format.

Table 7-1 shows the cartridge products, types, storage management subsystem (SMS) categories, and characteristics.

Table 7-1 Types of IBM 3592 tape cartridges

<table>
<tr><th rowspan="2">Product, type, and SMS category</th><th colspan="5">Native capacity</th><th rowspan="2">Part number</th></tr>
<tr><th>E08</th><th>E07</th><th>E06/EU6</th><th>E05</th><th>JA1</th></tr>
<tr><td rowspan="3">Data, JA, MEDIA5, (610 m)</td><td rowspan="3">Not supported</td><td>640 GB (596.04 GiB) E06 format</td><td>640 GB (596.04 GiB) E06 format</td><td rowspan="2">500 GB (465.66 GiB) E05 format</td><td rowspan="3">300 GB (279.39 GiB) (J1A format)</td><td rowspan="3">18P7534</td></tr>
<tr><td>500 GB (465.66 GiB) E05 format</td><td>500 GB (465.66 GiB) E05 format</td></tr>
<tr><td>300 GB (279.39 GiB) J1A format[1]</td><td>300 GB (279.39 GiB) J1A format</td><td>300 GB (279.39 GiB) J1A format</td></tr>
<tr><td rowspan="2">Extended Data, JB, MEDIA9, (825 m)</td><td rowspan="2">Not supported</td><td>1,600 GB (1490.12 GiB) E07 format</td><td rowspan="2">1,000 GB (931.32 GiB)</td><td rowspan="2">700 GB (651.93 GiB)</td><td rowspan="2">Not supported</td><td rowspan="2">23R9830</td></tr>
<tr><td>1,000 GB (931.32 GiB) E06 format</td></tr>
</table>

Product, type, and SMS category	Native capacity					Part number
	E08	E07	E06/EU6	E05	JA1	
Advanced Data, JC, MEDIA11, (825 m)	7 TB (6.37 TiB) E08 format	4,000 GB (3,725.29 GiB) E07 format	Not supported	Not supported	Not supported	46X7452
Advanced Type D read/write, JD, MEDIA13 (1,032 m)	10 TB (9.1 TiB) E08 format	N/A	N/A	N/A	N/A	2727263
Economy, JJ, MEDIA7, (246 m)	Not supported	128 GB (119.21 GiB) E06 format[1]	128 GB (119.21 GiB) E06 format	100 GB (93.13 GiB) E05 format	60 GB (58.88 GiB) J1A format	24R0317
		100 GB (93.13 GiB) E05 format	100 GB (93.13 GiB) E05 format			
		60 GB (58.88 GiB) J1A format[1]	60 GB (58.88 GiB) J1A format	60 GB (58.88 GiB) J1A format		
Advanced Economy, JK, MEDIA13, (146 m)	900 GB (838.19 GiB) E08 format	500 GB (465.66 GiB) E07 format	Not supported	Not supported	Not supported	46X7453
Advanced Type D Economy, JL, MEDIAnn (241 m)	2 TB (1.82 TiB) E08	Not supported	Not supported	Not supported	Not supported	2727264
Economy WORM, JR, MEDIA8, (246 m)	Not supported	128 GB (119.21 GiB) E06 format[1]	128 GB (119.21 GiB) E06 format	100 GB (93.13 GiB) E05 format	60 GB (58.88 GiB) J1A format	24R0317
		100 GB (93.13 GiB) E05 format	100 GB (93.13 GiB) E05 format			
		60 GB (58.88 GiB) J1A format	60 GB (58.88 GiB) J1A format	60 GB (58.88 GiB) J1A format		
WORM, JW, MEDIA6, (610 m)	Not supported	640 GB (596.04 GiB) E06 format	640 GB (596.04 GiB) E06 format	500 GB (465.66 GiB) E05 format	300 GB (279.39 GiB) J1A format	18P7538
		500 GB (465.66 GiB) E05 format	500 GB (465.66 GiB) E05 format			
		300 GB (279.39 GiB) J1A format[1]	300 GB (279.39 GiB) J1A format	300 GB (279.39 GiB) J1A format		

Product, type, and SMS category	Native capacity					Part number
	E08	E07	E06/EU6	E05	JA1	
Extended WORM, JX, MEDIA10, (825 m)	Not supported	1,600 GB (1,490.12 GiB) E07 format 1,000 GB (931.32 GiB) E06 format	1,000 GB (931.32 GiB) E06 format	700 GB (651.93 GiB)	Not supported	23R9831
Advanced WORM, JY, MEDIA12, (880 m)	7 TB (6.37 TiB) E08 format	4,000 GB (3,725.29 GiB) E07 format	Not supported	Not supported	Not supported	46X7454
Advanced WORM, JZ, MEDIA12 (1,032 m)	7 TB (6.37 TiB) E08 format	Not supported	Not supported	Not supported	Not supported	2727265
Cleaning, CLNxxxJA3	N/A	N/A	N/A	N/A	N/A	18P7535

Notes:
[1] The 3592-E07 supports reading JA, JJ, JR, and JW cartridges only with code level D3I3_5CD or later.
[2] The 3592-E08 supports reading JD, JL, and JZ cartridges only with code level D3I4_460 or later.

7.2 Tape device allocations

Tape allocations that are invoked by DFSMShsm can be handled by dynamic allocation in the first instance. But the WAIT and NOWAIT options of the following DFSMShsm device allocation parameters tell DFSMShsm whether DHSM tape allocation is suspended or not. The following parameters determine suspension:

- INPUTTAPEALLOCATION
- OUTPUTTAPEALLOCATION
- RECYCLETAPEALLOCATION

The behavior depends on the environment, such as the following factors:

- Whether the environment is a JES2 or a JES3 environment
- The number of available tape devices
- The general availability of the tape devices

7.2.1 Tape device selection

If one type of tape device exists at your site, no consideration for tape device selection exists. For any tape-input or tape-output operation, only one type of tape device can be selected.

However, you have more than one type of tape device at your site, a choice must be made about which device to select for a certain function.

You can direct MVS to select a device you specify, or you can allow MVS to choose the device for you. You can restrict MVS allocation to selecting the specific type of tape device you want for a certain function in either an SMS-managed tape library or in a non-library environment. Restricting device selection can improve tape processing when high-performance devices are selected and can ensure that new devices are selected when they are introduced into your tape environment.

7.2.2 Non-library tape device selection

When DFSMShsm requests that to allocate and mount a tape device, MVS allocates a tape device according to two aspects of the request.

If the allocation request is for a specific tape or for a scratch tape, MVS first honors the device type that DFSMShsm selects. If DFSMShsm specifies a generic unit name for a cartridge-type device, MVS allocates a cartridge-type device, but selects a cartridge-type device with or without a cartridge loader according to its own criteria. (If all of the devices contain cartridge loaders, the MVS criteria are unimportant.) If DFSMShsm specifies an esoteric unit name that is associated with devices with a cartridge loader, MVS allocates a device from that esoteric group.

If the DFSMShsm tape-allocation request is for a cartridge-type device, MVS selects a device by considering whether the request is for a specific tape or for a scratch tape. If the request is for a specific tape, MVS tries to allocate a device without a cartridge loader. If the request is for a scratch tape, MVS tries to allocate a device with a cartridge loader. If the preferred type of device is unavailable, MVS selects any available cartridge-type device.

7.2.3 Library tape device selection

Tape devices that are associated with SMS-managed tape libraries are selected based on the technologies they support. You can restrict selection of these devices by specifying the data-class attributes that are associated with a device.

7.2.4 Specifying the WAIT option

When DFSMShsm initiates a tape allocation request, an exclusive enqueue is issued against the SYSZTIOT resource. This enqueue remains until the tape allocation request is honored or the allocation might fail. A setting of the WAIT option causes all DFSMShsm activity to go on hold.

> **Note:** Use the WAIT option only in JES3 environments or environments with a dedicated DFSMShsm esoteric.

7.2.5 Specifying the NOWAIT option

When the NOWAIT option is used, dynamic allocation is not waiting for a device to become available. Either a device is returned to DFSMShsm or a failure occurs because a device was not available. DFSMShsm activity continues, and only the task that requests a tape device waits and reissues the tape device request later.

The setting for a NOWAIT request is shown in Example 7-1 on page 124.

Example 7-1 How to specify NOWAIT options in DFSMShsm

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

If the tape device request fails, DFSMShsm performs the following actions:

- Requests the tape device allocation six more times.
- After seven attempts to allocate a device, DFSMShsm sends a message to the operator console. The operator response is to either repeat or cancel the request.
- A message IEF238D might be issued if DFSMShsm detects spare devices as potential candidates for the tape mount. The operator must reply "cancel" to this request, if these devices are unusable. Message ARC0381A is issued when the request is ended.

> **Note:** The **SETSYS MOUNTWAITTIME(*nn*)** parameter in DFSMShsm specifies how long DFSMShsm waits. The numeric value in brackets specifies the number of minutes.

7.2.6 Specifying esoteric tape unit names to DFSMShsm

When DFSMShsm uses the **SETSYS USERUNITTABLE** command to specify esoteric tape unit names to DFSMShsm, DFSMShsm rejects mixed combinations of device names in an esoteric group.

Several exceptions exist to this rule:

- DFSMShsm allows the use of both 3480 and 3480X device names in a single group. Improved Data Recording Capability (IDRC), however, is not used with such a group because all devices in the group are treated as 3480s. If an esoteric group that is associated only with 3480Xs exists, 3480s must not be added to it because the data that was already written by using this esoteric tape unit name might create IDRC incompatibilities.
- Tape units 3592-1, 3592-2, and 3592-2E are allowed into the same esoteric because they share common write formats. Tape units 3592-2, 3592-2E, and 3592-3E are allowed into the same esoteric because they also share common write formats.

It is up to the user to ensure that the drives in a mixed esoteric will be used with a common recording technology. For example, if any 3592-1 drives are included, all drives in the esoteric must use EMFT1 for output. If the esoteric mixes 3592-2 and 3592-2E drives, all drives must be set up to use EFMT1 or EFMT2. If the esoteric contains mixed 3592-3E and 3592-2E tape units, all drives must be set up to use EFMT2 or EEFMT2, which are the common write formats.

If DFSMShsm rejects an esoteric tape unit name, it does not reject the rest of the valid esoteric names that are specified in the **USERUNITTABLE** command. Those names are now recognized by DFSMShsm as valid esoteric tape unit names. Each time that you specify **USERUNITTABLE**, the valid esoteric tape unit names that are identified with this parameter replace any esoteric tape unit names that are identified through a previous **USERUNITTABLE** parameter of the **SETSYS** command. The DFSMShsm tape environment allows the specification of tape unit names by using either generic or esoteric names.

Installations that have a mixture of non-SMS-managed 3590 devices that are defined under the 3590-1 generic name need to perform these steps:

1. Define a unique esoteric for each recording technology.
2. Use the **SETSYS USERUNITTABLE** command to define these esoteric names to DFSMShsm. This step also applies to mixed devices in the 3490 generic. Installations that use SMS-managed tape devices or with a single 3590-1 recording technology do not need to define an esoteric for those devices. However, if you have a mixed SMS-managed 3590 environment, the media type is required to be the same within a generic. The following example shows how to code a **USERUNITTABLE** parameter:

   ```
   USERUNITTABLE(3480 TAPEM VIRTUAL 3590-1 3590-2)
   ```

7.3 Protecting DFSMShsm-owned tapes

DFSMShsm tapes can be protected with the TAPEVOL resource class in RACF. However, the use of tape management systems replaces RACF for protecting tapes. Existing systems might still use RACF protection, so we describe this setup. RACF protection is not recommended for a new setup.

7.3.1 Protecting DFSMShsm tapes

To protect tapes that contain security-related data sets that are managed by DFSMShsm, you can protect tapes that are managed by DFSMShsm by installing and activating RACF.

Use one of the following options:

- Define to RACF the tapes you want to protect:
  - Defining the RACF TAPEVOL resource class in the RACF descriptor table (CDT).
  - Specifying the **SETSYS TAPESECURITY(RACF|RACFINCLUDE)** command.

> **Tip:** If you did not define RACF tape volume sets for DFSMShsm, but you want RACF to protect all of the tapes through a RACF generic profile, specify the **SETSYS TAPESECURITY(RACF|RACFINCLUDE)** command.

- Use RACF DATASET class protection by using one of the following options:
  - Using RACF SETROPTS TAPEDSN
  - Using DEVSUPxx TAPEAUTHDSN=YES

Using RACF TAPEVOL resources

You can protect volumes that are managed by DFSMShsm by using RACF TAPEVOL resources that are defined in the RACF descriptor table.

Protecting up to 10,000 tapes

If you are protecting up to a maximum of 10,000 tapes, you define two RACF resource names in the TAPEVOL resource class:

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

Issue the RACF commands that are shown in Example 7-2 on page 126.

Example 7-2 Define HSM resources in RACF

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

You can add tapes to RACF before DFSMShsm uses them. If you choose to add tapes to RACF, you must use the appropriate TAPEVOL. Use the command that is shown to add HSM tapes to RACF:

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

Recommendation: Do not add tapes to RACF, but instead let DFSMShsm add to the TAPEVOL automatically for you.

Protecting more than 10,000 tapes

To RACF-protect more that 10,000 tapes, you define multiple RACF resource names for DFSMShsm tape volume sets in the TAPEVOL resource class. The following resource names apply:

- HSMHSM (must be defined)
- HSMABR for aggregate backup and recovery tapes
- DFHSM*x*

  The *x* is a non-blank character (alphanumeric, national, or the hyphen) that corresponds to the last non-blank character of the tape volume serial number. You need to define a DFHSM*x* resource name for each *x* value that exists, based on your installation naming standards.

The RACF commands that are shown in Example 7-3 add resource names to the TAPEVOL class for HSMHSM (required), HSMABR (for aggregate backup and recovery tapes), and DFHSMA (for all tapes with a volume serial number that ends with the letter `A`):

Example 7-3 Define resource names to RACF TAPEVOL class

```
RDEFINE TAPEVOL HSMHSM
RDEFINE TAPEVOL HSMABR
RDEFINE TAPEVOL DFHSMA
   .    .    .
RDEFINE TAPEVOL DFHSMZ
RDEFINE TAPEVOL DFHSM@
RDEFINE TAPEVOL DFHSM$
RDEFINE TAPEVOL DFHSM#
RDEFINE TAPEVOL DFHSM-
RDEFINE TAPEVOL DFHSM0
   .    .    .
RDEFINE TAPEVOL DFHSM9
```

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

```
RALTER TAPEVOL HSMHSM ADDVOL(DFHSMx)
```

This command activates RACF TAPEVOL.

You can add RACF protection to the DFSMShsm tape volumes before DFSMShsm uses them, except for the HSMABR tapes. You must add the tape volume serial number to the appropriate DFHSM*x* tape volume set, which is based on the last non-blank character of the tape volume serial number. To protect a tape with a volume serial of POK33H, you can use a RACF command:

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

Important: Tapes that are already protected in the tape volume set of HSMHSM continue to be protected.

Using RACF DATASET class profiles

Protecting volumes that are managed by DFSMShsm by using RACF DATASET class profiles is described.

Tape data set authorization

You can use System Authorization Facility (SAF) to protect data sets on tape by using the RACF DATASET class without needing to activate the TAPEDSN option or the TAPEVOL class. In addition, you can specify that a user must have access to the first file on a tape volume to add files on that tape volume.

For optimum tape security, use the combined capabilities of DFSMSrmm, DFSMSdfp, and RACF. We recommend that you specify the following parameters:

- Specify these parameters in the DEVSUPxx PARMLIB member:
  - TAPEAUTHDSN=YES
  - TAPEAUTHF1=YES
  - TAPEAUTHRC4=FAIL
  - TAPEAUTHRC8=FAIL

  The parameters are described:

  TAPEAUTHDSN To enable tape authorization checks in the DATASET class.

  TAPEAUTHF1 Enables additional tape authorization checks in the DATASET class for existing files on the same tape volume when any other file on the tape volume is opened.

  TAPEAUTHRC4 Use this keyword to control PROTECTALL processing for tape data sets. This keyword applies to the results of RACROUTE processing when both TAPEAUTHDSN=YES and TAPEAUTHF1=YES are specified.

  TAPEAUTHRC8 Use this keyword as an aid to the implementation of TAPEAUTHDSN and TAPEAUTHF1.

  This task provides a managed and controlled implementation of tape authorization checks in the DATASET class, and applies only to the results of TAPEAUTHDSN=YES and TAPEAUTHF1=YES processing.

- Specify this parameter in the EDGRMMxx PARMLIB member:

  OPTION TPRACF(N)

  For TPRACF(NONE), DFSMSrmm does not create RACF tape profiles for any volumes in your installation.

- Specify these parameters in RACF:

  SETROPTS NOTAPEDSN NOCLASSACT(TAPEVOL)

You must create generic DATASET class profiles that cover each of the data set name prefixes that DFSMShsm currently uses. Use the **RACF ADDDATASET** commands that are shown in Example 7-4.

Example 7-4 RACF ADDDATASET command to protect your DFSMShsm data sets

```
ADDSD 'mprefix.**'
ADDSD 'bprefix.**'
ADDSD 'authid.**'
ADDSD 'bprefix.DMP.**'
ADDSD 'outputdatasetprefix.**'
```

The parameters in Example 7-4 are described:

mprefix.** Specifies the DFSMShsm-defined migrated data set prefix

bprefix.** Specifies the DFSMShsm-defined backup and dump data set prefix

authid.** Specifies the DFSMShsm prefix that is used for CDS backups

bprefix.DMP.** If you want dump tapes to be protected differently than back up tapes

outputdatasetprefix.**
Specifies that ABARS-created aggregates are protected

The combination of DFSMSrmm, DFSMSdfp, and RACF ensures the following conditions:

- Full 44-character data set name validation.
- Validation that the correct volume is mounted.
- Control of the overwriting of existing tape data sets.
- Management of tape data set retention.
- Control over the creation and destruction of tape volume labels.
- No limitations are caused by RACF TAPEVOL profile sizes and tape volume table of contents (TVTOC) limitations.
- Common authorization of all tape data sets on a volume.
- Use of generic DATASET profiles, enabling common authorization with DASD data sets.
- Authorization for all tape data sets regardless of the tape label type.
- Authorization for the use of bypass label processing (BLP).
- Exploitation of RACF erase on scratch support.
- Use of DFSMSrmm FACILITY class profiles for data sets that are unprotected by RACF.

Your authorization to use a volume outside of DFSMSrmm control through ignore processing also enables authorization to the data sets on that volume. To aid in the migration to this environment, DFSMSrmm provides the TPRACF(CLEANUP) option, and DEVSUPxx provides TAPEAUTHRC8(WARN) and TAPEAUTHRC4(ALLOW). The function in DFSMSdfp does not replace all of the functional capabilities that the RACF TAPEDSN option, TAPEVOL class, and TVTOC provide. However, together with the functions that DFSMSrmm provides, the capability is equivalent.

The enhanced DFSMSdfp function addresses the authorization requirements for tape data sets and relies on your use of a tape management system, such as DFSMSrmm, to perform the following operations:

- Verify full 44-character data set names.
- Control the overwriting of existing tape files.

- Handle tape data set retention.
- Control the creation and destruction of tape labels.

> **Important:** With the SAF tape security implementation, as opposed to a TAPEVOL and TAPEDSN implementation, you can perform the following tasks:
>
> - Write more than 500 data sets to one tape or one tape volume set.
> - A tape volume set can expand to more than 42 tape volumes.
> - Write duplicate data set names to one tape or one tape volume set.
> - The number of volumes that can be protected is not limited.

Use the MVS **SET DEVSUP** command to implement the new tape data set security settings. Figure 7-1 shows the successful result of the command.

```
T DEVSUP=18
IEE252I MEMBER DEVSUP18 FOUND IN SYS1.PARMLIB
IEE536I DEVSUP VALUE 18 NOW IN EFFECT
IEA253I DEVSUP 3480X RECORDING MODE DEFAULT IS COMPACTION.
IEA253I DEVSUP ISO/ANSI TAPE LABEL VERSION DEFAULT IS V3
IEA253I DEVSUP TAPE OUTPUT DEFAULT BLOCK SIZE LIMIT IS 32760
IEA253I DEVSUP COPYSDB DEFAULT IS INPUT
IEA253I DEVSUP TAPEAUTHDSN: YES
IEA253I DEVSUP TAPEAUTHF1: YES
IEA253I DEVSUP TAPEAUTHRC4: FAIL
IEA253I DEVSUP TAPEAUTHRC8: FAIL
```

Figure 7-1 Use and result of the SET DEVSUP command

In Figure 7-1, `18` is the suffix of the device support option member DEVSUP*nn* that needs to be activated.

7.4 DFSMShsm interaction with a tape management system

DFSMShsm communicates with other components of your z/OS environment through exits, including your tape management system. ARCTVEXT is the installation-wide exit that allows communication between DFSMShsm and your tape management system, except for DFSMSrmm.

To tell DFSMShsm to enable the ARCTVEXT installation-wide, add the following **SETSYS** command in the ARCCMDxx PARMLIB member. This command shows how to enable the ARCTVEXT exit for non-Data Facility Removable Media (DFRMM) tape management systems:

```
SETSYS EXITON(ARCTVEXT)
```

From z/OS V1.4 forward, the EDGTVEXT general-use programming interface is called by DFSMShsm to communicate with DFSMSrmm. DFSMShsm automatically calls DFSMSrmm to process tapes to be returned to the DFSMShsm tape pool or deleted from DFSMShsm.

The interface between DFSMShsm and DFSMSrmm is automatically created when you install DFSMS. A benefit of this interaction is that DFSMSrmm can prevent DFSMShsm from overwriting its own CDS backup, automatic dump, ABARS, and copies of backup or migration tapes.

You can manage the retention of valid tape volumes in two ways:

- Use DFSMSrmm vital record specifications (VRS) to retain tape volumes until DFSMShsm finishes with them. More details about how to define a VRS are provided later.
- Set up DFSMShsm to use expiration date-protected tape volumes. Use the date `99365` to prevent DFSMSrmm from considering the volumes for release at any time.

Use VRS because you can avoid expiration dates, operator message handling, and the need to re-initialize tape volumes before you can reuse them. However, if you use a tape pool that is managed by DFSMShsm, you can use expiration date protection without this overhead; DFSMShsm can override expiration protection for its own tape volumes. DFSMShsm uses the DFSMSrmm EDGTVEXT interface to maintain the correct volume status. DFSMSrmm provides a programming interface so you can use the DFSMShsm exit.

The following example shows how to code the security setting for tapes, which are shown from the DFSMShsm perspective:

```
TAPESECURITY(EXPIRATION)
```

7.5 Running DFSMShsm with DFSMSrmm

DFSMSrmm can provide enhanced management functions for the tape volumes that DFSMShsm uses for each of its tape functions. The manner in which the two products work together depends on how you use each of them in your installation. Run DFSMSrmm with DFSMShsm to enhance the management of the volumes that DFSMShsm uses. For example, DFSMSrmm can manage the movement of tapes that must be sent out of the library for disaster recovery. By using DFSMSrmm and DFSMShsm together, you can use the scratch tape pool rather than a DFSMShsm tape pool.

DFSMSrmm provides the EDGTVEXT and EDGDFHSM programming interfaces that can be used by products, such as DFSMShsm, object access method (OAM), and IBM Tivoli® Storage Manager. Use these programming interfaces for DFSMSrmm tape management so that you can maintain correct volume status.

DFSMSrmm treats DFSMShsm like any other tape volume user and retains DFSMShsm volumes based on VRSs and retention period. DFSMShsm automatically calls EDGTVEXT so you do not need to set up DFSMShsm to communicate with DFSMSrmm.

You can use the TVEXTPURGE PARMLIB option in the DFSMSrmm EDGRMMxx PARMLIB member to control the action that DFSMSrmm takes when DFSMShsm calls EDGTVEXT. A benefit of this interaction is that DFSMSrmm can prevent DFSMShsm from overwriting its own CDS backup, automatic dump, ABARS, and copies of backup or migration tapes. Although DFSMShsm checks its own migration and its own backup tapes, DFSMSrmm checks them, also.

7.5.1 Authorizing DFSMShsm to DFSMSrmm resources

Before you can use DFSMSrmm with DFSMShsm, you are required to authorize DFSMShsm to the RACF resource profiles STGADMIN.EDG.MASTER, STGADMIN.EDG.OWNER.user, and STGADMIN.EDG.RELEASE.

If you have multiple DFSMShsm user IDs, for example, in a multisystem or multihost environment, and any DFSMShsm ID can create tapes or return tapes to scratch status or return tapes to the DFSMShsm tape pool, you must authorize each DFSMShsm user ID.

Define STGADMIN.EDG.OWNER.*hsmid* for each DFSMShsm user ID and give the other DFSMShsm user IDs UPDATE access to it.

Table 7-2 and Table 7-3 show the necessary access level for DFSMShsm to manage scratch tapes from a global scratch pool or a scratch pool that is managed by DFSMShsm. This access is needed to use DFSMShsm with a global scratch pool.

Table 7-2 Required authorization to use scratch tapes with DFSMShsm

Resource	Access required
STGADMIN.EDG.RELEASE	READ
STGADMIN.EDG.MASTER	READ
STGADMIN.EDG.OWNER.hsmid	UPDATE

This access is necessary for a DFSMShsm-owned pool.

Table 7-3 Required authorization to use DFSMShsm with a DFSMShsm scratch pool

Resource	Access required
STGADMIN.EDG.MASTER	UPDATE
STGADMIN.EDG.OWNER.hsmid	UPDATE

7.5.2 Authorizing ABARS to DFSMSrmm resources

To use DFSMSrmm with DFSMShsm ABARS, you must assign ABARS IDs the correct levels of authorization to STGADMIN.EDG.MASTER, STGADMIN.EDG.OWNER.*user*, and STGADMIN.EDG.RELEASE. If you have multiple ABARS user IDs, for example in a multisystem environment, and any ABARS ID can return tapes to scratch status, you must authorize each ABARS user ID. Define STGADMIN.EDG.OWNER.*abarsid* for each ABARS user ID and give the other ABARS user IDs UPDATE access to it. This method allows one ABARS ID to release the tapes initially that are obtained from scratch by the other ABARS ID.

7.5.3 Setting DFSMSrmm options when using DFSMShsm

You use the DFSMSrmm PARMLIB EDGRMMxx to specify the installation options for DFSMSrmm as described in ““EDGRMMxx PARMLIB member” on page 141.

If you use expiration dates to manage tapes, consider the values that you specify for the **parmlib** command MAXRETPD, RETENTIONMETHOD, and TVEXTPURGE operands and the **VLPOOL** command EXPDTCHECK operand:

- The MAXRETPD operand specifies the maximum retention period that a user can request for data sets on volumes.
- The RETENTIONMETHOD operand specifies the retention method (EXPDT or VRSEL) that DFSMSrmm uses for DFSMShsm volumes.
- The TVEXTPURGE operand specifies how you want to handle the release of DFSMShsm tape volumes.
- The **VLPOOL** command EXPDTCHECK operand tells DFSMSrmm how to manage a volume based on the expiration date field in the volume label.

You can use the EXPDT retention method to avoid the processing of DFSMShsm volumes on each inventory management VRSEL run. This retention method requires that you use expiration date protection for DFSMShsm tape volumes, EXPDT=`99365`.

When DFSMShsm sets `99365` as the expiration date to manage its tapes, `99365` means permanent retention or no expiration. If you choose to use DFSMShsm expiration date-protected tape volumes, DFSMShsm sets the date `99365` to prevent DFSMSrmm from considering the volumes for release at any time. You must specify the MAXRETPD(NOLIMIT) operand to ensure that DFSMSrmm honors the `99365` date.

You also use the DFSMSrmm PARMLIB MAXRETPD operand value to reduce the expiration date for all volumes, including DFSMShsm volumes. If you want to reduce the `99365` permanent retention expiration date, specify the MAXRETPD with a value `0` - `99999` days.

Recommendation: If you want to avoid the overhead of VRSEL processing for tape volumes that are managed by DFSMShsm (and similar applications), use the EXPDT retention method. If, however, you require DFSMSrmm to manage the movement of DFSMShsm volumes, use DFSMSrmm VRS instead of the `99365` permanent retention date to retain DFSMShsm volumes.

7.5.4 Setting DFSMShsm options when using DFSMSrmm

As DFSMShsm uses a tape volume, DFSMSrmm records information about data sets and multivolume data sets at OPEN time. DFSMSrmm can use this information to manage the volumes based on the DFSMSrmm policies that you define. DFSMShsm uses the data control block (DCB) Open/end of volume (EOV) volume security and verification exit to ensure that an acceptable volume is mounted for DFSMShsm use. DFSMShsm uses this exit to reject unacceptable volumes. For example, DFSMShsm rejects volumes that are already in use by DFSMShsm and volumes that are not authorized for use by DFSMShsm. DFSMSrmm records information only for those volumes that are not rejected by DFSMShsm.

DFSMSrmm provides facilities so that DFSMShsm can tell DFSMSrmm when it no longer requires a tape volume or when a tape volume changes status. The benefit is that DFSMShsm cannot mistakenly overwrite one of its own tape volumes if an operator mounts a tape volume in response to a request for a non-specific tape volume.

7.5.5 Setting DFSMShsm system options

The DFSMShsm system options that relate to the use of a tape management system other than DFSMSrmm with global or private scratch pools are shown in Figure 7-11 on page 141.

Example 7-5 DFSMShsm system options

```
SETSYS EXITON(ARCTVEXT)/EXITOFF(ARCTVEXT) -
       SELECTVOLUME                       -
       TAPEDELETION                       -
       TAPESECURITY                       -
       PARTIALTAPE
```

The ARCVEXT is activated for a non-DFSMSrmm tape management system.

7.5.6 Setting DFSMShsm dump definitions

The DFSMShsm dump definitions are shown in Example 7-6.

Example 7-6 Setting DFSMShsm DUMP parameters

```
DEFINE DUMPCLASS(class -
       AUTOREUSE -
       TAPEEXPIRATIONDATE(date) -
       RETENTIONPERIOD(day))
```

The use of **AUTOREUSE** ensures that tapes are returned to the global scratch pool when they expire.

7.5.7 DFSMSrmm support for DFSMShsm naming conventions

DFSMSrmm supports all DFSMShsm options and any of the naming conventions for DFSMShsm tape data sets, except for password security on tape volumes.

DFSMSrmm support for retention and pooling

With DFSMShsm, you can use DFSMSrmm system-based scratch tape pools, exit-selected scratch pools, or tape pools that are managed by DFSMShsm.

Recommendation: Use a DFSMSrmm scratch pool. Use the pool that is managed by DFSMShsm only when necessary. For example, use the pool that is managed by DFSMShsm if you want to keep pools that are managed by DFSMShsm for Enhanced Capacity Cartridge System Tapes because DFSMShsm fully uses a tape's capacity. You must let DFSMShsm decide whether a tape volume contains valid data and whether it returns to the DFSMSrmm scratch pool or a tape pool that is managed by DFSMShsm.

Define VRSs to retain tape volumes until DFSMShsm finishes with them. DFSMSrmm uses the retention period that is determined by the VRS to extend any expiration date or retention period that was previously set for the volume. Additionally, you can use VRSs to identify volumes that need to be moved out of the installation media library for safekeeping, or moved from an ATL to a manual library. For tapes that are managed by DFSMShsm, you do not need to respond to the IEC507D messages that are issued for expiration data-protected tapes because DFSMShsm can override expiration for its own tapes. If you choose to use DFSMShsm expiration date protected tape volumes, DFSMShsm sets the expiration date to 99365, which means permanent retention to prevent DFSMSrmm from considering the volumes for release at any time.

Retaining DFSMShsm tapes by using expiration dates

To use DFSMShsm expiration date protection, specify the DFSMShsm startup option:

```
SETSYS TAPESECURITY(EXPIRATION)
```

If you use a system scratch tape pool for DFSMShsm tapes, you need a way to manage tapes that are protected with expiration dates that are set by DFSMShsm. To help you manage this situation, you can automate the responses to expiration date protection messages for scratch pool tape volumes with DFSMSrmm. Use the PARMLIB member **VLPOOL** command to set up this automation. Set the VLPOOL EXPDTCHECK operand to EXPDTCHECK(N). DFSMSrmm automatically lets your users reuse the volumes in the pool without operator intervention and without creating data integrity exposures.

If you use a tape pool that is managed by DFSMShsm, DFSMShsm validates and overrides expiration dates on its emptied, previously used tapes.

Defining VRSs to manage DFSMShsm tapes

Movement and retention policies are defined by using VRSs by specifying data set names or volume serial numbers. To define VRSs, use the **RMM ADDVRS** subcommand. *DFSMSrmm Managing and Using Removable Media,* SC26-7404, offers information about the **RMM ADDVRS** subcommand and the operands you can specify. When you specify the **RMM ADDVRS** command operands, it might be helpful to think about the operands in three categories:

- Operands to define retention policies, including COUNT and CYCLES that are used in these examples
- Operands to define movement policies, including DELAY, LOCATION, and STORENUMBER, that are used in these examples
- Operands to manage the VRS itself, including DELETEDATE and OWNER that are defaults in the **RMM ADDVRS** subcommand:
  - DELETEDATE(1999/365) specifies the date when the VRS no longer applies. The default value is 1999/365, which means the VRS is permanent. It can be manually deleted only if it is no longer appropriate.
  - OWNER(*owner*) specifies the user ID that owns the VRS.

As shown in the following examples, you can specify data set name masks in VRSs to manage the following functions:

- Migration
- Backup
- Dumps
- TAPECOPY
- DUPLEX tape feature
- Tapes that are written by ABARS
- ABARS accompany tapes
- CDS version backups

You can tailor the examples to define policies for your DFSMShsm tapes. As you gain more experience defining VRSs, you see several ways to define the retention and movement policies you want.

The examples use the current DFSMShsm data set naming formats. Certain older name formats were in use before APAR OY20664. APAR OY20664 requires you to use **PATCH** commands to use the new format. The new format is standard in DFSMShsm. If the older naming formats exist in your installation, you might need to define VRSs that use both naming formats. Delete the VRSs with the old naming format when the old names no longer exist. For more information, see *z/OS DFSMShsm Implementation and Customization Guide,* SC35-0418. This book also provides detailed information about retaining all types of DFSMShsm tapes by using VRS.

Examples of DFSMSrmm panels

The easy way to enter the DFSMSrmm panels is to select the Interactive Storage Management Facility (ISMF) Primary Option Menu panel and enter R for the Removable Media Manager for the DFSMSrmm option, as shown in Figure 7-2 on page 135.

```
ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R13
Selection or Command ===>

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
G  Report Generation        - Create Storage Management Reports
L  List                     - Perform Functions Against Saved ISMF Lists
P  Copy Pool                - Specify Pool Storage Groups for Copies
R  Removable Media Manager  - Perform Functions Against Removable Media
X  Exit                     - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 7-2 ISMF Primary Option Menu

After your selection, the primary DFSMSrmm panel appears, as shown in Figure 7-3. Enter 3 to view the Administrator functions.

```
REMOVABLE MEDIA MANAGER (DFSMSrmm) - z/OS V1R13
Option ===>

0  OPTIONS       - Specify dialog options and defaults
1  USER          - General user facilities
2  LIBRARIAN     - Librarian functions
3  ADMINISTRATOR - Administrator functions
4  SUPPORT       - System support facilities
5  COMMANDS      - Full DFSMSrmm structured dialog
6  LOCAL         - Installation defined dialog
X  EXIT          - Exit DFSMSrmm Dialog

Enter selected option or END command. For more info., enter HELP or PF1.
```

Figure 7-3 Option 3 Administrator is selected

The panel that is shown in Figure 7-4 on page 136 is displayed. First, we work with VRSs. Enter option 3 to reach this function.

```
                    DFSMSrmm Administrator Menu - z/OS V1R13
Option ===>

0  OPTIONS    - Specify dialog options and defaults
1  VOLUME     - Display or change volume information
2  OWNER      - Display or change owner information
3  VRS        - Vital record specifications

Enter selected option or END command. For more info., enter HELP or PF1.
```

Figure 7-4 Option 3 is selected to work with VRSs

To add a VRS, enter option 2, as shown in Figure 7-5.

```
                      DFSMSrmm Vital Record Specification Menu
Option ===>

0  OPTIONS    - Specify dialog options and defaults
1  DISPLAY    - Display a vital record specification
2  ADD        - Add a vital record specification
3  CHANGE     - Change a vital record specification
4  DELETE     - Delete a vital record specification
5  SEARCH     - Search for vital record specification


Enter selected option or END command. For more info., enter HELP or PF1.
```

Figure 7-5 Option 2 to add a VRS is selected

In Figure 7-6, enter a data set mask.

```
                   DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . . 'hsm.dmp.**'
       Job name mask . .          Add data set filter VRS . .    (Yes or No)

  Volume serial . .         ( May be generic )

  VRS name  . . . .
```

Figure 7-6 DFSMSrmm Add Vital Record Specification

In Figure 7-7 on page 137, enter the count, retention type, location, owner, description, and deletion date.

```
                    DFSMSrmm Add Data Set VRS
Command ===>

Data set mask . : 'HSM.DMP.**'                                   GDG . . NO
     Job name mask . :

Count . . . . 99999                 Retention type  . . . . . . DAYS
                                    While cataloged . . . . . .
Delay . . . .     Days              Until expired . . . . . . .

Location  . . . . . . CURRENT
Number in location
Priority  . . . . . . 0
                                    Release options:
Next VRS in chain . .               Expiry date ignore  . . . .
    Chain using . . .               Scratch immediate . . . . .

Owner . . . . . . HSMTASK
Description . . . RETAIN ALL DFSMSHSM DUMP DS
Delete date . . . 1999/365    ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 7-7 DFSMSrmm Add Data Set VRS

Table 7-4 describes the fields that are used to add a VRS.

Table 7-4 DFSMSrmm fields to add a VRS

Field	Field descriptions
Data set mask	Specify the fully or partially qualified data set name.
Job name mask	This field can be left blank because DFSMShsm creates the tapes.
Generation data group (GDG)	DFSMShsm does not create GDGs, so enter `NO`.
Description	Any valid description that is meaningful for the VRS.
Owner	DFSMShsm is a large user of your tape resources. You need to create a unique DFSMSrmm owner. In this case, we created `HSMTASK` as the owner for all DFSMShsm tapes. The *z/OS DFSMShsm Implementation and Customization Guide,* SC35-0418, describes how to create an owner.
Retention type	If you want to send your tapes offsite or to the vault for a specific period, for example, 14 days for your dump tapes, specify `DAYS`. This value relates to the Store number in the location field.
While cataloged	Specify `NO` for DFSMShsm data sets.
Until expired	Specify `NO` for DFSMShsm data sets.

Field	Field descriptions
Location	Valid entries are HOME, LOCAL, DISTANT, and REMOTE, but see the *z/OS DFSMShsm Implementation and Customization Guide,* SC35-0418.
Delete date	This field specifies the date when the VRS will be deleted. DFSMSrmm uses the default value of 1999/365, which means "never delete".
Vital record count	99999 indicates that all vital records are kept.
Delay	You can delay the movement of your offsite data. A 0 indicates to move the tapes immediately. A 1 indicates to move the tapes the following day.
Store number in location	This value indicates the number of days that any particular tape is to stay offsite. We used DAYS in the Retention type field so our tapes will stay offsite for 14 days in location LOCAL. (LOCAL can be any storage facility in the nearby area.)
Next VRS name	Use this field only if you are planning to stage your tape movements, for example, DISTANT, then REMOTE, and then LOCAL. This field points to the next VRS to look at in the chain.

To see the vital records that are already defined in DFSMSrmm, enter option 5 SEARCH to search for vital record specifications on the DFSMSrmm VRS menu. Figure 7-8 shows a generic search for all vital records that are defined in this DFSMSrmm.

```
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                             DFSMSrmm Search VRSs
Command ===>

Optionally specify one of:
Data set mask                                                   GDG  . .
     Job name mask                                              ( Yes or No )
Volume serial                   Retention type
VRS name  . . .                 While cataloged      ( Yes or blank for all)
                                Until expired        ( Yes or blank for all)
Location  . . . .               Release options:
Next VRS in chain               Expiry date ignore       ( Yes or No )
  Chain using . .               Scratch immediate        ( Yes or No )
Owner . . . . . . *
Limit . . . . . . *             Limit search to first n VRSs. Default is *
Dates            Start          End       Date, date range or relative value
  Reference . .             . .
  Changed . . .             . .
Clist . . . . . .               YES to create a data set, or NO, or blank

Enter HELP or PF1 for the list of available line commands
```

Figure 7-8 Generic search for all vital records in DFSMSrmm

Tip: To see all defined VRS definitions, leave all fields blank except for asterisks in the OWNER and LIMIT fields.

Figure 7-9 shows an example of the search results.

```
                         DFSMSrmm VRSs (Page 1 of 4)          Row 1 to 3 of 3
Command ===>                                                  Scroll ===> PAGE

Enter HELP or PF1 for the list of available line commands.
Use the LEFT and RIGHT commands to view other data columns.

S  Volume/Data set/Name specification          Job name Type Location Prty
-- -------------------------------------------- -------- ---- -------- -----
   ABEND                                                DSN  HOME     0
   ABEND                                       DFHSM*   DSN  HOME     0
   ...
   HSM.**                                               DSN  HOME     0
   HSM.DMP.**                                           DSN  CURRENT  0
   ...
   RMM.F13002.**                                        DSN  HOME     0
   ...
   OPEN                                                 DSN  HOME     0
   OPEN                                        DFHSM*   DSN  HOME     0
```

Figure 7-9 Listing of vital records in DFSMSrmm

Figure 7-10 on page 140 shows an example of how to create a vital record to protect your daily DFSMShsm backup tapes.

```
                         DFSMSrmm Add Data Set VRS
Command ===>

Data set mask . : 'HSM.BACKTAPE.DATASET'                        GDG . . NO
     Job name mask . :

Count . . . . 99999                Retention type  . . . . . . cycles
                                   While cataloged . . . . . . NO
Delay . . . . 0   Days             Until expired . . . . . . . NO

Location  . . . . . . CURRENT
Number in location    99999
Priority  . . . . . . 0
                                   Release options:
Next VRS in chain . .              Expiry date ignore  . . . . NO
    Chain using . . .              Scratch immediate . . . . . NO

Owner . . . . . . HSMTASK
Description . . . HSM backup tapes
Delete date . . . 1999/365    ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 7-10 Creating VRSs for DFSMShsm backup tapes

In this example, we changed three fields:

- Retention type was changed to `CYCLES` because we want any tape that uses this VRS to continue to use it until the tape is either recycled, expires naturally, or is deleted.
- Location can be changed to `HOME` if you do not want daily backup tapes sent offsite.
- Number in location is changed to `99999`, which is the DFSMSrmm default. This value works with the CYCLES selection to ensure that all tapes are retained in location HOME.

A similar vital record definition must be created for your migration tapes.

TVEXTPURGE extra days

With z/OS V1R12 DFSMSrmm, the **TVEXTPURGE** parameter used the options `RELEASE`, `EXPIRE`, and `NONE`. With z/OS V1.13, DFSMSrmm has a new option for the `EXPIRE`(*days*).

If tapes are expired by using the EDGTVEXT HSM exit, extra days for retention can be defined with no additional processing.

Use of extra days

You can use DFSMSrmm to release DFSMShsm tapes that are requested to be purged by DFSMShsm. You can also specify that DFSMSrmm retain a tape for a few days after its expiration date is reached. By default, the expiration date for DFSMShsm tapes is protected by DFSMShsm. DFSMShsm uses `1999/365` as the expiration date for permanent retention. To enable extra days of retention for purged DFSMShsm tape volumes, use the **TVEXTPURGE(EXPIRE(*days*))** option or set retention options in the VRSs. These specifications are used to retain the tape volumes.

EDGRMMxx PARMLIB member

This member specifies how DFSMSrmm processes volumes that are purged by callers of EDGTVEXT or EDGDFHSM. The TVEXTPURGE operand specifies how you want to handle the release of DFSMShsm tape volumes, by using the new specification that is shown in Figure 7-11.

```
                               |—TVEXITPURGE(RELEASE)——————————————————|
                               |                                       |
▸ —OPTION————————————————————————————————————————————————————————————————————◂◂
                               |                                       |
                               |—TVEXITPURGE(--———————————————————————)—|
                                       |                           |
                                       |          |—0————|         |
                                       |—EXPIRE(——————————————)—|
                                       |          |—days—|         |
                                       |—NONE——————————————————————|
```

Figure 7-11 EDGTVEXT TVEXTPURGE operand

The command has the following parameters:

EXPIRE(*days*)
: Use the EXPIRE(*days*) option to set the volume expiration date to the current date plus *days* for volumes to be purged. Use the EXPIRE(*days*) parameter when you use the EXPDT retention method, by using days as a way to delay expiration of the volume. When you use the VRSEL retention method, you can optionally use this operand in combination with VRSs that use the UNTILEXPIRED retention type. Apply EXPIRE(*days*) to set a new volume EXPDT. Then, run VRSEL to extend retention by using the extra days retention type. For example, specify EXPIRE(0) when you use a VRS with extra days retention. You can also use a non-zero EXPIRE(*days*) value and avoid using an extra days retention VRS. The *days* parameter is the number of days that DFSMSrmm retains the volume before the volume is considered for release. The value is a 1-digit to 4-digit decimal number and is added to today's date to compute the new expiration date. If the value exceeds the maximum retention period (MAXRETPD), it is reduced to the MAXRETPD value. The default value for days is 0. TVEXTPURGE(EXPIRE) is the same as TVEXTPURGE(EXPIRE(0)).

NONE
: DFSMSrmm takes no action for volumes to be purged.

RELEASE
: DFSMSrmm releases a volume to be purged according to the release actions that are set for the volume. You must run expiration processing to return a volume to scratch status. This parameter is the default. You can specify that volumes that are purged from DFSMShsm are to be retained for a few extra days. With this process, you can ensure that purged volumes do not contain any data that might still be needed. DFSMShsm migration and backup volumes can be retained by EXPDT=`99365`. You can optionally set EXPDT to the current date or a future date when the volume is purged from DFSMShsm with EDGTVEXT. You can release volumes or set the volume expiration date either to the current date or based on a number of days from the current date. The effect of setting the expiration date depends on the retention method (EXPDT or VRSEL) that is already specified for the volume.

How to code expiration in DFSMShsm and DFSMSrmm

Coding expiration in DFSMShsm and DFSMSrmm is described.

DFSMShsm

HSM supports the changes to the expiration date that were described previously. In summary, when DFSMShsm sets 99365 as the expiration date to manage its tapes, 99365 means permanent retention or no expiration. If you choose to use DFSMShsm expiration date-protected tape volumes, DFSMShsm sets the date to 99365. This setting prevents DFSMSrmm from considering the volumes for release at any time. You must specify the MAXRETPD(NOLIMIT) operand to ensure that DFSMSrmm recognizes the 99365 date.

DFSMSrmm

You also use the DFSMSrmm PARMLIB MAXRETPD operand value to reduce the expiration date for all volumes that include DFSMShsm volumes. If you want to reduce the 99365 permanent retention expiration date, specify the MAXRETPD with a value of 0 - 99999 days. If you want to avoid the processor needs of VRSEL processing for tape volumes that are managed by DFSMShsm (and similar applications), use the EXPDT retention method. If you require DFSMSrmm to manage the movement of DFSMShsm volumes, use DFSMSrmm VRSs instead of the 99365 permanent retention date to retain DFSMShsm volumes.

Managing DFSMShsm tapes by using EDGDFHSM

DFSMShsm uses the EDGDFHSM programming interface to release DFSMShsm tape volumes in DFSMSrmm. The interface between DFSMShsm and DFSMSrmm is automatically in place when you install DFSMS, so you do not need to activate it or to call it. Any caller of EDGDFHSM must be defined to RACF. If the caller is a started task, define the user ID with the STARTED class. Authorize your application to release its own tape volumes.

You must also consider how volumes are retained until the application calls the EDGDFHSM programming interface to release the volumes. You can use this program interface from programs other than DFSMShsm to release tape volumes. For information about setting up DFSMShsm with DFSMSrmm, see Chapter 14, “Problem determination” on page 401. You can use this information as an example for setting other applications that manage tape. You can retain tapes by defining VRSs such as the VRSs that are shown in this example.

Example: Define policies to retain all of the data until a volume is released by the application. See Example 7-7.

Example 7-7 Defining policies to retain data until a volume is released by the application

```
RMM ADDVRS DSN('**') JOBNAME(jobname) LOCATION(CURRENT) DAYS COUNT(99999)
RMM ADDVRS DSN('ABEND') JOBNAME(jobname) LOCATION(CURRENT) DAYS COUNT(99999)
RMM ADDVRS DSN('DELETED') JOBNAME(jobname) LOCATION(CURRENT) DAYS COUNT(99999)
RMM ADDVRS DSN('OPEN') JOBNAME(jobname) LOCATION(CURRENT) DAYS COUNT(99999)
```

Another way to retain tapes is with the EXPDT retention method. You can use the DFSMSrmm PARMLIB OPTION TVEXTPURGE operand to control the processing that EDGDFHSM programming interface performs. You can release volumes or set the volume expiration date either to the current date or based on a number of days from the current date. The effect of setting the expiration date depends on the retention method (EXPDT or VRSEL) that is already specified for the volume. For more information, see TVEXTPURGE, which is described in ““EDGRMMxx PARMLIB member” on page 141.

Callers of the EDGDFHSM programming interface can request the processing of volumes only when their RACF user ID is authorized, as defined by the RACF FACILITY class profiles for DFSMSrmm.

7.6 Using an SMS-managed tape library

A common tape setup for a DFSMShsm environment is based on SMS-managed tape libraries. A system-managed tape library consists of tape volumes and tape devices that are defined in the tape configuration database.

The tape configuration database is an integrated catalog facility user catalog that is marked as a volume catalog (VOLCAT) and contains tape volumes and tape library records.

A system-managed tape library can be either automated or manual. An *automated tape library* (ATL) is a device that consists of robotic components, cartridge storage areas (or shelves), tape subsystems, and controlling hardware and software, with the set of tape volumes that reside in the library and can be mounted on the library tape drives. A *manual tape library* (MTL) is a set of tape drives and the set of system-managed volumes the operator can mount manually on those drives.

If backup or migration uses a system-managed tape library, specify the **`LIBRARYBACKUP`** or **`LIBRARYMIGRATION`** parameters with the **`SETSYS TAPEUTILIZATION`** command. Virtual tape systems generally use a PERCENTFULL value of 97%, unless a larger value is needed for virtual tapes that are larger than the nominal 400 MB standard capacity MEDIA1 or 800 MB enhanced capacity MEDIA2 tapes.

In the case of the newer virtual tape systems (TS7700 Release 1.4 and later), where DFSMShsm derives media capacity by checking the mounted virtual tape, DFSMShsm allows a PERCENTFULL value up to 110%. Any larger value is reduced to 100%.

For older virtual tape systems, where DFSMShsm cannot dynamically determine virtual tape capacity, PERCENTFULL values that are larger than 110% are honored.

7.6.1 Converting to a DFSMS-managed library

Consider how you will convert to an SMS-managed tape library. Is this move a physical move of all tapes into the library? Alternatively, it might be a DFSMShsm RECYCLE into the library from the old DFSMShsm tapes with directing new allocations to the library? It might be a combination of these two approaches.

Information about the physical move approach is described. In this scenario, we focus only on DFHSM.

Considerations for moving tapes into an SMS-managed library

The tapes that you insert in an SMS-managed tape library either contain valid data or are empty.

Tapes that contain valid data need to be in private status. They must be associated with a tape storage group. And, they need to be assigned the correct recording technology, compaction, and media type through data class assignment.

Empty tapes that are not defined to DFSMShsm need to be assigned a scratch status and associated with the global scratch pool. The use of a global scratch pool is the recommended solution to assign scratch tapes to DFSMShsm today.

Empty tapes that are defined to DFSMShsm (specific scratch pool) need to be assigned a PRIVATE status and they need to be associated with the tape storage group for the DFSMShsm function to which the tapes were set up with ADDVOL. After a storage group is assigned to a tape volume, the storage group cannot be changed except with ISMF, JCL, or by returning the tape to scratch status.

7.6.2 Identifying tape library tapes

First, ensure that all of the esoterics that point to the library are in place. Adjust your USERUNITTABLE in the DFSMShsm PARMLIB to ensure that all of the esoterics that point to the library are in place.

DFSMShsm uses connected sets. Identify these sets and ensure that you move these connected sets together as the transition into a library occurs. You can move all of your DFSMShsm tapes into the library. However, it might be helpful to test the connectivity and functionality before you move all of your tapes. You can use the DFSMShsm **LIST** command to identify a connected set.

First, look at the tapes that are connected (or for tapes that are not connected). Use the commands that are shown in Example 7-8.

Example 7-8 DFSMShsm list of connected or unconnected sets

```
LIST TTOC SELECT(CONNECTED)
LIST TTOC SELECT(NOTCONNECTED)
```

Also, look at the filling criteria on your current tape. The commands that are shown in Example 7-9 can be useful.

Example 7-9 DFSMShsm list of filling criteria on the DFSMShsm tapes

```
LIST TTOC SELECT(FULL)
LIST TTOC SELECT(NOTFULL)
LIST TTOC SELECT(EMPTY)
```

You might want to stop using the empty tapes. You can set the empty tapes as DELVOL. Specify MARKFULL on the partially filled tapes.

To identify your disaster recovery tapes (duplex or alternate volumes), issue the **LIST TTOC** command, as shown in Example 7-10.

Example 7-10 DFSMShsm list of DUPLEX or alternate volumes

```
LIST TTOC SELECT(ALTERNATEVOLUME)
LIST TTOC SELECT(DISASTERALTERNATEVOLUME)
```

These tapes might be candidates for future ejects or they can be transferred to an offsite library. The following command shows a DFSMShsm list of alternate tapes with errors:

```
LIST TTOC SELECT(ERRORALTERNATE)
```

These tapes are marked as full. Consider your action for each tape in the list.

The following example shows the **LIST AGGREGATE** command to list aggregate backups:

```
LIST AGGREGATE(agname)
```

These backups are candidates for being moved offsite or being created in an offsite library.

Next, set your ACS routines to point to the new library and set all future DFSMShsm tape allocations to occur in the new library (if possible). You might need to look at the exceptions where you want a tape to be allocated outside of the library (specific media type that is going to an outside location). Ejects for specific purposes might also be needed, such as disaster recovery tapes or data that is intended for partners that can read these tapes.

Through the ACS routines, allocations can also be made to an offsite library for disaster recovery, if a library is available.

Also, look into your procedures, such as backup, dump, recycle, and aggregate backup, to verify that they also occur in the new setup.

7.7 Using Virtual Tape Servers for DFSMShsm

The most commonly used tape setup today for a DFSMShsm environment is based on *Virtual Tape Servers* (VTSs), the TS7700 Virtualization Engine, or the older B10 and B20 hardware. The VTS uses the 3494 or 3584 tape library for the back end. Virtual tape is transparent to physical tape allocation in relation to the DFSMShsm PARMLIB setup and the ACS routines.

Running a peer-to-peer virtual tape setup might require changes that relate to the duplexing of tapes. Consider whether to use peer-to-peer copy instead of a previous duplex copy that is based on DFSMShsm duplication. Also, if you use duplex copy for offsite recovery as part of your disaster recovery plan, consider whether to use the COPY EXPORT feature or whether the peer-to-peer copy meets your disaster recovery requirements.

Using virtual tape for DFSMShsm on the TS7700 is described. VTS (B10 and B20) is referenced only in exceptions.

7.7.1 Considerations for using VTS/TS7700 for DFSMShsm

DFSMShsm is a good design with physical tapes. The TS7700 does not have the same challenge as individual tape. The TS7700 adds to the capacity with actual data only and not to the logical capacity.

DFSMShsm benefits because of the potential of more logical tape drives (up to 256 in a single cluster grid and 512 in a 2-cluster grid). The DFSMShsm bandwidth can be increased by adding more drives to the DFSMShsm tasks and also by adding more DFSMShsm auxiliary tasks (MASH).

Another good reason to run DFSMShsm in a virtual environment is that the maintenance functions, such as AUDIT MEDIACONTROLS and TAPECOPY, can run more smoothly on smaller media in a virtual environment.

7.7.2 Larger logical volume sizes in TS7700

Consider the use of larger logical volume sizes for DFSMShsm, especially for backup tapes. These tapes will not be reread that often, compared to migration tapes that might be read occasionally. The *Recall Takeaway* risk is increased when you use larger volume sizes for the ML2 volumes. Several options are described. Consider which options are the best for your system.

The TS7700 supports logical volumes with the following maximum sizes:

- 400 MB
- 800 MB
- 1,000 MB
- 2,000 MB
- 4,000 MB
- 6,000 MB
- 25,000 MB

However, effective sizes can be larger if data is compressed. For example, if your data compresses with a 3:1 ratio, the effective maximum logical volume size for a 25,000 MB logical volume is 75,000 MB.

Depending on the logical volume sizes that you choose, the number of required volumes to store your data grows or shrinks, depending on the media size that you convert from. If your data sets fill native 3590 volumes, even with 25,000 MB logical volumes, you need more TS7700 logical volumes to store the data, which is stored as multi-volume data sets. You can specify 400 MB CST-emulated cartridges or 800 MB ECCST-emulated cartridges when you add volumes to the TS7700.

You can use these sizes directly or use policy management to override them to provide for the 1,000 MB, 2,000 MB, 4,000 MB, 6,000 MB, and 25,000 MB sizes. A logical volume size can be set by VOLSER and can change dynamically by using the DFSMS Data Class storage construct when a job requires a scratch volume. The amount of data that is copied to the stacked cartridge is only the amount of data that was written to a logical volume. The choice between all available logical volume sizes does not affect the real space that is used in either the TS7700 cache or the stacked volume. Unless you need CST emulation (400 MB), specify the ECCST media type when you insert volumes in the TS7700.

When you plan for the necessary number of logical volumes, first determine the number of private volumes that make up the current workload you plan to migrate. Look at the amount of data on your current volumes and then match that amount to the supported logical volume sizes. Match the volume sizes and include the compressibility of your data. If you do not know the average ratio, use the conservative value of 2:1.

DFSMShsm writes tapes as a single file by using connected sets. User data is written to a tape volume in 16 K logical block increments. A user data set can span a maximum of 254 volumes. So, considering the 254 maximum volume count, DFSMShsm will assign a new volume after 215 volumes to ensure that it is always able to write a user data set at the maximum size.

Considerations for larger volumes can be based on the limitation of data sets to span only 254 volumes. If you work with an 800 MB virtual tape, the maximum data set size, which is based on an expected compression factor of 2.5, is (254 x 800 x 2.5) = 508 GB. Because the largest logical volume size is 25,000 MB, the maximum data set on a data set that spans 254 DFSMShsm volumes is (254 x 25,000 x 2.5) = 15,875 GB.

Figure 7-12 on page 147 shows an example from the TS7700 graphical user interface (GUI) for managing constructs. In this example, the construct is data classes. Outboard Policy Management is used in the TS7700 to assign the policies.

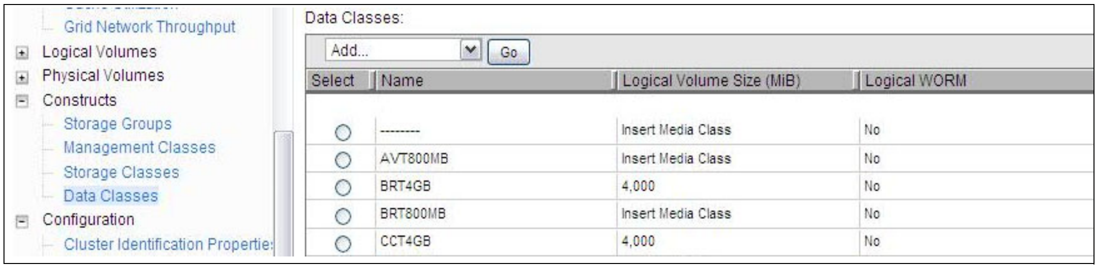

Figure 7-12 TS7700 GUI: Data class constructs

7.7.3 The TS7700 and awareness of tape capacity

The TS7700 is aware of the virtual volume sizes from code level 1.4 and higher. Therefore, the DFSMShsm **PERCENTFULL** parameter is less important and no longer required. In addition to the TS7700 required code level for this support, the host support is described in APAR OA24969.

APAR OA24969 states that virtual tape systems need to generally use a PERCENTFULL value of 97% unless a larger value is needed to account for virtual tapes that are larger than the nominal 400 MB standard capacity MEDIA1 or 800 MB enhanced capacity MEDIA2 tapes. In the case of the newer virtual tape systems (TS7700 R 1.4 and higher) where HSM derives media capacity by checking the mounted virtual tape, HSM allows a PERCENTFULL value up to 110%. Any value that is larger is reduced to 100%. For older virtual tape systems where HSM cannot dynamically determine virtual tape capacity, PERCENTFULL values that are larger than 110% are honored.

For a detailed description of the **SETSYS TAPEUTILIZATION** command, see *z/OS DFSMShsm Storage Administration,* SC35-0421.

The same is true for the **SETOAM TAPECAPACITY** command in OAM. With this support, the awareness of the tape capacity is in the hardware and the command is not needed.

If you change the usage of virtual volumes to larger virtual volume sizes, ensure that DFSMShsm uses those volumes. The following example shows a DFSMShsm command to identify tapes that are not full:

```
LIST TTOC SELECT(NOTFULL)
```

All of the tapes in this list must be marked full with these DFSMShsm commands, as shown in Example 7-11.

Example 7-11 DFSMShsm command for marking volumes full

```
DELVOL volser MIGRATION(MARKFULL)
DELVOL volser BACKUP(MARKFULL)
```

Consider using the DFSMShsm option **SETSYS PARTIALTAPE MARKFULL**. By using this option, DFSMShsm assigns a new tape at the next operation instead of retrieving the old tape into TS7700 cache and requiring a physical mount.

7.7.4 The TS7700 and the MOUNTWAITTIME setting

We recommend that you set the mount wait time value in DFSMShsm by using the **SETSYS MOUNTWAITTIME(12)** command. This value allows the virtual tape to be staged into cache in time. For the setting in the IECIOSxx member in SYS1.PARMLIB, consider the MIH value: MIH MOUNTMSG=YES, MNTS=10. Selecting this value ensures that mount pending messages appear. This value can be adjusted to the specific needs of your environment.

7.7.5 The TS7700 and scratch volumes

The use of volumes from the global scratch pool is recommended. Also, assign the Fast Ready category to this scratch pool to ensure fast mount time.

Example 7-12 shows example commands to select these DFSMShsm settings.

Example 7-12 Selecting DFSMShsm settings for scratch pools

```
SETSYS SELECTVOLUME(SCRATCH)
SETSYS TAPEDELETION(SCRATCHTAPE)
SETSYS PARTIALTAPE(MARKFULL)
```

7.7.6 The TS7700 and DFSMShsm DUMP parameters

When you use the TS7700 for the AUTODUMP function, avoid using the parameter settings that are shown in Example 7-13.

Example 7-13 AUTODUMP previous recommendation for physical tape usage

```
DEFINE DUMPCLASS(dclass STACK(nn))
BACKVOL SG(sgname ! VOLUMES(volser) DUMP(dclass STACK(10))
```

These settings were recommended previously for physical volume mounts to make these mounts more effective. In a virtual tape environment, specify `NOSTACK` to prevent unnecessary multivolume processing. This processing reduces parallelism during a restore.

7.7.7 The TS7700 physical limitations

The TS7700 excels at parallel mounts, especially writes. However, if your environment requires massive parallel recalls of data from the TS7700 back end and that data is no longer in TS7700 cache, consider your physical setup. Consider the number of available back-end drives for reading and writing data and potentially reclaiming data at the same time. These considerations apply to the number of parallel tasks in DFSMShsm that relate to backup, migration, recall, and recycle.

7.7.8 The TS7700 and tape management systems

No special actions are required for DFSMShsm in relation to a tape management system when you migrate to a TS7700 GRID.

For DFSMSrmm, you might need to change the VRSs and vault rules to reflect a virtual tape environment.

7.8 Preparing for a Virtual Tape Server or an ATL

The implementation of virtual tape and SMS-managed tape libraries involves several tasks. These tasks are fully described in the *DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Tape Libraries,* SC35-0427. To implement DFSMShsm functions in an SMS-managed tape environment, follow these steps:

- Determine the tape functions that you want to process in a VTS/tape library.
- Set up a global scratch pool or DFSMShsm-owned pool.
- Define or update a data class to compact tape data.
- Define or update a storage class to enable a storage group.
- Define or update a storage group to associate tape devices with the library.
- Set up or update ACS routines to filter data sets to the library.
- Prepare DFSMShsm for using tape (ARCCMDxx PARMLIB member updates).

7.8.1 Determine the functions to process in a VTS/tape library

VTS and tape libraries can process any DFSMShsm tape functions. You must decide which DFSMShsm functions to process in an automated tape system. Each DFSMShsm function uses a unique tape data set name. The ACS routine can recognize the functions that you want to process in an SMS-managed tape library by the data set names.

The assignment of classes in the ACS routines can be based on a data set naming convention (the DFSMShsm standard usage of data set names for backup and migration), as shown in Table 7-5.

Table 7-5 Overview of DFSMShsm data set naming conventions that relate to individual functions

DFDSMhsm function	Tape data set names	Commands with unit type restrictions
Back up to original	prefix.BACKTAPE.DATASET	SETSYS BACKUP(TAPE(unittype))
Back up to alternate	prefix.COPY.BACKTAPE.DATASET	
RECYCLE of backup tapes to ORIGINAL	prefix.BACKTAPE.DATASET	SETSYS RECYCLEOUTPUT(BACKUP(unittype))
RECYCLE of backup tapes to alternate	prefix.COPY.BACKTAPE.DATASET	
MIGRATION to original	prefix.HMIGTAPE.DATASET	SETSYS TAPEMIGRATION(-DIRECT(TAPE(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))
RYCLE of MIGRATION tapes to alternate	prefix.COPY.HMIGTAPE.DATASET	
DUMP	prefix.DMP.dclassVvolser.Dyyddd.Tssmmhh	DEFINE DUMPCLASS(class UNIT(unittype))
SPILL	prefix.BACKTAPE.DATASET	SETSYS SPILL(TAPE(unittype))

DFDSMhsm function	Tape data set names	Commands with unit type restrictions
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(BACKUPDEVICEC ATEGORY(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.Go0Vnnnn outputdatasetprefix.D.Go0Vnnnn outputdatasetprefix.U.Go0Vnnnn outputdatasetprefix.O.Go0Vnnnn	ABACKUP AGNAME UNIT(unittype)

7.9 Setting up a global scratch pool

The recommended setup for DFSMShsm scratch tape usage is by using a global scratch pool that is managed by a tape management product, such as DFRMM.

The assignment of a common scratch pool from a DFSMShsm perspective is based on three **`SETSYS`** parameters:

- The **`SELECTVOLUME`** parameter decides whether this mount is a specific or non-specific scratch mount. This parameter is valid for backup, migration, and dump processing.
- The **`PARTIALTAPE`** parameter specifies whether to mark migration or backup volumes full, and consequently if a mount occurs on a tape that is partially used or a new scratch tape.
- For the **`TAPEDELETION`** parameter, this setting determines whether to return the tape to a DFSMShsm-owned pool or global scratch pool after expiration.

Note: For DUPLEX tapes, writes always occur to a scratch tape.

Example 7-14 shows an example of setting up this global scratch pool.

Example 7-14 Example of setting up a global scratch pool for DFSMShsm

```
SETSYS SELECTVOLUME(SCRATCH)
SETSYS TAPEDELETION(SCRATCHTAPE)
SETSYS PARTIALTAPE(MARKFULL)
```

A *global scratch pool* is a pool of tapes that are not defined to DFSMShsm. A global scratch pool can be used by all tasks. When a tape is assigned by a task, it enters PRIVATE status until it is returned to the global scratch pool.

When DFSMShsm requests a scratch tape to be mounted, it creates a record in the offline control data set (OCDS). The record is stored by DFSMShsm in the OCDS during the records' use by DFSMShsm. Then, the record is deleted, and the tape returns to the global scratch pool. When a scratch tape from the global scratch pool is used by DFSMShsm, its status is changed to PRIVATE. Global scratch pools work well with tape management systems, such as DFSMSrmm.

7.10 Setting up a DFSMShsm-owned scratch pool

A specific scratch pool consists of tapes that are predefined to DFSMShsm (or another specific user) and can in this case be used only by DFSMShsm. These tapes are manually added to DFSMShsm with the **ADDVOL** command. For example, to add tape POK001 for use by DFSMShsm started procedure HSM1 as a backup tape in an ATL, issue the following command. This command adds a DFSMShsm-owned tape:

```
F HSM1, ADDVOL POK001 BACKUP UNIT(3590-1)
```

We recommend that you use global scratch pools. Output processing in a global scratch pool environment calls for any scratch cartridge that reduces the mount wait time, especially when you use devices with automatic cartridge loaders or an ATL.

If specific tasks and applications use their own pools, overhead is created and the number of available scratch tapes that are needed increases.

To return DFSMShsm-owned tapes to a global scratch pool, the empty tapes must be identified. (Use the **LIST TTOC SELECT(EMPTY)** command.) Next, these volumes must be deleted from DFSMShsm through the **DELVOL** command.

7.11 Preparing for SMS-managed tape

The tasks to implement SMS-managed tape are described.

7.11.1 Defining a storage class

A storage class needs to be assigned to DFSMS-managed tape allocations. No availability attributes are needed. In a virtual tape environment (VTS), improved cache management is provided through your installation's ACS routines to select a cache preference group of 0 or 1.

You can use the storage class initial access response time (**IART**) parameter at the host to select the preference group. If the value that is specified in this parameter is greater than or equal to 100, the logical volume is associated with cache preference group 0. If the value that is specified is less than 100, the logical volume is associated with cache preference group 1, which is also the default.

When space is needed in the cache, logical volumes that are associated with preference group 0 will be removed from the cache before logical volumes that are associated with preference group 1. Volumes are removed from preference group 0 based on their size, with the largest volumes removed first.

Volumes continue to be removed from preference group 1 based on a least recently used (LRU) algorithm. Data that is written to the VTS for backup or long-term archival can assign a storage class that specifies an initial access response time parameter that is greater than or equal to 100.

It makes sense to use preference groups and differ between the DFSMShsm creation of backup and migration tapes. Give DFSMShsm backup tapes a preference group of 0 because the probability of reusing these tapes is lower than for migration tapes.

Outboard policy management on the tape library with equally named storage classes ensures that the function is also enabled in the library. Any value in the library setting will override the storage class setting in DFSMS.

Figure 7-13 and Figure 7-14 on page 153 show a display of a storage class that is used for tape. Two panels are involved and no definitions were added except for defaults.

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

 To perform Storage Class Operations, Specify:
   CDS Name  . . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                                (1 to 44 character data set name or 'Active' )
   Storage Class Name  . . SCLIB2    (For Storage Class List, fully or
                                     partially specified or * for all)
 Select one of the following options  :
   3  1. List         - Generate a list of Storage Classes
      2. Display      - Display a Storage Class
      3. Define       - Define a Storage Class
      4. Alter        - Alter a Storage Class
      5. Cache Display - Display Storage Classes/Cache Sets
      6. Lock Display  - Display Storage Classes/Lock Sets
 If List Option is chosen,
    Enter "/" to select option      Respecify View Criteria
                                    Respecify Sort Criteria
 If Cache Display is Chosen, Specify Cache Structure Name . .
 If Lock Display is Chosen, Specify Lock Structure Name . . .
 Use ENTER to Perform Selection; .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Figure 7-13 Storage class definition (Page 1 of 2)

```
Panel  Utilities  Scroll  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              STORAGE CLASS DEFINE            Page 1 of 2
 Command ===>

 SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
 Storage Class Name  : SCLIB3
 To DEFINE Storage Class, Specify:
   Description ==>
               ==>
   Performance Objectives
    Direct Millisecond Response . . . .             (1 to 999 or blank)
    Direct Bias . . . . . . . . . . . .             (R, W or blank)
    Sequential Millisecond Response . .             (1 to 999 or blank)
    Sequential Bias . . . . . . . . . .             (R, W or blank)
    Initial Access Response Seconds . .             (0 to 9999 or blank)
    Sustained Data Rate (MB/sec)  . . .             (0 to 999 or blank)
    OAM Sublevel  . . . . . . . . . . .             (1, 2 or blank)
   Availability . . . . . . . . . . . . N           (C, P ,S or N)
   Accessibility  . . . . . . . . . . . N           (C, P ,S or N)
    Backup  . . . . . . . . . . . . . .             (Y, N or Blank)
    Versioning  . . . . . . . . . . . .             (Y, N or Blank)
 Use ENTER to Perform Verification; Use DOWN Command to View next Page;
```

Figure 7-14   Storage class definition panel (Page 2 of 2)

7.11.2 Defining a data class

The data class construct can provide the tape devices and media type information for tape data sets. The following attributes can be specified in a data class construct:

- The type of media to use
- Whether the data is to be compacted
- Recording technology (18 track, 36 track, 128 track, or 256 track)

Different tape devices and different media types can coexist at your installation, so you might need to assign specific data classes for the selection of specific devices and tape media. To define a data class, go to the ISMF Primary Option menu and enter option 4.

Figure 7-15 on page 154 through Figure 7-18 on page 155 show the data class definition panels.

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

 To perform Data Class Operations, Specify:
   CDS Name . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                                (1 to 44 character data set name or 'Active' )
   Data Class Name  . . DCTAPE    (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; .  .  .  .  .  .  .  .
```

Figure 7-15 Data Class Application Selection panel in ISMF

```
Panel  Utilities  Scroll  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                               DATA CLASS DEFINE                 Page 1 of 5
 Command ===>

 SCDS Name . . . : SYS1.SMS.MHLRES3.SCDS
 Data Class Name : DCTAPE

 To DEFINE Data Class, Specify:
   Description ==> Data class for tape
               ==>
   Recfm . . . . . . . . . .             (any valid RECFM combination or blank)
   Lrecl . . . . . . . . . .             (1 to 32761 or blank)
   Override Space  . . . . . N           (Y or N)
   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)
   Retpd or Expdt  . . . . .             (0 to 93000, YYYY/MM/DD or blank)
   Volume Count  . . . . . . 1           (1 to 255 or blank)
   Add'l Volume Amount . . .             (P=Primary, S=Secondary or blank)
 Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

Figure 7-16 Data Class Define panel (Page 1 of 5)

```
Panel  Utilities  Scroll  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              DATA CLASS DEFINE                 Page 2 of 5
 Command ===>      D

 SCDS Name . . . : SYS1.SMS.MHLRES3.SCDS
 Data Class Name : DCTAPE

 To DEFINE Data Class, Specify:

   Data Set Name Type  . . . . .        (EXT, HFS, LIB, PDS, Large or blank)
     If Ext  . . . . . . . . . .        (P=Preferred, R=Required or blank)
     Extended Addressability . . N      (Y or N)
     Record Access Bias  . . . .        (S=System, U=User or blank)
   Space Constraint Relief . . . N      (Y or N)
     Reduce Space Up To (%)  . .        (0 to 99 or blank)
     Dynamic Volume Count  . . .        (1 to 59 or blank)
   Compaction  . . . . . . . . .        (Y, N, T, G or blank)
   Spanned / Nonspanned  . . . .        (S=Spanned, N=Nonspanned or blank)
   System Managed Buffering  . .             (1K to 2048M or blank)
   System Determined Blocksize   N      (Y or N)
   EATTR . . . . . . . . . . . .        (O=Opt, N=No or blank)
 Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
```

Figure 7-17   Data Class Define panel (Page 2 of 5)

```
Panel  Utilities  Scroll  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              DATA CLASS DEFINE                 Page 3 of 5
 Command ===>

 SCDS Name . . . : SYS1.SMS.MHLRES3.SCDS
 Data Class Name : DCTAPE

 To DEFINE Data Class, Specify:

   Media Interchange
     Media Type . . . . . . . 2    (1 to 13 or blank)
     Recording Technology . . 256  (18,36,128,256,384,E1,E2-E4,EE2-EE4 or  ')
     Performance Scaling  . .      (Y, N or blank)
     Performance Segmentation      (Y, N or blank)
   Block Size Limit  . . . .             (32760 to 2GB or blank)
   Recorg  . . . . . . . . .             (KS, ES, RR, LS or blank)
   Keylen  . . . . . . . . .             (0 to 255 or blank)
   Keyoff  . . . . . . . . .             (0 to 32760 or blank)
   CIsize Data . . . . . . .             (1 to 32768 or blank)
   % Freespace CI  . . . . .             (0 to 100 or blank)
               CA  . . . . .             (0 to 100 or blank)
 Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
```

Figure 7-18   Data Class Define panel (Page 3 of 5)

The Data Class Define panel fields that affect tape device and media selection are explained. *Compaction* specifies whether data compaction needs to be selected for data sets that are assigned to this data class. We recommend that you always set the compaction to YES, even if you plan to direct DFSMShsm output tapes to a VTS. Media Type specifies the tape cartridge media that is used for data sets that are associated with this data class. Capacity on the tape depends on the tape drive that is used (how many tracks are being written). The following media type values are valid:

- 1 For MEDIA1 (3490 standard, 400 MB physical; CST)
- 2 For MEDIA2 (3490 enhanced, 800 MB physical; ECCST)
- 3 For MEDIA3 (3590 standard, 10 GB physical; HPCT)
- 4 For MEDIA4 (3590 extended, 20 GB physical; EHPCT)
- 5 For MEDIA5 (3592 Enterprise, 300/500 GB (EFMT1/EFMT2)
- 6 For MEDIA6 (3592 WORM, 300/500 GB (EFMT1/EFMT2)
- 7 For MEDIA7 (3592 Economy, 60/100 GB (EFMT1 / EFMT2)
- 8 For MEDIA8 (3592 Economy WORM, 50/100 GB (EFMT1/EFMT2)
- 9 For MEDIA9 (3592 Extended, 700 GB (EFMT2 only)
- 10 For MEDIA 10 (3592 WORM, 700 GB (EFMT2 only)
- 11 For MEDIA 11 (3592 Enterprise, 4000 GB (EFMT4, EEFMT4)
- 12 For MEDIA 12 (3592 WORM, 4000 GB (EFMT4/EEFMT4), 10,000 GB EFMT5)
- 13 For MEDIA 13 (3592 Advanced Economy, 500 GB (EFMT4/EEFMT4), 2,000 GB EFMT5)
- Blank cartridge type not specified

Recording Technology specifies the number of tracks on tape cartridges that are used for data sets that are associated with this data class. Table 7-6 shows the number of tracks that are written on recent IBM technology.

Table 7-6 IBM tape drives and support recording technology

3502 drive	**EFMT1 (512 track)**	**EFMT2 (896 track)**	**EFMT3 (1,152 track)**	**EFMT4 (2,560 track)**	**EFMT5 (5,120 track)**
Model J1A	R/W	Not supported	Not supported	Not supported	Not supported
Model E05	R/W	R/W	Not supported	Nut supported	Not supported
Model E06	R	R/W	R/W	Not supported	Not supported
Model E07	R	R	R/W	R/W	Not supported
Model E08	Not supported	Not supported	Not supported	R/W	R/W

Figure 7-19 on page 157 shows the panel that is used for setting the key, if encryption is used.

```
                              DATA CLASS DEFINE                 Page 4 of 5
Command ===>

SCDS Name . . . : SYS1.SMS.MHLRES3.SCDS
Data Class Name : DCTAPE

To DEFINE Data Class, Specify:

  Encryption Management
    Key Label 1 . . .     (1 to 64 characters or blank)

    Key Label 2 . . .


  Encoding for Key Label 1  . . . . .           (L, H or blank)
  Encoding for Key Label 2  . . . . .           (L, H or blank)




Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
```

Figure 7-19 Data Class Define panel (Page 4 of 5)

Figure 7-20 shows the last panel in the Data Class Define application in ISMF. It is shown only for reference because it contains no attributes that point to tape.

```
                              DATA CLASS DEFINE                 Page 5 of 5
Command ===>

SCDS Name . . . : SYS1.SMS.MHLRES3.SCDS
Data Class Name : DCTAPE

To DEFINE Data Class, Specify:

  Shareoptions Xregion  . . .      (1 to 4 or blank)
               Xsystem  . . .      (3, 4 or blank)
  Reuse . . . . . . . . . . . N    (Y or N)
  Initial Load  . . . . . . . R    (S=Speed, R=Recovery or blank)
  BWO . . . . . . . . . . . .      (TC=TYPECICS, TI=TYPEIMS, NO or blank)
  Log . . . . . . . . . . . .      (N=NONE, U=UNDO, A=ALL or blank)
  Logstream Id  . . . . . . .
  FRlog . . . . . . . . . . .      (A=ALL, N=NONE, R=REDO, U=UNDO or blank)
  RLS CF Cache Value  . . . . A    (A=ALL, N=NONE, U=UPDATESONLY)
  RLS Above the 2-GB Bar  . . N    (Y or N)
  Extent Constraint Removal   N    (Y or N)
  CA Reclaim  . . . . . . . . Y    (Y or N)
Use ENTER to perform Verification; Use UP Command to View previous Panel;
```

Figure 7-20 Data Class Define panel (Page 5 of 5)

Defining storage groups

The *storage group* definition in DFSMS is designed to assign tape volumes to a specific group and a specific library.

On the Storage Group Application Select Enter Storage Group Type panel, enter `3` to create a tape storage group, as shown in Figure 7-21.

```
                     STORAGE GROUP APPLICATION SELECT ENTER STORAGE GROUP TYPE
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Storage Group Name    SGLIB3           (For Storage Group List, fully or
                                         partially specified or * for all)
  Storage Group Type    tape             (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                         OBJECT, OBJECT BACKUP, or TAPE)
Select one of the following options  :
  3  1. List    - Generate a list of Storage Groups
     2. Display - Display a Storage Group (POOL only)
     3. Define  - Define a Storage Group
     4. Alter   - Alter a Storage Group
     5. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
        Space Info in GB           Respecify Sort Criteria
Use ENTER to Perform Selection;
```

Figure 7-21 Storage group definition main panel

The new storage group is related to a library, as shown in Figure 7-22.

```
                         TAPE STORAGE GROUP DEFINE
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : SGLIB3

To DEFINE Storage Group, Specify:

  Description ==>
              ==>

  Library Names  (1 to 8 characters each):
    ===> SCLIB2    ===>           ===>           ===>
    ===>           ===>           ===>           ===>

  DEFINE    SMS Storage Group Status . . N  (Y or N)


Use ENTER to Perform Verification and Selection;
```

Figure 7-22 Storage group definition: Tape Storage Group Define panel

If you entered Y in the Define SMS Storage Group Status field (Figure 7-22 on page 158), you enter the information that is shown in Figure 7-23. You can set the storage group status on individual logical partitions (LPARs).

```
                        SMS STORAGE GROUP STATUS DEFINE         Page 1 of 2
Command ===>

SCDS Name  . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name : SGLIB3
Storage Group Type : TAPE
To DEFINE Storage Group System/                      ( Possible SMS SG Status
Sys Group Status, Specify:                            for each:
                                                      - Pool SG Type
 System/Sys      SMS SG   System/Sys      SMS SG        NOTCON, ENABLE, DISALL
 Group Name      Status   Group Name      Status        DISNEW, QUIALL, QUINEW
 ----------      ------   ----------      ------    - Tape SG Type
 SC63       ===> ENABLE   SC64       ===> ENABLE        NOTCON, ENABLE,
 SC65       ===> ENABLE   SC70       ===> ENABLE        DISALL, DISNEW
            ===>                     ===>           - Copy Pool Backup SG Type
            ===>                     ===>               NOTCON, ENABLE )
            ===>                     ===>           * SYS GROUP = sysplex
            ===>                     ===>             minus Systems in the
            ===>                     ===>             Sysplex explicitly
            ===>                     ===>             defined in the SCDS
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

Figure 7-23 SMS Storage Group Status Define panel

ACS routines and tape allocations

The ACS routines assign a data class to the tape allocations that match the requested device type and media. If the correct assignment fails, the allocation fails. Example 7-15 shows a data class ACS routine that assigns a 4 GB tape in a VTS to a DFHSM allocation request for a backup tape.

Example 7-15 Assignment of 4 GB DATACLAS for tape allocation

```
FILTLIST DCTAP4GB  INCLUDE('JHSMB.BACKTAPE.DATASET' ....
-----
WHEN (&UNIT = &VTS_UNITS && &DSN = &DCTAP4GB) DO
 SET &DATACLAS = 'DCTAP4GB'
 WRITE 'IGDST10I DATA CLASS SET TO DCTAP4GB - VTS'
 EXIT CODE(0)
END
```

The storage class assignment in the ACS routine is the point where the allocation is determined to be SMS-managed or non-SMS-managed and whether to go to DASD or tape.

In Example 7-16, the allocation is assigned to a tape storage class.

Example 7-16 Assignment of STORCLAS for tape allocation

```
FILTLIST  VTS_UNITS INCLUDE('VTS'...
-----
WHEN (&UNIT = &VTS_UNITS) DO
 SET &STORCLAS = 'SCTAPE01'
 WRITE 'IGDST10I STORAGE CLASS SET TO SCTAPE01 - VTS'
 EXIT CODE(0)
END
```

The storage group with the correct tape pool must be assigned. Based on the unit name request in this example, the allocation will target the storage group that is needed for this allocation, as shown in Example 7-17.

Example 7-17 Assignment of a storage group in the ACS routine

```
FILTLIST  VTS_UNITS INCLUDE('TAPEV','TS77XX')
------
WHEN (&UNIT = &VTS_UNITS) DO
 SET &STORGRP = 'SGXXXXX'
 WRITE 'IGDST10I STORAGE GROUP SET TO SGXXXXX - PEER-TO-PEER VTS'
 EXIT CODE(0)
END
```

The allocation of the tape drive and media is now ongoing. Because this tape allocation is a DFSMShsm backup tape, this allocation is for the lifetime of the task. When the task ends, the drive is deallocated.

7.12 Preparing DFSMShsm for using tape

When you use DFSMShsm, certain functions, such as backup, ABARS, dump, migration, and space management, must be activated. Certain details (for example, unit names) must also be specified. The options that are shown in the sample PARMLIB in Example 7-18 are the functions that specifically relate to the DFSMShsm use of TAPE.

Example 7-18 Example of how to prepare DFSMShsm for tape use

```
/***********************************************************************/
/* SETSYS COMMANDS IN THE ARCCMDXX PARMLIB MEMBER THAT DEFINE THE */
/* DFSMSHSM ENVIRONMENT FOR AN SMS-MANAGED TAPE LIBRARY. */
/***********************************************************************/
/*
SETSYS DUPLEX(BACKUP MIGRATION)
SETSYS SELECTVOLUME(SCRATCH)
SETSYS PARTIALTAPE(MARKFULL)
SETSYS TAPEDELETION(SCRATCHTAPE)
SETSYS BACKUP(TAPE)
SETSYS TAPEMIGRATION(ML2TAPE)
SETSYS TAPESECURITY(EXPIRATION)
SETSYS DEFINE DUMPCLASS(ATLHSM -
NORESET AUTOREUSE NODATASETRESTORE -
DISPOSITION('AUTOMATE LOCATION'))
SETSYS INPUTTAPEALLOCATION(NOWAIT)
```

```
SETSYS OUTPUTTAPEALLOCATION(NOWAIT)
SETSYS RECYCLETAPEALLOCATION(NOWAIT)
SETSYS TAPEFORMAT(SINGLEFILE)
SETSYS TAPEUTILIZATION(PERCENTFULL(97))
SETSYS TAPEUTILIZATION(LIBRARYBACKUP PERCENTFUL(97))
SETSYS TAPEUTILIZATION(LIBRARYMIGRATION PERCENTFUL(97))
SETSYS MOUNTWAITTIME(10)
SETSYS TAPEHARDWARECOMPACT
```

With the parameters set for tape, we are ready to begin. This environment will be a duplex environment that uses a global scratch pool. Whenever a function ends, tapes are marked full so that the next occurrence of this event will start by using new scratch tapes. When tapes expire, they will return to the global scratch pool (TAPEDELETION). Backup and ML2 writes occur on tape. TAPESECURITY is limited to DFSMShsm setting an expiration date on all tapes, leaving it to the tape management system to handle protection and expiration. A dump class is assigned. AUTOREUSE returns tapes to the scratch pool without requiring a **DELVOL** command.

7.12.1 Usage of DFSMShsm RECYCLE

RECYCLE is an essential function in DFSMShsm tape maintenance. The constant expiration of data leaves DFSMShsm backup and migration tapes with unused blocks. When tape utilization falls under a limit of, for example, 50%, consider the use of RECYCLE to move the remaining valid data to new tapes that are filled at a higher percentage. This action at the same time frees DFSMShsm tapes, which can be returned to the global scratch pool.

To prepare for RECYCLE, your DFSMShsm PARMLIB must be updated with a statement that is similar to the following statement:

`SETSYS RECYCLEOUTPUT(BACKUP(`*`unittype`*`) MIGRATION(`*`unittype`*`))`

Execute RECYCLE according to the needs of your environment, for example, once a week to multiple times a week. This decision also depends on an adequate window of time.

Typically, RECYCLE will be executed in a batch job that is triggered by your scheduling tool. Example 7-19 shows an example of a batch scheduled job.

Example 7-19 Example of a batch scheduled DFSMShsm RECYCLE

```
HSEND RELEASE RECYCLE
HSEND SETSYS MAXRECYCLETASKS(4)
HSENDCMD WAIT RECYCLE ML2   EXECUTE PERCENTVALID(50)
```

RECYCLE is also often used when you move your DFSMShsm tapes to a new tape platform, for example, from ATL-based tapes to virtual tapes in a TS7700. Example 7-20 shows sample commands.

Example 7-20 Using RECYCLE

```
HSEND SETSYS USERUNITTABLE(TS7740 TAPE 3590-1)
HSEND SETSYS RECYCLEOUTPUT(BACKUP(TS7740) MIGRATION(3590-1))
HSENDCMD WAIT RECYCLE BACKUP EXECUTE PERCENTVALID(50)
```

If you use an esoteric that is named `3590-1` in an ATL and you want to move to your TS7740, you can use the esoteric for the TS7740 to cover the unit addresses that are associated with the TS7740 environment. In Example 7-20 on page 161, a controlled RECYCLE from the old ATL that is based on the esoteric `3590-1` for BACKUP now is recycled into the TS7740.

7.12.2 DB2 and DFSMShsm interaction

DFSMShsm can automatically migrate and recall archive log data sets and image copy data sets. If IBM DB2® needs an archive log data set or an image copy data set that DFSMShsm migrated, a recall begins automatically and DB2 waits for the recall to complete before DB2 continues.

For processes that read more than one archive log data set, such as the **RECOVER** utility, DB2 anticipates a DFSMShsm recall of migrated archive log data sets. When a DB2 process finishes reading one data set, it can continue with the next data set without delay because the data set might already be recalled by DFSMShsm.

If you accept the default value of YES for the **RECALL DATABASE** parameter on the Operator Functions panel (DSNTIPO), DB2 also recalls migrated table spaces and index spaces. At data set open time, DB2 waits for DFSMShsm to perform the recall. You can specify the amount of time DB2 waits while the recall is performed with the **RECALL DELAY** parameter, which is also on panel DSNTIPO. If RECALL DELAY is set to `0`, DB2 does not wait, and the recall is performed asynchronously.

You can use SMS to archive DB2 subsystem data sets, including the DB2 catalog, DB2 directory, active logs, and work file databases (DSNDB07 in a non-data-sharing environment). However, before you start DB2, recall these data sets by using DFSMShsm. Alternatively, you can avoid migrating these data sets by assigning them to a management class that prevents migration.

If a volume has a STOGROUP specified, you must recall that volume only to volumes of the same device type as others in the STOGROUP. In addition, you must coordinate the DFSMShsm automatic purge period, the DB2 log retention period, and the MODIFY utility usage. Otherwise, the image copies or logs that you might need during a recovery might already be deleted by DFSMShsm due to a specification in the management class. No synchronization exists between DB2 and DFSMShsm for expiration. If a DB2 table space was expired by DFSMShsm, it still exists in the DB2 catalog.

With the **MIGRATE**, **HMIGRATE**, or **HRECALL** commands, which can specify specific data set names, you can move data sets from one disk device type to another disk device type within the same DB2 subsystem. Do not migrate the DB2 directory, DB2 catalog, and the work file database (DSNDB07). Do not migrate any data sets that are in use frequently, such as the bootstrap data set and the active log. With the **MIGRATE VOLUME** command, you can move an entire disk volume from one device type to another device type.

7.13 DFSMShsm tape functions

The ways that DFSMShsm creates multiple tape copies are described.

7.13.1 Disaster recovery considerations

DFSMShsm includes functions that support disaster recovery. By using the DUPLEX or TAPECOPY function, you can have your tape copies created either directly in a disaster recovery location, or you can have your tape copies brought to a disaster recovery location daily to protect you against a disaster. These DFSMShsm tape copies can be combined with other backups, dumps, image copies, or ABARS backups.

Today, another popular alternative is to use two-cluster grid tape environments, where the hardware ensures two immediate copies of all of your tapes or a part of these tapes based on your policy setting. Additionally, a two-cluster grid also has the COPY EXPORT function to create a physical tape for recovery in a remote site. Migrating to this type of hardware might reduce the need for the DFSMShsm tape copy functions.

The concept of disaster recovery is that copies of all of your critical data sets are stored at a location that is separate from your computer site and a computer system of approximately the same capabilities will be available to use to resume operations.

You must have both of the following items to provide disaster backup:

- A way to make copies of DFSMShsm-owned tape volumes and to identify the copied volumes to DFSMShsm.
- A way to cause DFSMShsm to use the copied volumes instead of the original volumes. To fulfill these needs, DFSMShsm provides the duplex tape option or the **TAPECOPY** command, and the **TAPEREPL** command.

Duplex tape and TAPECOPY comparison

You can use the DUPLEX or the TAPECOPY function in DFSMShsm for an additional copy of your DFSMShsm tapes.

The major difference between the two options is that the DUPLEX function creates two tapes concurrently. TAPECOPY runs asynchronously and can run in a window of time that is convenient for you, based on the availability of tape drives.

DFSMShsm duplexing

The DUPLEX tape option provides an alternative to TAPECOPY processing for backup and migration cartridge-type tapes. Two tapes are created concurrently with one tape that is designated the original, and the other tape is designated as the alternate. The intent is that the original tape is kept onsite, and the alternate tape can either be created in a remote tape library, or taken offsite. If you have enough tape drives, the duplex tape option is recommended over TAPECOPY.

The option to duplex backup, migration, or both is determined by the **SETSYS** parameters that are shown in Example 7-21.

Example 7-21 Activating the DUPLEX function in DFSMShsm

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

The first command activates duplexing on backup tapes. The next command activates duplexing for migration. The last command activates duplexing for both. One tape is designated as the original to stay onsite, and the other tape is the alternate to take offsite or be created in a remote location.

The **ERRORALTERNATE** keyword can be used with the **DUPLEX** command. This command has two options, as shown in Example 7-22.

Example 7-22 Activating the DUPLEX function with the ERRORALTERNATE keyword

```
SETSYS DUPLEX MIGRATION(Y ERRORALTERNATE(CONTINUE))
STSYS DUPLEX MIGRATION(Y ERRORALTERNATE(MARKFULL))
```

In the first example, the setting for ERRORALTERNATE is `CONTINUE`. If an error occurs when writing to the alternate tape, writing to the primary tape continues and a TAPECOPY is issued afterward. `CONTINUE` is also the default on the **ERRORALTERNATE** command.

When the `MARKFULL` setting is used on the **ERRORALTERNATE** keyword, the function with the failure is terminated, both tapes are marked full, and the function continues on a new set of tapes.

Alternate tapes with failures can be listed with the following command:

```
LIST TTOC SELECT(ERRORALTERNATE)
```

The pair of tapes maintains the original versus the alternate distinction. The original tape volume data set names are determined by the function that created the tape.

Example 7-23 DFSMShsm migration and backup tape data set names

```
prefix.HMIGTAPE.DATASET
prefix.BACKTAPE.DATASET
```

The alternate tape volume names use the following format, as shown in Example 7-24.

Example 7-24 DFSMShsm migration and backup tape data set names on DUPLEX tapes

```
prefix.COPY.HMIGTAPE.DATASET
prefix.COPY.BACKTAPE.DATASET
```

We recommend that you use the same format for the original and alternate copy. Because two copies are created concurrently, you will need at least two output devices for each backup or migration task that will run, including the RECYCLE function of DFSMShsm. If DUPLEX is in use for a function, each RECYCLE task needs three tape drives: one for input and two for output.

The process to select tapes occurs in this order:

1. A partial tape and its alternate are selected.
2. If no partial tapes with alternates are identified, an empty ADDVOL volume and a scratch volume as an alternate are selected.
3. If no empty ADDVOL volumes are available, two scratch volumes are selected.

TAPECOPY function in DFSMShsm

Use the **TAPECOPY** command to make copies of your single-file-format DFSMShsm-owned cartridge tapes. The copies of the original tape are known as *alternates* and the terminology is similar to the terminology that is used in tape duplexing.

The alternate volumes start as scratch volumes to prevent backup, dump, or ML2 tapes from being used as alternate volumes.

Use the **TAPECOPY** command to copy the categories of tapes that are shown in Example 7-25.

Example 7-25 Using TAPECOPY to copy single-file-format tapes

```
TAPECOPY BACKUP
TAPECOPY MIGRATIONLEVEL2
TAPECOPY ALL
```

These commands cause copies to be taken of full tapes that do not already have an alternate copy. You can make copies of all volumes that are categorized as either ML2, backup, or both, or you can make copies of individual tape volumes. You can issue one **TAPECOPY** command for each DFSMShsm host on the condition that the requests do not overlap.

For example, you can issue a **TAPECOPY ML2** command on one host and a **TAPECOPY BACKUP** command on another host if the following parameter is already specified in your ARCCMDxx PARMLIB member:

```
SETSYS PARTIALTAPE(MARKFULL)
```

This parameter is the **SETSYS** parameter for copying partially filled tapes.

Nothing else is needed because this command instructs DFSMShsm to mark partially filled tapes as full at the end of each task. If, instead of `MARKFULL`, you coded `REUSE` in your ARCCMDxx PARMLIB member, you must issue the **SETSYS PARTIALTAPE(MARKFULL)** command. After a few backup and migration cycles, the volumes will be marked as full. Alternatively, you can issue one of the commands that are shown in Example 7-26 to mark the volumes as full.

Example 7-26 Alternative command to mark volumes full

```
DELVOL volser BACKUP(MARKFULL)
DELVOL volser MIGRATION(MARKFULL)
```

The commands that are shown in Example 7-26 mark the volumes that are specified in *volser* as full. ML2 tape volumes are not marked as full at the end of data set migration. An exception exists for ML2 tapes, even in a MARKFULL environment. ML2 tapes are not marked as full after data set migration for a simple reason. If, for example, command migration caused one data set to migrate to tape, that tape is an ML2 tape for only one data set. This situation is not an optimum tape utilization, so ML2 tapes remain as partially filled until a **TAPECOPY** command is issued for that volume. **TAPECOPY**, in a MARKFULL environment, marks ML2 volumes as full and makes the copy.

TAPECOPY can also make copies of individual volumes, as shown in Example 7-27.

Example 7-27 Using TAPECOPY to make copies of individual volumes

```
TAPECOPY ORIGINALVOLUMES(ovol1,ovol2,ovoln)
TAPECOPY ORIGINALVOLUMES(ovol1,ovol2,ovoln) -
ALTERNATEVOLUMES(avol1,avol2,avoln)
```

This command causes a copy of the volume or volumes that you specify with the **ovol** parameters. If you use the first form of the command, a default volume serial number of PRIVAT is used for the alternate tape volumes. PRIVAT indicates to DFSMShsm that a scratch volume must be mounted for each alternate tape volume.

However, if you use the second form of the command, DFSMShsm uses the volumes that you specify with the **ALTERNATEVOLUMES** parameter to become the alternate volumes. If you use the **ALTERNATEVOLUMES** parameter, you must specify the same number of volumes with the **ALTERNATEVOLUMES** parameter as you specified with the **ORIGINALVOLUMES** parameter. A one-to-one correspondence exists between the original volumes and alternate volumes that you specify; that is, `ovol1` is copied to `avol1`; `ovol2` is copied to `avol2`; and so on.

The alternate volume serial is recorded in the tape table of contents (TTOC) record of the original tape. DFSMShsm records only the current copy of the volume in the TTOC record. An ARC0436I message is issued each time that you make a copy when a copy exists.

Issue the **LIST TTOC** command to get a list of those volumes:

```
LIST TTOC SELECT(NOALTERNATEVOLUME)
```

If you output the list to a data set, it can be used directly in the **TAPECOPY** command, by using the following command:

```
TAPECOPY INDATASET(volcopy.list.dsname)
```

In the previous command, *`volcopy.list.dsname`* is the name of the data set, including the list of volumes you want to create an alternate tape volume for. You can use an edited version of the output from the **LIST** command to generate this list of volumes.

If a request for TAPECOPY is issued for a specific backup tape and the tape is in use, the TAPECOPY operation fails. The system issues message ARC0425I, which indicates that the tape is in use.

The TAPECOPY process with either the **ALL** or **BACKUP** keyword copies full tapes only. If you want copies of partial backup tapes, use the following process:

- Issue one of the **HOLD** commands to stop backup for all hosts that are running backup. The following **HOLD** commands cause the data set backup tasks with mounted tapes to unmount the tapes at the end of the current data set:
  - HOLD BACKUP holds all backup functions.
  - HOLD BACKUP(DSCOMMAND) holds data set command backup only.
  - HOLD BACKUP(DSCOMMAND(TAPE)) holds data set command backup to tape only.
- A LIST TTOC SELECT(BACKUP NOTFULL) lists the tapes that are not full. This list also shows which tapes do not have an alternate tape.
- A DELVOL MARKFULL is used for those tapes that you do not want extended after the TAPECOPY is made.
- A TAPECOPY OVOL(*volser*) is used for those tape volsers that do not have an alternate volume that is indicated in the LIST output.
- When the copies are complete, the corresponding **RELEASE BACKUP** command issue option can be used (on each host that was held) to allow the backup to be *usable.*

For TAPECOPY command or DUPLEX option

If you are copying the contents of one tape to another tape with the **TAPECOPY** command or are using the concurrent creation option DUPLEX, you need to be aware of minor inconsistencies that can exist in the length of cartridge-type tapes.

Because the **TAPECOPY** command copies the entire contents of one tape to another tape, it is important that enough media is available to copy the entire source tape to its target. Therefore, when you are copying tapes with the **TAPECOPY** command, use the default options (the equivalent of specifying the **TAPEUTILIZATION** command with the **PERCENTFULL** option of 97%). DFSMShsm marks the end of the volume when tapes are 97% full.

When you use the duplex option, it is recommended that you use the value 97% to ensure that you can write the same amount of data to both tapes. During duplexing, the **NOLIMIT** parameter of **TAPEUTILIZATION** will be converted to the default of 97%. If you are not copying tapes with the **TAPECOPY** command and you are not creating two tapes by using the DUPLEX option, you can specify the **TAPEUTILIZATION** command with a **PERCENTFULL** option of 100%.

DFSMShsm TAPEREPL command

The **TAPEREPL** command provides a way of replacing an original ML2 volume or a backup volume with its alternate volume. It also is important during an actual disaster recovery or a disaster recovery test run because you can replace original volumes with their alternate volumes.

DFSMShsm provides a way, assuming that you created alternate tapes or duplexed alternate tapes, to use those alternates in place of the original tapes for recalls or recovers. The designation of "disaster alternate volumes" helps distinguish between alternate volumes that are initially present at the recovery site and the alternate volumes that were created later at the recovery site. You might need to use your alternate volumes on the recovery site in a real disaster or for a disaster recovery test. To recover the volumes, run the commands that are shown in Example 7-28.

Example 7-28 Sample commands to recover volumes

```
TAPEREPL ALL DISASTERALTERNATEVOLUMES
SETSYS DISASTERMODE
```

The **DISASTERALTERNATEVOLUMES** parameter causes each alternate volume to be flagged as a disaster alternate volume. Only base DFSMShsm offline control data set (OCDS) TTOC records of ML2 and backup tapes with alternate volumes are updated. The **SETSYS DISASTERMODE** command specifies that DFSMShsm established a tape substitution mode. While DFSMShsm is running in disaster mode, it checks the TTOC record of the original tape volume to see whether a disaster alternate volume exists. If so, request the use of it to return the data. When you finish the use of the disaster alternate volumes, you can reset the flag with the following command:

```
TAPEREPL ONLYDISASTERALTERNATES(RESET)
```

If you are recovering from a real disaster and you want to return to your original site, you need to convert all disaster alternate volumes to original tapes. To convert all disaster alternate volumes to original tapes, you must issue the following command:

```
TAPEREPL ONLYDISASTERALTERNATES
```

Using DFSMShsm alternate tapes with DFSMSrmm

Previously, we showed you how to create alternate tapes for DFSMShsm-owned tape volumes. The major use for these alternate tapes is at a remote site in a disaster. The following process applies for using alternate volumes as disaster alternate volumes:

- Have a copy of the DFSMShsm CDSs at the disaster site.
- Perform a **TAPEREPL** command that specifies the **DISASTERALTERNATE** parameter. This command flags each existing alternate tape as a disaster alternate.
- Place DFSMShsm in DISASTER mode. When in DISASTER mode, DFSMShsm dynamically checks before it mounts an input tape whether the needed data resides on a tape with a disaster alternate. If the needed data is on a tape with a disaster alternate, the disaster alternate is requested.

DFSMSrmm recognizes when DFSMShsm is opening tape data sets and tolerates the data set names that DFSMShsm uses on condition that the last 17 characters of the data set name match. When you perform a true replacement by using the **TAPEREPL** command without the **DISASTERALTERNATE** keyword, DFSMShsm uses the data set name that it uses for the original tapes.

Set up ABARS processing

The aggregate backup and recovery support (ABARS) function creates the following files on tape volumes, which can then be transported to the recovery site:

- Data file
- Control file
- Instruction/activity log file (optional)

The output files are physical sequential data sets. Each of these files must be written to a separate physical tape volume (or volumes because the files can extend over several volumes). The size of the data file can be large, which makes it a good candidate for 3590 cartridges. The control and instruction files tend to be smaller, so you might want to write them on smaller capacity media. However, with ABARS, you cannot specify the output unit individually for these files. The unit name that is specified in the **ABACKUP** command applies to all three files.

To allocate the aggregate backup and ABARS files to 3590 in 3490 emulation mode devices, issue the following command:

```
ABACKUP aggregatename EXECUTE UNIT(EMUL3490)
```

EMUL3490 is the esoteric unit name for our IBM 3590 in 3490 emulation mode. No additional definitions need to be included in DFSMSrmm. If volume pools are not already defined to DFSMSrmm, you must define a new volume pool by adding the following definition in the EGDRMMxx member of PARMLIB, as shown in Example 7-29.

Example 7-29 Defining a new volume pool

```
VPOOL PREFIX(M*O TYPE(SCRATCH) MEDIANAME(3590) -
DESCRIPTION('3590 SCRATCH POOL') RACF(N) EXPDTCHECK(N)
```

Then, use the **ADDRACK** and **ADDVOLUME** DFSMSrmm subcommands to define the new volumes in your system, as shown in Example 7-30.

Example 7-30 Using ADDRACK and ADDVOLUME to define new volumes

```
RMM ADDRACK M00000 COUNT(100)
RMM ADDVOLUME M00000 COUNT(1000 POOL(M*)-
STATUS(SCRATCH) INIT(Y) MEDIANAME(3590)
```

ABARS and DFRMM

ABARS performs data backup and recovery processes on a predefined set of data, which is called an *aggregate*. During backup processing, the data is packaged as a single entity in preparation for taking it off-site, which enables the recovery of individual applications in user-priority sequence. In most cases, the output of ABACKUP is directed to tape media because it is easily transported from one location to another. With this ease of movement comes the task of managing the tape contents, location, and security.

Many sites manage tapes with a tape management system (DFSMSrmm, for example) and DFSMShsm. A tape begins as a scratch tape. The tape is used by DFSMShsm to store data and is returned to the tape management system to be reused as a scratch tape. To implement this form of concurrent tape management, communications must be coordinated whenever you define the environment and data sets for the use of a tape management system. The following example shows how to create a FACILITY class profile for the ABARS user ID, which in our case is `abarsid`.

```
RDEFINE FACILITY STGADMIN.EDG.OWNER.abarsid
```

Next, give user ID `abarsid` access to this profile and additional DFSMSrmm profiles that must be already defined on a DFSMSrmm system, as shown in Example 7-31.

Example 7-31 Giving the abarsid ID access to profiles

```
PE STGADMIN.EDG.OWNER.abarsid CLASS(FACILITY) ID(abarsid) ACCESS(UPDATE)
PE STGADMIN.EDG.RELEASE CLASS(FACILITY) ID(abarsid) ACCESS(READ)
PE STGADMIN.EDG.MASTER CLASS(FACILITY) ID(abarsid) ACCESS(READ)
```

Setting up DFSMShsm to use WORM output tapes for ABACKUP

In an SMS tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select Write Once Read Many (WORM) tapes for ABACKUP processing. The output data set prefix that is specified in the aggregate group definition can be used by the ACS routines to select a WORM data class.

Set up the ACS routine and the output data set name to uniquely identify the ABARS output files that must go to WORM tape. In a non-SMS tape environment, the default allows tape pooling to determine whether ABARS data sets go to WORM or R/W media. Optionally, if the DEVSUP*xx* parameter, **`ENFORCE_DC_MEDIA=ALLMEDIATY`** or **`ENFORCE_DC_MEDIA=MEDIA5PLUS`**, is used, the data class must request the appropriate media type to be successfully mounted.

Reducing the number of partially full tapes

When an ML2 tape that is used as a migration or recycle target is needed as input for a recall or ABACKUP, the requesting function can take the tape away whenever migration or recycle completes processing its current data set. When recall or ABACKUP is finished with a partial tape, DFSMShsm leaves it available as a migration or recycle target again, regardless of the PARTIALTAPE setting.

During the initial selection for an ML2 tape for output from a migration or recycle task, DFSMShsm tries to select the partial tape with the highest percentage of written data. An empty tape is selected only if no partial tapes are available. Three options for the **`LIST TTOC`** command are related:

- **`SELECT(RECALLTAKEAWAY)`** lists only those ML2 tapes that are taken away by recall. (If this list is too extensive, tune your ML2 tape migration criteria.)
- **`SELECT(ASSOCIATED)`** lists only those ML2 partial tapes that are currently associated with migration or recycle tasks as targets on this host or any other in the HSMplex (two or more hosts that are running DFSMShsm that share a common migration control data set (MCDS), OCDS, backup control data set (BCDS), and journal).

- **SELECT(NOTASSOCIATED)** lists only those ML2 partial tapes that are not associated with migration or recycle tasks as targets. The **RECYCLE** command for ML2 can recycle several of these partial tapes. By using the **SETSYS ML2PARTIALSNOTASSOCIATEDGOAL** parameter, you can control the trade-off between a few underused ML2 tapes versus the time that is needed to recycle them. If the number of unassociated ML2 partial tapes in the HSMplex that meets the PERCENTVALID and SELECT criteria for recycle exceeds the number that is specified by the **SETSYS** parameter, recycle includes enough partial tapes to reduce the number of partial tapes to the specified number. Recycle selects those tapes to process in the order of least amount of valid data to most amount of valid data.

Tape encryption on the TS7740

The importance of data protection is increasingly apparent with news reports of security breaches, loss, theft of personal and financial information, and government regulation. Encrypting the stacked cartridges minimizes the risk of unauthorized data access without excessive security management burdens or subsystem performance issues. The encryption solution for tape virtualization consists of several components:

- The encryption key manager
- The TS1120 and TS1130 encryption-enabled tape drives
- The TS7740 Virtualization Engine

You still see quite a few virtual tape environments that are not encrypted because of the perception that tapes are not brought outside of the physical robotics or outside of the data center. This scenario does not apply to certain tapes, such as Copy Export tapes. You can encrypt select tapes only, such as in this exception. Always plan for a data center relocation project with decrypted tapes. Is it necessary to encrypt tapes before a data center relocation?

For DFSMShsm, consider the following types of tapes that might be carried outside of the data center (and consequently might require encryption):

- Dump tapes
- ABARS tapes
- Duplex copies
- Tape backups of CDSs

Encryption key manager

The TS7700 Virtualization Engine can use one of the following encryption key managers:

- Encryption Key Manager (EKM)
- IBM Security Key Lifecycle Manager (formerly Tivoli Key Lifecycle Manager (TKLM))

In this book, we use *key manager* for EKM, TKLM, or Security Key Lifecycle Manager. The key manager is the central point from which all encryption key information is managed and served to the various subsystems. The key manager server communicates with the TS7740 Virtualization Engine and tape libraries, control units, and open systems device drivers.

Prerequisites for tape encryption in the TS7740

The IBM TS1120 and TS1130 tape drives provide hardware that performs the cryptography function without reducing the data transfer rate.

The TS7740 provides the means to manage the use of encryption and the keys that are used on a storage pool basis. It also acts as a proxy between the tape drives and the key manager servers, by using redundant Ethernet to communicate with the key manager servers and Fibre Channel connections to communicate with the drives. Encryption must be enabled in each of the tape drives. Encryption on the TS7740 is controlled on a storage pool basis.

The storage group DFSMS construct that is specified for a logical tape volume determines the storage pool that is used for the primary and optional secondary copies in the TS7740 Virtualization Engine. The storage pools were originally created for management of physical media. They are enhanced to include encryption characteristics. Storage pool encryption parameters are configured through the TS7740 Virtualization Engine management interface under physical volume pools.

For encryption support, all drives that attach to the TS7740 Virtualization Engine must be encryption-capable and encryption must be enabled. If the TS7740 Virtualization Engine uses TS1120 Tape Drives, they must also be enabled to run in their native E05 format. The management of encryption is performed on a physical volume pool basis. Through the management interface, one or more of the 32 pools can be enabled for encryption. Each pool can be defined to use specific encryption keys or the default encryption keys that are defined at the key manager server:

- Specific encryption keys

  Each pool that is defined in the TS7740 Virtualization Engine can have its own unique encryption key. As part of enabling a pool for encryption, enter two key labels for the pool and an associated key mode. The two keys might or might not be the same. Two keys are required by the key manager servers during a key exchange with the drive. A key label can be up to 64 characters. Key labels do not have to be unique per pool: The management interface provides the capability to assign the same key label to multiple pools. For each key, a key mode can be specified. The supported key modes are *Label* and *Hash*. As part of the encryption configuration through the management interface, you provide IP addresses for a primary and an optional secondary key manager.

- Default encryption keys

  The TS7740 Virtualization Engine encryption supports the use of a default key. This support simplifies the management of the encryption infrastructure because no future changes are required at the TS7740 Virtualization Engine. After a pool is defined to use the default key, the management of encryption parameters is performed at the key manager:

  – Creation and management of encryption certificates
  – Device authorization for key manager services
  – Global default key definitions
  – Drive-level default key definitions
  – Default key changes as required by security policies

Encryption processing

For logical volumes that contain data to encrypt, host applications direct them to a specific pool that is enabled for encryption by using the storage group construct name. All data that is directed to a pool that is enabled for encryption will be encrypted when it is pre-migrated to the physical stacked volumes or reclaimed to the stacked volume during the reclamation process. The storage group construct name is bound to a logical volume when it is mounted as a scratch volume. Through the management interface, the storage group name is associated with a specific pool number.

When the data for a logical volume is copied from the tape volume cache to a physical volume in an encryption-enabled pool, the TS7740 Virtualization Engine determines whether a new physical volume needs to be mounted. If a new cartridge is required, the TS7740 Virtualization Engine directs the drive to use encryption during the mount process. The TS7740 Virtualization Engine also provides the drive with the key labels that are specified for that pool.

When the first write data is received by the drive, a connection is made to a key manager and the necessary key to encrypt is obtained. Physical scratch volumes are encrypted with the keys in effect at the time of first write to beginning of tape (BOT). Any partially filled physical volumes continue to use the encryption settings in effect at the time the tape was initially written from BOT. The encryption settings are static until the volumes are reclaimed and rewritten again from BOT.

TS7740 communication with key manager

Communicating with the key manager is through the same Ethernet interface that is used to connect the TS7740 Virtualization Engine to your network for access to the management interface. The request for an encryption key is directed to the IP address of the primary key manager. Responses are passed through the TS7740 Virtualization Engine to the drive. If the primary key manager did not respond to the key management request, the optional secondary key manager IP address is used.

After the TS1120 or TS1130 drive completes the key management communication with the key manager, it accepts data from the tape volume cache. When a logical volume needs to be read from a physical volume in a pool that is enabled for encryption, either as part of a recall or reclamation operation, the TS7740 Virtualization Engine uses the key manager to obtain the necessary information to decrypt the data. The affinity of the logical volume to a specific encryption key or the default key can also be used as part of the search criteria through the TS7700 Virtualization Engine management interface.

DFSMShsm considerations for the TS1130 tape drive

DFSMShsm, a z/OS functional component, automatically manages low activity and inactive data in both system-managed and non-system-managed environments. DFSMShsm also provides automatic backup and recovery of active data in those environments. DFSMShsm can use the TS1130 tape drive (3592 Model E06, 3592-3E) for all functions. DFSMShsm normally uses non-WORM media (MEDIA5, MEDIA7, and MEDIA9) for non-ABARS functions. DFSMShsm uses all media, including WORM (MEDIA6, MEDIA8, and MEDIA10) for ABARS processing. DFSMShsm can use the WORM media for non-ABARs processing if the WORM media is specifically allowed for non-ABARs processing by your installation.

Modifying your dump classes

The method for requesting encryption depends on whether you plan to use hardware encryption or host-based encryption:

- To request hardware encryption for a dump class, specify it in the SMS data class for the dump data.
- To request host-based encryption for a dump class, use the DFSMShsm **DEFINE DUMPCLASS(ENCRYPT)** command. With **ENCRYPT**, include the **RSA** or **KEYPASSWORD** subparameters to specify the type of host-based encryption. **ENCRYPT(NONE)** specifies host-based encryption will not occur.

If your dump classes are currently defined to use host-based encryption (and possibly host-based compression before encryption), it is recommended that you remove the host-based encryption requests from any dump classes for which you plan to use tape hardware encryption.

During the process of migrating your dump classes to use hardware encryption, certain dump classes might still be defined to use host-based encryption, while their associated SMS data classes are defined to use tape hardware encryption. Here, DFSMSdss ignores requests for host-based encryption for these tape volumes and, instead, uses hardware encryption.

With this processing, you can complete the migration to hardware encryption without needing to modify your dump-requesting jobs. However, removing host-based encryption requests from a dump class when tape hardware encryption is also requested can avoid confusion about which process is active.

> **Tips:** To determine whether hardware encryption or host-based encryption was used for a particular tape volume, check the associated dump volume record (DVL).
>
> If more than one dump class is specified (creating more than one dump copy), those dump classes specify host-based encryption. If each dump class has a unique data class assigned, and certain, but not all of the associated data classes request tape hardware encryption, all dump copies will fail. Tape hardware encryption can override host-based encryption for all dump classes that are associated with a source volume or none of the dump classes, but it cannot override a subset of those dump classes.

Reducing recall processing time

In a multihost environment, DFSMShsm can experience delays during recall processing when the **SETSYS TAPERECALLLIMITS** command is not specified. This optional parameter specifies how long a single tape can remain mounted before DFSMShsm checks to see whether higher priority recall requests are waiting to process from the same system or another host. The syntax of the **TAPERECALLLIMITS** command is shown:

```
SETSYS TAPERECALLLIMITS(TASK(time) TAPE(time))
```

If this command is not specified, DFSMShsm uses the default values for this parameter, as shown:

```
SETSYS TAPERECALLLIMITS(TASK(15) TAPE(20))
```

The optional parameters of the **TAPERECALLLIMITS** command are explained in detail:

TASK This parameter specifies the number of minutes that a recall task is active on a single tape mount before DFSMShsm checks for any higher priority recalls on that host. The maximum number of tape recall tasks must be active. If a higher priority task is identified, DFSMShsm completes its current recall task, dismounts the tape, and processes the new higher priority task.

TAPE This parameter specifies the number of minutes that a recall task is active on a single tape mount before additional multihost criteria are considered. If a higher priority task is identified on another DFSMShsm host, DFSMShsm completes the current recall task and dismounts the tape for use on the other host.

After this action is performed, DFSMShsm excludes any recall request from that tape to allow the host system time to try its delayed requests again and catch up.

So in the default setting of TASK(15), DFSMShsm processes recalls from a single tape mount for 15 minutes before it checks for higher priority tasks on that system, which it will honor if it identifies any.

In the default setting of TAPE(20), DFSMShsm processes recalls from a single tape mount for 20 minutes before it checks for higher priority tasks on another host, which it will honor if it identifies any.

The following message indicates that TAPERECALLLIMITS is in effect:

```
ARC0312I RECALLS FROM TAPE volser TERMINATED FOR USAGE BY HOST h
```

In this message, HOST *h* identifies the host with the highest priority recall request.

When the recall processing stops for a specific recall task on a DFSMShsm host, all queued recall requests on that host that require the dismounted tape volume are excluded from consideration for 5 minutes. This period allows other hosts an opportunity to try their delayed requests again for the dismounted tape volume.

When DFSMShsm processes multiple recall requests that use one tape mount, the recall task processes the highest priority tape recall request that requires the currently mounted tape. If DFSMShsm forces a dismount of the tape volume, the task and the tape are freed to process other higher priority recalls.

The **TAPERECALLLIMITS** parameter means the same when it is applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

> **Notes:** TAPE(*time*) is longer than TASK(*time*) because TAPE(*time*) requires delaying additional queued requests.
>
> The storage administrator can disable the TAPERECALLLIMIT function by setting both the TASK time and TAPE time to large values, such as `1000` or even `65535` minutes.

Optimizing tape usage

In an average installation, 50% of tape data sets contain inactive data (data that is written once and never read again). Most inactive data sets are point-in-time backups, which are application-initiated data set backups that are used only if an application or system failure occurs. Although data set backup is critical, application-initiated backups to tape use the tape media and subsystem poorly and require many costly operator tape mounts. The remaining tape data sets, active data, cause the same inefficiencies in tape cartridge use. However, they are usually not active; 60% - 90% of all tape data sets have all accesses on the same calendar date.

You can use system-managed storage facilities to optimize management of both inactive and active tape data sets. Tape mount management helps you understand your tape workload and use advanced tape hardware and DFSMS facilities to provide the following benefits:

- Reduce tape mounts. Data sets that are good candidates to tape mount management are written to a system-managed DASD buffer instead of a tape. They are then written to tape by DFSMShsm with other data sets, and they are automatically retrieved by DFSMShsm if they are accessed.
- Reduce tape library inventory and maximize media usage. Tape mount management candidates are written by DFSMShsm to tape in single-file, compacted form. DFSMShsm attempts to fill the entire tape volume before it uses another volume.
- Improve turnaround time for batch jobs, depending on tape data sets. Most tape processing jobs are queued on tape drives, not mount time or tape I/O because, when the drive is finally allocated, little data is written to the cartridge. Batch jobs that use data sets that are written to the system-managed DASD buffer do not wait for tape mounts, and can perform I/O at DASD or cache speed.

Tapespansize statement

In this statement, you specify the maximum number of megabytes of tape that DFSMShsm can leave unused on tape while it tries to eliminate spanning data sets. When space is insufficient to contain the entire next data set on the current tape without exceeding the requested maximum usage, the next data set begins on an empty tape. The default value for this parameter is 500 MB, but IBM recommends 4000 MB for 3490 and 3590.

Using a TAPESPANSIZE other than the default size is a trade-off between the use of DFSMShsm tapes and needing to mount tapes more frequently to read spanned data sets.

The following example shows how to set the TAPESPANSIZE on tapes to a recommended value for 3490 and 3590:

```
SETSYS TAPESPANSIZE(4000)
```

Note: If the **SETSYS TAPEUTILIZATION** parameter is set to NOLIMIT, DFSMShsm makes no attempt to reduce data set tape volume spanning. If the **SETSYS TAPEUTILIZATION** parameter is set to a percent, DFSMShsm performs reduced data set spanning.

7.14 Tape mount management

Tape mount management provides bandwidth in the batch flow. Tape mount management also absorbs the smaller suitable tape allocations on DASD instead of mounting physical tapes and filling these tapes to a limited extent.

Today, virtual tape systems absorb the partially filled tapes. Virtual tape systems reduce the need for tape mount management because the bandwidth is available through many virtual drives (up to 512 in one tape environment).

You might still want to implement tape mount management because you still use physical tape mounts for your customer and these tapes are only partially filled, or the number of drives is limited. If you implement tape mount management, you might then want to offload your tape environment for physical mounts with partially filled tapes (that are not needed for disaster recovery).

Follow these steps to implement tape mount management:

1. Tape data sets are categorized according to size, pattern of usage, and other criteria, so that appropriate SMS policies can be assigned to the tape mount management candidates.
2. Data sets that are written to tape are intercepted at allocation and, if eligible for tape mount management, redirected to a system-managed DASD pool. The pool serves as a staging area for these data sets until they are migrated to tape by DFSMShsm. The location of the data is transparent to the application program.
3. During interval migration that runs by the hour, DFSMShsm checks the occupancy of the DASD buffer storage group to ensure that space is available when needed, migrating data sets to a lower level of the storage hierarchy when they are no longer required on primary DASD volumes.
4. DFSMShsm eventually moves the data to tape by using single-file format and data compaction to create full tape cartridges.
5. If an application later requests a data set, DFSMShsm automatically recalls it from where it resides in the storage hierarchy and allocates it on primary DASD for access. This process can significantly reduce tape mounts and the number of cartridges that are required to store the data.

Tape mount management implementation plan

Implementing tape mount management requires careful planning. Include the following major tasks in your implementation plan:

1. Analyze your current tape environment.
2. Simulate the proposed tape mount management environment.
3. Select DASD volumes to satisfy buffer requirements.
4. Consider and define SMS constructs for the candidate data sets.
5. Create the ACS routines, or update the current ACS routines.
6. Monitor the result and expect to need to tune DFSMShsm operations.

Volume mount analyzer

The *volume mount analyzer* is a program that helps you analyze your current tape environment. You use the volume mount analyzer to study tape mount activity, monitor tape media use, and implement tape mount management at your installation. The volume mount analyzer produces reports that you can use to tailor data classes, management classes, and ACS routine filters, depending on your specific tape usage profile. By using tape mount management, you can maximize the use of your tape media and reduce your tape mounts. The volume mount analyzer consists of two programs: GFTAXTR and GFTAVMA.

By analyzing the SMF records of your installation, the volume mount analyzer can help you determine the following information:

- The space that each tape data set uses on a tape volume
- The unused space on a tape volume
- The number of tape mounts you will save by implementing tape mount management
- The tape mount management DASD buffer space that you will need to hold the data sets until they are transferred to tape storage

Using the volume mount analyzer for tape mount management

The volume mount analyzer helps you analyze the costs and savings of automating your tape data management by using DFSMS. It also helps you choose data sets for tape mount management. Tape mount management is a method of reducing tape mounts by performing the following functions:

- Allowing the system to manage the placement of data
- Taking advantage of hardware and software compaction
- Taking advantage of new tape technology
- Filling each tape volume to capacity automatically

Using DFSMS and tape mount management can help you reduce the number of both tape mounts and tape volumes that your installation requires. The volume mount analyzer reviews your tape mounts and creates reports that provide you with information you need to effectively implement the tape mount management methodology that we suggest.

Information that is obtained by volume mount analyzer

The volume mount analyzer produces both summary and detailed reports of your tape activity for a selected time period. You can use these reports with an estimate of the average cost of a tape mount to determine the value of managing tape data with DFSMS or to write ACS routines that implement tape mount management most effectively.

Related to DFSMShsm, DFSMS class analysis is used to see how data uses the different DFSMShsm tiers. This understanding can be used to tune the environment.

The DFSMShsm usage of tape can be isolated and analyzed. The tape workload from DFSMShsm can then be tuned and adjusted, if needed.

7.15 DFSMShsm and 3592 hardware support

DFSMShsm and 3592 hardware support are described.

7.15.1 DFSMShsm modifications for 3592 Model J

Specific changes to DFSMShsm support for the new 3592 Model E05 tape drives are described.

DFSMShsm can use 3592 Model E05 for all functions. The 3592 Model E05 uses existing cartridge media (MEDIA5, MEDIA6, MEDIA7, and MEDIA8) and also supports two new media types: MEDIA9 (700 GB R/W) and MEDIA10 (700 GB WORM). MEDIA9 is available for all DFSMShsm functions. MEDIA10 is intended specifically for ABARS processing of ABACKUP tapes.

DFSMShsm also adds support for enterprise format 2 (EFMT2), a new recording format that is required for using the new MEDIA9 and MEDIA10 cartridge media. The changes are listed:

- Planning and installation
- Input tape utilization
- Output tape selection
- Output tape utilization
- Reuse capacity table
- EFMT2 formatted volume display
- WORM tape cartridge rejection at OPEN time
- ABARS with WORM tape cartridge

7.15.2 Planning and installation

In an HSMplex, you need to apply the z/OS support program temporary fixes (PTFs) without enabling the PTFs to coexisting systems without full support for 3592-E05. This way, a coexisting system can reject partial tapes that are written by the 3592-E05 with EFMT2 technology.

7.15.3 Input tape selection

You can use MEDIA5, MEDIA7, and MEDIA9 tapes as input for all DFSMShsm functions. You can also use MEDIA6, MEDIA8, and MEDIA10 tapes for ABARS processing. MEDIA5, MEDIA6, MEDIA7, and MEDIA8 tapes can be written in either of two recording formats: EFMT1 or EFMT2. The MEDIA9 and MEDIA10 tapes must be written in EFMT2 format.

7.15.4 Output tape selection

DFSMShsm selects 3592 Model J tape drives for output in SMS and non-SMS tape environments. DFSMShsm performs all of its allocation requests by using these standard dynamic allocation interfaces:

- Non-SMS-managed output tape selection: If multiple types of tape drives are installed that emulate the 3590 device type, you must define an esoteric name for each model that DFSMShsm uses. You must then define the esoteric names to DFSMShsm by using the **SETSYS USERUNITTABLE(esoteric1:esoteric1,esoteric2:esoteric2,...)** command. You must also specify the esoteric names as the unit names for the DFSMShsm functions that you want. If a single type of tape drive is installed that emulates the 3590 device type, you do not need to define an esoteric name; instead, you can specify the 3590-1 generic name for the DFSMShsm functions that you want.
- SMS-managed output tape selection: DFSMShsm performs a non-specific allocation; it then finds an acceptable output tape for the already allocated drive. If the 3590-1 generic name is used with multiple types of tape drives that are emulating the 3590 device type, see APAR OW57282 for further instructions. For output tape utilization, DFSMShsm writes to 97% of the capacity of MEDIA5, MEDIA6, MEDIA7, and MEDIA8 tapes unless the percentage is otherwise specified by the installation. You can specify other percentages by using the **SETSYS TAPEUTILIZATION** command, depending on the needs of the installation. DFSMShsm uses the reported cartridge type on the physical device to determine the tape's capacity.

7.15.5 Output tape utilization

DFSMShsm writes to 97% of the capacity of MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10 tapes unless the percentage is otherwise specified by the installation. Other percentages can be specified through the **SETSYS TAPEUTILIZATION** command, depending on the needs of the installation. DFSMShsm uses the reported cartridge type on the physical device to determine the tape's capacity.

For scratch tapes, the 3592-E05 can use empty MEDIA5, MEDIA7, or MEDIA9 tapes for all DFSMShsm functions and also use empty MEDIA6, MEDIA8, and MEDIA10 tapes for ABARS processing. Because the 3592-E05 can write in either of two recording formats (EFMT1 or EFMT2) to MEDIA5, MEDIA6, MEDIA7, and MEDIA8 tapes, you must modify your installation's ACS routines to select the recording format to use on blank media (through the data class that is assigned to the tape) if you want the 3592 Model E05 drives to use EFMT1.

For duplexed tapes, ensure that the data class selects the same media type and recording technology for the original and the alternate copy. Otherwise, the process can fail when the duplex tape is mounted for output, or when you use the alternate copy after a tape replace. If different media or machine types are needed for the original and alternate tapes, see APARs OW52309, OA04821, and OA11603 for more information.

7.15.6 Partial tapes

When the 3592-E05 uses a MEDIA5, MEDIA6, MEDIA7, or MEDIA8 partial tape for output, the tape can be written in either EFMT1 or EFMT2 recording technology. In contrast, MEDIA9 and 10 *partial tapes* are always recorded by using EFMT2 technology. DFSMShsm automatically determines the format in which a tape is written and extends the tape in the same format.

7.15.7 Reuse capacity table

The reuse capacity table supports MEDIA5 and MEDIA7 tape cartridges for 3592 Model J tape drives. DFSMShsm uses the reuse capacity table to determine the tapes that are eligible for RECYCLE processing based on capacity for each media type. Each media type has separate reuse capacities for backup and migration.

7.15.8 Worm tape cartridge rejection

DFSMShsm examines the OEVSWORM field that is passed by OPEN/EOV processing. The bit is turned on by OPEN/EOV when a WORM tape cartridge is mounted. If a WORM tape cartridge is mounted for any function other than ABACKUP or ARECOVER, DFSMShsm returns to OPEN/EOV with RC08, resulting in ABEND913 RC34 and a failure of OPEN. DFSMShsm fails the function and issues either message ARC0309I (for non-ABARs) or ARC6410E (for an ARECOVER ML2 tape).

7.16 DFSMShsm use of performance scaling on 3592

If your installation requires fast access to data on a MEDIA5 or MEDIA9 tape, consider using the 3592 performance scaling feature. In DFSMShsm, performance scaling applies in both tape libraries and stand-alone environments. Performance scaling uses 20% of the physical space on each tape and keeps data sets closer together and closer to the initial load point. Performance scaling permits the same amount of data to exist on a larger number of tapes, allowing more input tasks to run concurrently.

With performance scaling, you can effectively increase the "bandwidth" of operations that read in data from tape. In contrast, performance segmentation allows the use of most of the physical media, while enhancing performance for the first and last portions of the tape.

7.16.1 How to use performance scaling and segmentation for 3592

The 3592 Model E05 supports performance scaling and performance segmentation of media tape cartridges. With these functions, you can optimize performance for MEDIA5 and MEDIA9 cartridges. *A cartridge can be defined for performance scaling or performance segmentation, but not both.*

Performance scaling, also known as *capacity scaling*, is a function to contain data in a specified fraction of the tape, yielding faster locate and read times. Performance scaling for the 3592 Model E05 limits the data that is written to the first 20% (the optimally scaled performance capacity) of the cartridge. To select performance scaling for a cartridge, follow these steps:

1. Define a data class that requests performance scaling.
2. Modify or create ACS routines to associate the tape output functions to use performance scaling with a data class that requests performance scaling (set to `Y`).

   The 3592 Model E05 tape drive also divides the tape into longitudinal segments. By using this capability, you can segment the tape into two segments: one segment is a fast access segment to be filled first, and the other segment is additional capacity to be filled after. Performance scaling applies to MEDIA5 only.

If you decide to use the performance segmentation attribute, follow these steps:

1. Define a data class that requests performance segmentation (set to `Y`).
2. Modify or create ACS routines to associate the tape output functions to use performance segmentation with a data class that requests performance segmentation.
3. Performance segmentation can also be used with the 3592 Model J1A with MEDIA5.

> **Note:** Performance scaling and segmentation cannot occur on the TS7700 back-end drives and media.

7.16.2 Activating performance segmentation or scaling

A data class must exist with the performance segmentation or performance scaling attribute set to `Y`:

- The ACS routine must map a DFSMShsm single file tape data set name to the data class.
- Because the data class now determines whether a performance option is used, non-SMS tape needs ACS routines if you want to use a 3592 performance option.

The IEC205I at the close of tape will indicate whether performance segmentation or scaling was used. Performance segmentation and performance scaling apply only to MEDIA5.

> **Note:** Performance segmentation and performance scaling are mutually exclusive.

7.16.3 List of SETSYS commands that are used for tape

The following **`SETSYS`** commands are used for tape:

- `TAPEOUTPUTPROMPT`
- `TAPESPANSIZE`
- `PARTIALTAPE`
- `PERCENTFULL`
- `TAPEHARDWARECOMPACT`
- `SETSYS INPUTTAPEALLOCATION(NOWAIT)`
- `SETSYS TAPEMRIGRATION`
- `TAPEUTILIZATION`
- `USERUNITTABLE`
- `TAPERECALLLIMITS`
- `SETSYS TAPEMAXRECALLTASKS(tasks)`
- `SETSYS ARECOVERUNITNAME`
- `UNITNAME`
- `MIGUNITNAME`
- `TAPEMIGRATION`
- `BACKUP/DSBACKUP/BACKDS`
- `RECYCLEOUTPUT`

7.16.4 Commands that address the tape management system

The following commands address the tape management system:

- `SETSYS CDSVERSIONBACKUP`
- `BACKUPDEVICECATEGORY(TAPE)`
- `SETSYS SELECTVOLUME(SCRATCH)`
- `SETSYS TAPEDELETION(SCRATCHTAPE)`
- `SETSYS TAPESECURITY(RACF)`
- `SETSYS EXITON(ARCTVEXT)`

8

HSMplexes, GRSplexes, and sysplexes

In this chapter, we describe the concepts of and relationships among hierarchical storage management complexes (HSMplexes), global resource serialization complexes (GRSplexes), and sysplexes.

8.1 Introduction to the HSMplex

So far, the information in this book focused on the implementation of a single HSM environment. Depending on the number of systems in your environment that share data and the amount of data that you want DFSMShsm to manage, you might want to implement an HSMplex.

8.1.1 Sample HSMplex

An *HSMplex* is two or more instances of DFSMShsm that share the same set of control data sets (CDSs) and journal. These instances of DFSMShsm can be installed on the same z/OS image or logical partition (LPAR), on multiple LPARs, or both. Multiple instances of DFSMShsm that are installed on the same LPAR are called a *multi-address space DFSMShsm* (MASH) LPAR.

An HSMplex can contain up to 39 instances of DFSMShsm, which is the number of unique host IDs that exist within DFSMShsm. For a visual representation of the HSMplex concept, see Figure 8-1, which shows an example HSMplex.

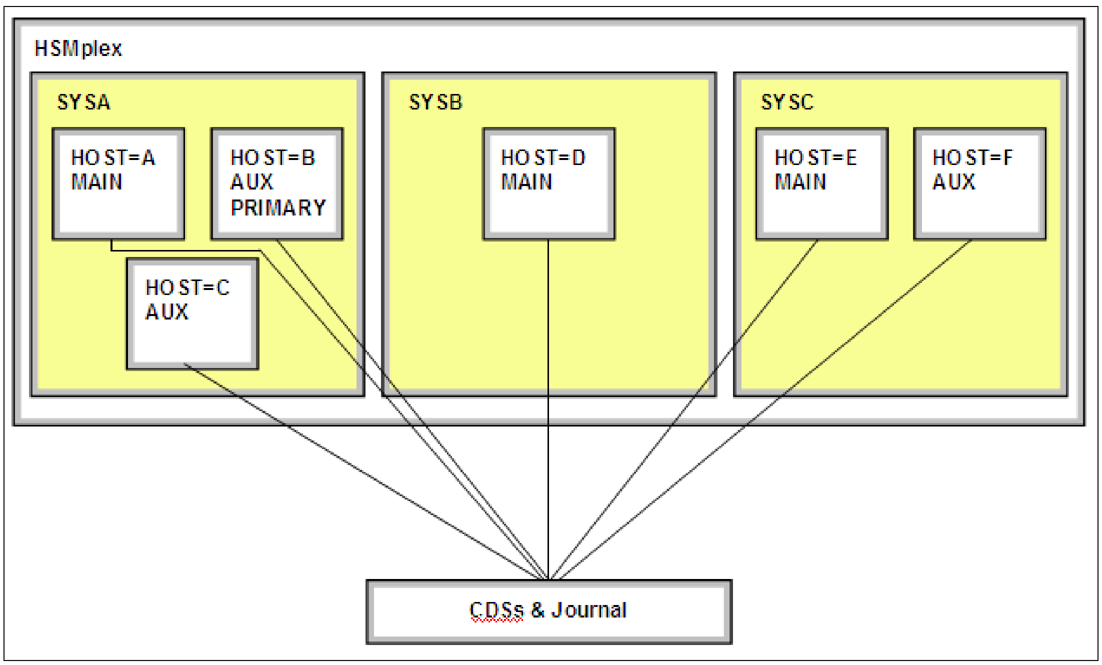

Figure 8-1 Example HSMplex configuration

In Figure 8-1, the HSMplex consists of six DFSMShsm hosts: Three on SYSA, one on SYSB, and two on SYSC. All six of these hosts share access to the same set of CDSs and journal. Each LPAR has one MAIN host. The MAIN host on each LPAR can perform the following unique functions:

- Process implicit requests, such as recalls (in a non-common recall queue (CRQ) environment) and delete migrated data sets, from user address spaces
- Process explicit commands from Time Sharing Option (TSO), such as **HSEND** and **HMIGRATE**
- Manage aggregate backup and recovery support (ABARS) secondary address spaces

Hosts A, D, and E are MAIN hosts. If an LPAR has more than one DFSMShsm host (MASH LPAR), the hosts other than the MAIN host are designated as auxiliary (AUX) hosts. AUX hosts and MAIN hosts can perform the following functions:

- Allow modify commands from a console
- Run automatic space management, backup, and dump

Hosts B, C, and F are AUX hosts. Each HSMplex has one host that is designated as the PRIMARY host. The PRIMARY host performs certain functions on behalf of all of the hosts in an HSMplex to avoid duplicate effort. See 8.3.1, "Defining a primary DFSMShsm" on page 198 for a list of the duties that a PRIMARY host performs. Host B is the PRIMARY host.

8.2 Data integrity and resource serialization

When a single-host DFSMShsm environment is used, DFSMShsm protects the integrity of its owned and managed resources by serializing access to data within the DFSMShsm address space. In an HSMplex, you must choose one of the following methods of resource serialization:

- *Global enqueue*: User data sets are protected with the ENQ and DEQ macros.
- *Volume reserve*: DFSMShsm issues reserves against the source volume to protect data sets during volume processing. The protection is requested by the RESERVE macro.

Global enqueues use the ENQ macro to obtain access to a resource and the DEQ macro to free the resource. The ENQ macro serializes resources between tasks within the same address space on the same LPAR. If you choose to employ this method of serialization in a multi-LPAR HSMplex, you need to activate the global resource serialization (GRS) element of z/OS or an equivalent cross-system enqueue product.

Volume reserves use the RESERVE macro to obtain access to a resource and the DEQ macro to free the resource. The RESERVE macro serializes an entire volume against updates by other LPARs, but allows shared access between tasks within the same address space, or between address spaces on the owning LPAR.

8.2.1 Global resource serialization

GRS is a z/OS product that is designed to protect the integrity of resources in a multiple DFSMShsm host environment. When the systems that share resources are combined into a global resource serialization complex (GRSplex), GRS serializes access to shared resources. User data sets are protected by associating them with the SYSDSN resource and then passing the SYSDSN token to the other LPARS in the GRSplex. The SYSDSN resource is passed to cross-system (global) enqueues. DFSMShsm shares its resources according to ranges of control that are defined by z/OS that are known as *scopes* in GRS terminology.

These scopes, in association with GRS resource name lists (RNLs), define to the entire complex the resources that are local and the resources that are global. The GRS scopes are listed:

STEP	Scope within a z/OS address space
SYSTEM	Scope within a single z/OS image
SYSTEMS	Scope across multiple z/OS images

The GRS RNLs are listed:

- SYSTEM inclusion RNL

  This RNL lists the resources that are requested with a scope of SYSTEM that you want GRS to treat as global resources.

- SYSTEMS exclusion RNL

  This RNL lists the resources that are requested with a scope of SYSTEMS that you want GRS to treat as local resources.

- RESERVE conversion RNL

  This RNL lists resources that are requested on RESERVE macro instructions for which you want GRS to suppress the hardware reserve.

For the full list of resources that must be considered when you configure your DFSMShsm serialization parameters, see Table 8-1. In this table, each pair of QNAME and RNAME values uniquely identifies a resource; the SCOPE indicates the range of control. The DISP (disposition) column indicates whether DFSMShsm or the system requests the resource for exclusive (EXCL) or shared (SHR) use. The significance of the "Can convert from RESERVE" column is explained in 8.2.3, "Using GRS resource name lists" on page 195. The "Must propagate" column indicates whether GRS must communicate the ENQ to all shared systems.

Table 8-1 DFSMShsm GSRs

QNAME (Major)	RNAME (Minor)	SCOPE	DISP	Can convert from RESERVE	Must propagate	Description
Serialization of control data sets (CDSs)						
ARCAUDIT[a]	ARCBCDS ARCMCDS ARCOCDS	SYSTEMS	SHR	NO	N/A[b]	The associated volume reserves of the CDS volumes prevent updates from other LPARS while AUDIT FIX is running in the LPAR that is issuing the reserve.
ARCBACK[a]	ARCBCDS ARCMCDS ARCOCDS	SYSTEMS	SHR	NO	N/A	The associated volume reserves of the CDS volumes prevent access or updates to the CDSs from other LPARs while the CDSs are backed up in the LPAR that is issuing the reserve.
ARCGPA[a]	ARCBCDS ARCMCDS ARCOCDS	SYSTEMS	SHR	NO	N/A	The associated volume reserves of the CDS volumes prevent access or updates to the CDSs from other LPARs while CDSs are updated.

QNAME (Major)	RNAME (Minor)	SCOPE	DISP	Can convert from RESERVE	Must propagate	Description
ARCGPA	ARCRJRN	SYSTEMS	EXCL	YES	YES	The associated volume reserve of the journal volume prevents access to the journal from other DFSMShsm hosts while the journal is being read or written in the DFSMShsm host that is issuing the reserve.
ARCGPAL[a]	ARCMCDS	SYSTEMS	SHR	NO	N/A	The associated volume reserve of the migration control data set (MCDS) volume prevents access to the level 2 control record (L2CR) from other LPARs while the L2CR is updated in the LPAR that is issuing the reserve.
ARCUPDT[a]	ARCBCDS ARCMCDS ARCOCDS	SYSTEMS	SHR	NO	N/A	The associated volume reserves of the CDS volumes prevent access to the CDS from other LPARs while they are being recovered by UPDATEC processing in the LPAR that is issuing the reserve.
ARCENQG[c]	ARCBCDS ARCMCDS ARCOCDS	SYSTEMS	EXCL	N/A	YES	This enqueue is issued in CDSQ=YES environments and is held while accessing the CDSs from the DFSMShsm host that is issuing the enqueue.
ARCENQG	ARCCDSVF	SYSTEMS	EXCL	N/A	YES	This enqueue is issued to ensure that only one CDS version backup function is running in the HSMplex.
ARCENQG	ARCCDSVD	SYSTEMS	EXCL	N/A	YES	This enqueue is issued to ensure that the data area that is used in CDS version backup is not updated by any other DFSMShsm host while the function is running.
ARCENQG (VSAM record-level sharing (RLS) mode)	ARCCAT	SYSTEMS	SHR	N/A	YES	This enqueue is issued as a CDS update resource only in RLS mode.
Serialization of functions						

QNAME (Major)	RNAME (Minor)	SCOPE	DISP	Can convert from RESERVE	Must propagate	Description
ARCENQG	ARCBMBC	SYSTEMS	EXCL	N/A	YES	This enqueue is issued in SETSYS USERSERIALIZATION environments and ensures that only one instance of move backup versions is running in the HSMplex.
ARCENQG	ARCL1L2	SYSTEMS	EXCL	N/A	YES	This enqueue is issued in SETSYS USERSERIALIZATION environments and ensures that only one instance of level 1 to level 2 migration is running in the HSMplex.
ARCENQG	ARCMCLN	SYSTEMS	EXCL	N/A	YES	This enqueue is issued in SETSYS USERSERIALIZATION environments and ensures that only one instance of migration cleanup is running in the HSMplex.
ARCENQG	EXPIREBV	SYSTEMS	EXCL	N/A	YES	Only one instance of the EXPIREBV command is allowed in an HSMplex.
ARCENQG	RECYC-L2	SYSTEMS	EXCL	N/A	YES	This enqueue is issued during RECYCLE to establish a limit of one instance of recycling migration-level 2 (ML2) tapes in the HSMplex.
ARCENQG	RECYC-DA	SYSTEMS	EXCL	N/A	YES	This enqueue is issued during RECYCLE to establish a limit of one instance of recycling daily backup tapes in the HSMplex.
ARCENQG	RECYC-SP	SYSTEMS	EXCL	N/A	YES	This enqueue is issued during RECYCLE to establish a limit of one instance of recycling spill backup tapes in the HSMplex.

QNAME (Major)	RNAME (Minor)	SCOPE	DISP	Can convert from RESERVE	Must propagate	Description
ARCENQG	COPYPOOL I I *cpname*	SYSTEMS	EXCL	N/A	YES	The scope of the fast replication copy pool extends beyond an HSMplex because a copy pool is defined at the SMSplex level. All DFSMShsm hosts, regardless of which HSMplex they reside in, are prevented from processing the same copy pool. The resource is obtained unconditionally and if the resource is not immediately available, it waits.
ARCENQG	CPDUMP&& *cpname*&& V*nnn*	SYSTEMS	EXCL	N/A	YES	This enqueue is used for the dumping of copy pools.
ARCBTAPE	*volser*	SYSTEMS	EXCL	N/A	YES	This enqueue is used for the Recover Tape Takeaway function.
ARCBTAPE	*volser*. TAKEAWAY	SYSTEMS	EXCL	N/A	YES	This enqueue is used for the Recover Tape Takeaway function.
SERIALIZATION of user data sets by DFSMShsm						
ARCDSN	DSNAME	SYSTEMS	[d]	N/A	YES	For SETSYS USERSERIALIZATION environments, this enqueue enables DFSMShsm to protect the integrity of the data set that relates to concurrent processing in an HSMplex. For SETSYS HSERIALIZATION environments, this enqueue enables DFSMShsm to protect the integrity of the data set that relates to concurrent processing only within the DFSMShsm host that is issuing the enqueue.

QNAME (Major)	RNAME (Minor)	SCOPE	DISP	Can convert from RESERVE	Must propagate	Description
ARCENQG	*dsname*	SYSTEMS	[e]	N/A	YES	This ENQ prevents catalog locate requests from getting a “not cataloged” response in that interval of time during migration or recall when the volser is being changed from MIGRAT to non-MIGRAT or from non-MIGRAT to MIGRAT. It is also used to determine whether a recall with an “in process” flag set on really means “in process” or is a residual condition after a system outage.
ARCBACV	volserx[f]	SYSTEMS	EXCL	YES	YES	This reserve is issued only when you are running in a SETSYS HSERIALIZATION environment when you perform volume backup. The associated volume reserve of the user volume prevents updates of a user data set from other DFSMShsm hosts while it is being copied by the DFSMShsm host that is issuing the reserve.
ARCMIGV	volserx[f]	SYSTEMS	EXCL	YES	YES	This reserve is issued only when you run in a SETSYS HSERIALIZATION environment when you perform volume migration. The associated volume reserve of the user volume prevents updates of a user data set from other DFSMShsm hosts while it is being copied by the DFSMShsm host that is issuing the reserve.
Serialization of user data sets by the system						
SYSDSN	*dsname*	[g]	[e]	N/A	YES	This enqueue is the method MVS allocation uses to provide data integrity when it allocates data sets.

QNAME (Major)	RNAME (Minor)	SCOPE	DISP	Can convert from RESERVE	Must propagate	Description
SYSVSAM	*dsname*	SYSTEMS	[e]	N/A	YES	This enqueue is the method that Virtual Storage Access Method (VSAM) uses to provide data integrity commensurate with the share options of VSAM data sets.
SYSVTOC	*volser*	SYSTEMS	[e]	YES	YES	This enqueue is the method that DFSMSdfp direct access device space management (DADSM) uses to provide the integrity of a volume's volume table of contents (VTOC). **Note:** For more information about journal volume dumps, see 8.2.3, "Using GRS resource name lists" on page 195.

a. These resources are requested only in CDSR=YES environments. In CDSQ=YES environments, serialization is achieved through a global enqueue that uses the Qname of ARCENQG.
b. N/A means "not applicable".
c. Valid if CDSQ=YES.
d. EXCL is used for migration, recall, recover, and ARECOVER. SHR is used for backup and ABACKUP.
e. The DISP of this resource can be either EXCL or SHR and requests are both types.
f. Used only in SETSYS DFHSMDATASETSERIALIZATION.
g. The SCOPE of the SYSDSN resource is SYSTEM only but is automatically propagated by GRS, if the default GRSRNL00 member that is supplied by Multiple Virtual Storage (MVS) is used. GRS-equivalent products also need to propagate this resource.

8.2.2 Serialization groups

Serialization within DFSMShsm is divided into four groups:

- Serialization of control data sets (CDSs)
- Serialization of user data sets by DFSMShsm
- Serialization of user data sets by the system
- Serialization of DFSMShsm functions

Control data set serialization

DFSMShsm provides two parameters that can be included in the startup procedure in SYS1.PROCLIB to control the method of serialization to be used against the CDSs by DFSMShsm:

- CDSQ: Serialize the CDSs by using global enqueues
- CDSR: Serialize the CDSs by using volume reserves

CDSQ

By specifying CDSQ=YES and CDSR=NO in the DFSMShsm startup procedure, you instruct DFSMShsm to serialize the CDSs by using global enqueues and that you have a cross-system serialization product to propagate enqueues across the systems in this HSMplex. Do not specify CDSQ=YES unless you have a cross-system serialization product, such as GRS, to propagate the enqueues.

> **Note:** All hosts in the HSMplex must implement the same serialization method.

If you want to start an instance of DFSMShsm in AUX mode, you must specify this method of serialization (or CDSSHR=RLS).

CDSR

By specifying CDSR=YES and CDSQ=NO in the DFSMShsm startup procedure, you instruct DFSMShsm to serialize the CDSs by using volume reserves. All hosts in the HSMplex must use the same serialization method. If this method of serialization is used, all hosts within the HSMplex must be started in MAIN mode.

Default

If you do not specify a serialization method in the startup procedure, the serialization parameters that are used are CDSR=YES, CDSQ=NO. If an instance of DFSMShsm is started in AUX mode by using these defaults, DFSMShsm will issue message ARC0006I and shut down.

CDSSHR

The `CDSSHR` parameter in the startup procedure is used with your specifications for the `CDSQ` and `CDSR` parameters. Table 8-2 on page 192 shows the serialization that is achieved when you use different combinations of these keywords.

Table 8-2 DFSMShsm serialization with startup procedure keywords

<table>
<tr><th>CDSQ keyword</th><th>CDSR keyword</th><th>CDSSHR keyword</th><th>Serialization</th></tr>
<tr><td>YES</td><td>YES</td><td>YES</td><td>Both CDSQ and CDSR options are used.</td></tr>
<tr><td>YES</td><td>NO or not specified</td><td>YES</td><td>Only the CDSQ option is used.</td></tr>
<tr><td colspan="2">With any other combination of specifications</td><td>YES</td><td>Only the CDSR option is used.</td></tr>
<tr><td>N/A</td><td>N/A</td><td>RLS</td><td>Uses VSAM RLS.</td></tr>
<tr><td colspan="2">With any other combination of specifications</td><td>NO</td><td>No multiple host serialization.</td></tr>
<tr><td colspan="2">With any other combination of specifications</td><td>Not specified</td><td>Performs multi-host serialization of type CDSQ and CDSR (whichever is specified) only if the DASD volume where the index of the MCDS resides was generated by using SYSGEN as SHARED or SHAREDUP.</td></tr>
</table>

No default exists for the `CDSSHR` parameter. We recommend that you use Table 8-2 to determine the appropriate CDSSHR setting for your environment.

Sharing the CDS in an HSMplex

DFSMShsm determines that it is in a shared CDS environment by performing the following steps:

1. Recognizing the CDSSHR=YES or CSDSHR=RLS startup parameters
2. Determining whether the index component of the migration control data set (MCDS) resides on a DASD volume that was generated (SYSGEN) with a SHARED or SHAREDUP device attribute

If either of these conditions are met, DFSMShsm performs the appropriate serialization.

> **Note:** The one exception to this rule is if CDSSHR=NO is specified. If CDSSHR=NO is specified, DFSMShsm will *not* perform multiple-host serialization regardless of how the DASD volume that contains the index component of the MCDS was generated (SYSGEN).

For example, if the DASD volume that contains the index component of the MCDS was generated with SYSGEN as SHARED and CDSSHR=NO is specified, DFSMShsm will not perform any multiple-host serialization. However, if CDSSHR=YES and the DASD volume where the index component of the MCDS was not generated (SYSGEN) as SHARED, DFSMShsm will perform the appropriate serialization for a shared CDS environment.

Specifying CDSSHR=YES or CDSSHR=RLS in an HSMplex will avoid any damage if the DASD volume is specified incorrectly as non-shared on one or more of the sharing systems.

If the CDSs reside on a SHARED DASD volume but are only used by a single DFSMShsm host, specifying CDSSHR=NO eliminates unnecessary serialization overhead.

> **Note:** Each DFSMShsm host in the HSMplex needs the same setting for the `CDSSHR` startup parameter.

Serialization of user data sets by DFSMShsm

DFSMShsm provides two `SETSYS` commands that can be included in the ARCCMDxx startup PARMLIB member to control the method of serialization to be performed against user data sets by DFSMShsm:

- `USERDATASETSERIALIZATION`: Serialize the data sets by using global enqueues
- `DFHSMDATASETSERIALIZATION`: Serialize the data sets by using volume reserves

USERDATASETSERIALIZATION

By specifying SETSYS USERDATASETSERIALIZATION, you are instructing DFSMShsm to serialize data sets that are processed by volume migration and volume backup at the data set level rather than the volume level during data set processing. You can use this specification in an HSMplex where either the instances of DFSMShsm do not share volumes or you have a cross-system serialization product, such as GRS, to propagate the enqueues across systems.

DFHSMDATASETSERIALIZATION

By specifying SETSYS DFHSMDATASETSERIALIZATION, you are instructing DFSMShsm to serialize data sets that are processed by volume migration and volume backup by reserving the source volume during data set processing. DFSMShsm releases the volume after the data is copied because each volume's data sets are migrated or backed up.

Although volume reserves ensure data integrity, they also prevent users on other systems from accessing other data sets on the reserved volume while DFSMShsm is processing a data set on that volume. Also, if one system issues multiple reserves for a device, that system can tie up the device. Other systems cannot access the shared device until the reserve count on the owning system is zero and the system releases the shared device.

The following considerations must be made when you select the serialization method of the user data set by DFSMShsm:

- The setting of this parameter affects the value of the DFSMShsm integrity age.
- In DFSMShsm Version 1 Release 5, the incremental backup function was restructured to improve the performance of that function. This improvement is effective only when USERDATASETSERIALIZATION is specified. Use the **`SETSYS DFHSMDATASETSERIALIZATION`** command only if it is required by your environment. Otherwise, it is recommended that you use the **`SETSYS USERDATASETSERIALIZATION`** command.
- The Fast Subsequent Migration function supports reconnection only in a USERDATASETSERIALIZATION environment.
- Certain data sets, such as multivolume physical sequential data sets, are processed with **`SETSYS USERDATASETSERIALIZATION`** only.

Default

If you do not specify a serialization method in the startup PARMLIB member, DFHSMDATASETSERIALIZATION is used.

Serialization of user data sets by the system

The system serializes user data sets by using three resources:

SYSDSN This enqueue is the method that MVS allocation uses to provide data integrity when it allocates data sets.

SYSVSAM This enqueue is the method that VSAM uses to provide data integrity commensurate with the share options of VSAM data sets.

SYSVTOC This enqueue is the method that DFSMSdfp DADSM uses to provide integrity of a volume's VTOC.

Non-VSAM user data sets are protected by associating them with the SYSDSN resource. VSAM user data sets are protected by associating them with both the SYSDSN and SYSVSAM resources. The SYSVTOC resource is used to protect the integrity of a volume's VTOC.

In an HSMplex, the SYSDSN and SYSVSAM resources must be propagated across all of the systems by using GRS or an equivalent cross-system enqueue product. The SYSVTOC resource needs to be propagated only if a global enqueue is used to serialize it.

Serialization of DFSMShsm functions

Certain DFSMShsm functions in an HSMplex environment can be executed only by one DFSMShsm host at a time. To ensure that this requirement is met, certain resources need to be propagated across all systems in an HSMplex by using GRS or an equivalent cross-system enqueue product. Table 8-3 displays these resource names and a description of the function that each resource name provides.

Table 8-3 GRS resource names

Major (qname) resource name	Minor (rname) resource name	Serialization result
ARCENQG	ARCL1L2	Allows only one DFSMShsm host to perform level 1 to level 2 migration.
	ARCMCLN	Allows only one DFSMShsm host to perform migration cleanup.
	ARCBMBC	Allows only one DFSMShsm host to move backup versions.
	RECYC-L2	Allows only one DFSMShsm host to perform recycle on ML2 tape volumes.
	RECYC-SP	Allows only one DFSMShsm host to perform recycle on spill tape volumes.
	RECYC-DA	Allows only one DFSMShsm host to perform RECYCLE on daily tape volumes.
	EXPIREBV	Allows one instance of the **EXPIREBV** command in an HSMplex.

8.2.3 Using GRS resource name lists

You can use three GRS resource name lists to configure your environment:

- SYSTEM inclusion RNL
- SYSTEMS exclusion RNL
- RESERVE conversion RNL

You can use the RNLs to modify the default serialization method for each resource, as described in Table 8-1 on page 186. Each resource is defined by using a major name (qname) and a minor name (rname). Each RNL definition must be included in the GRSRNLxx member of SYS1.PARMLIB.

SYSTEM inclusion RNL

This RNL is used to list the resources that are requested with a scope of SYSTEM that you want GRS to treat as global resources. If you are in a multiple-LPAR DFSMShsm environment, you must be sure to include the SYSDSN resource in this RNL to ensure that the resource is propagated across all systems. (The SYSDSN resource is included, by default, in the GRSRNL00 member that is supplied in z/OS.) You must include this resource in the GRSRNLxx:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN)
```

SYSTEMS exclusion RNL

This RNL is used to list the resources that are requested with a scope of SYSTEMS that you want GRS to treat as local resources. If you are in a CDSR=YES and DFHSMDATASETSERIALIZATION environment, you need to include the resources that are shown in Example 8-1 in your GRSRNLxx to ensure that the resources are treated as local resources (SCOPE=SYSTEM).

Example 8-1 Include these resources in your GRSRNLxx

```
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCAUDIT)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCBACK)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCGPA)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCGPAL)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCUPDT)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCBACV)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(ARCMIGV)
```

If you are in a CDSQ=YES and USERDATASETSERIALIZATION environment, you do not need to include any of the previous resources in this RNL because they are used only in a CDSR=YES and SETSYS DFHSMDATASETSERIALIZATION environment. The exception to this rule is the ARCGPA/ARCRJRN resource. This resource is also used when you use global enqueues but you do not need to specify it in any RNL because it is propagated as a SYSTEMS enqueue, by default.

RESERVE conversion RNL

This RNL is used to list the resources that are requested on RESERVE macro instructions for which you want GRS to convert hardware reserves to global enqueues.

If you are in a CDSR=YES and DFHSMDATASETSERIZATION environment, you can convert one or more of the following volume reserves to global enqueues.

Table 8-4 Resources that can be converted from volume reserves to global enqueues

Major (qname) resource name	Minor (rname) resource name
ARCGPA	ARCRJRN
ARCBACV	volserx
ARCMIGV	volserx

If you choose to convert one or more of the volume reserves to global enqueues, the required conversion entries for each of these volume reserves in the GRSRNLxx member are shown in Example 8-2.

Example 8-2 Required conversion entries to convert volume reserves to global enqueues

```
RNLDEF RNL(CON) TYPE(SPECIFIC) QNAME(ARCGPA) RNAME(ARCRJRNL)
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(ARCBACV)
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(ARCMIGV)
```

Note: If the volume that contains the DFSMShsm journal data set is dumped by invoking DFSMSdss for a full volume dump, the ARCGPA/ARCRJRN resource and SYSVTOC/volser resource must both be treated consistently by either treating both as volume reserves or both as global enqueues. Failure to follow this rule can result in lockouts or long blockages of DFSMShsm processing.

For more information about GRS RNLs, see *z/OS MVS Planning: Global Resource Serialization*, SA22-7600.

8.2.4 VSAM SHAREOPTIONS for CDSs

The two methods that can be used to define the share options for the DFSMShsm CDSs are described. You can use the first method if you start only one DFSMShsm host for each LPAR in an HSMplex. You must use the second method if any of the LPARs in the HSMplex contain more than one DFSMShsm host.

Method 1: VSAM SHAREOPTIONS(2 3)

If you start only one DFSMShsm host for each LPAR, the following share option strategy will provide maximum protection against accidental, non-DFSMShsm concurrent updates:

- Define the CDSs with VSAM SHAREOPTIONS(2 3).
- Use the GRS RNL exclusion capability to avoid propagating the VSAM resource of SYSVSAM for the CDS components to the other systems.

Cross-region share option 2 allows only one job at a time to open a data set for output. If that data set is in the SYSTEMS exclusion list, the open is limited to one LPAR. This combination sets a limit of one open for each LPAR with the expectation that the one open will be DFSMShsm. When DFSMShsm is active on each LPAR in the HSMplex, no jobs other than DFSMShsm will be able to update the CDSs. This method is an additional level of protection that is not afforded by SHAREOPTION(3 3).

Add the statements that are shown in Example 8-3 to the GRSRNLxx PARMLIB member to avoid propagating the SYSVSAM resource for the CDS components to the other systems.

Example 8-3 GRS RNLDEF statements for SHAREOPTIONS(2 3)

```
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)
```

If you want to implement this sharing option and start multiple instances of DFSMShsm in the same LPAR (MASH), you must use VSAM record-level sharing (RLS) to access the CDSs (CDSSHR=RLS).

Note: The GRS RNLDEF statements cannot be used with method 1 when you use RLS.

Additional considerations apply when you use this sharing strategy:

- Reserve contentions can occur when an installation does not use a global serialization product and that site processes DFSMShsm and applications concurrently with VSAM data sets on the same volume.
- Do not attempt to reorganize your CDSs while DFSMShsm is running on any LPAR that uses those CDSs.
- You can use this share option to access the CDSs in RLS mode. When you use RLS in a VSAM SHAREOPTIONS(2 3) mode, VSAM allows non-RLS access that is limited to read processing.

Method 2: VSAM SHAREOPTIONS(3 3)

If you start more than one DFSMShsm host for each LPAR or want to use the default starter set to allocate the CDSs, you can use this share option.

The DFSMShsm starter set allocates the CDSs with VSAM SHAREOPTIONS(3 3) to allow multiple DFSMShsm hosts to be started in one or more LPARs in an HSMplex. This method relies on DFSMShsm to provide an appropriate serialization protocol to ensure the read and update integrity of the CDSs.

This share option can be used if you want to use RLS (CDSSHR=RLS) to access the CDSs.

Note: Do not attempt to reorganize your CDSs while DFSMShsm is running on any LPAR that uses those CDSs.

Important integrity exposure: Even with one of these share options chosen, a data integrity exposure might exist if DFSMShsm is not active on all LPARS in the HSMplex. Because of this exposure, you must strictly control any CDS reorganization or maintenance procedure so that the utility job is allocated with a disposition of OLD (DISP=OLD). A disposition of OLD causes an exclusive enqueue to be issued against the SYSDSN resource for the CDS cluster name. Because any active DFSMShsm in an HSMplex holds a shared enqueue on this resource, the job cannot process until each of the active DFSMShsm hosts in the HSMplex is shut down.

8.3 Starting DFSMShsm in an HSMplex

The tasks to start DFSMShem in an HSMplex are described.

8.3.1 Defining a primary DFSMShsm

In an environment with multiple DFSMShsm hosts (in one or more LPARS), define one host as the PRIMARY DFSMShsm. This host automatically performs backup and dump functions that do not relate to one data set or volume. The following “level functions” are included:

- Automatic backup processing:
  - Phase 1: Backing up CDSs
  - Phase 2: Backing up data sets that migrated before being backed up
  - Phase 3: Moving backup versions of data sets from migration-level 1 (ML1) volumes to backup volumes
- Automatic dump processing:
  - Phase 1: Deleting expired dump copies
  - Phase 2: Deleting excess dump VTOC copy data sets

The storage administrator must specify the primary DFSMShsm in the DFSMShsm startup procedure, or as a parameter on the `START` command. If no primary DFSMShsm is specified, DFSMShsm does not perform the level functions that are listed. If you start more than one primary DFSMShsm, DFSMShsm might process the level functions more than once a day.

The primary host can be either a MAIN or an AUX host. Designating an AUX host as the primary host reduces contention between its level functions and the responsibilities that are unique to the MAIN host, such as recalls and deletes.

8.3.2 Defining all DFSMShsm hosts in a multihost environment

In a multiple DFSMShsm host environment, ensure that the host identifier for each host is unique by considering how you specify the **HOST=x** keyword of the DFSMShsm startup procedure. The **x** keyword represents the unique host identifier for each host. For more information and examples, see Chapter 3, "Getting started" on page 29.

If you choose to use a single startup procedure to start multiple DFSMShsm hosts in a single LPAR, two alternatives exist to identify these startups for subsequent **MODIFY** commands, as shown in Example 8-4 on page 199.

Example 8-4 Two alternatives to identify startup procedures

```
S procname.id1,HOST=A,HOSTMODE=MAIN,other parms
S procname.id2,HOST=B,PRIMARY=Y,HOSTMODE=AUX,other parms
```

or

```
S procname,JOBNAME=id1,HOST=A,HOSTMODE=MAIN,other parms
S procname,JOBNAME=id2,HOST=B,PRIMARY=Y,HOSTMODE=AUX,other parms
```

The common procedure needs to specify PRIMARY=N so that you need to override it only for the one primary host.

If you need to issue the same command to multiple DFSMShsm hosts that are started with identifiers with the same set of leading characters, you can use an asterisk wildcard with the **MODIFY** command, as shown in Example 8-5.

Example 8-5 Modify command with an asterisk wildcard

```
F DFHSM*,Q ACT
```

8.4 HSMplex considerations

The considerations for implementing an HSMplex are described.

8.4.1 CDS backup version considerations

The CDS backup version considerations vary depending on the devices to which you back up your CDSs.

CDS backup to DASD

The CDS backup function is executed by the PRIMARY host as the first phase of automatic backup. If you want a secondary DFSMShsm host to back up the CDSs by using the **BACKVOL CDS** command, you need to ensure that the pre-allocated backup versions are accessible to the secondary DFSMShsm host.

CDS backup to tape

If your CDSs are not in a shared user catalog, the backup versions will not "roll off" if they are created on one LPAR and deleted on another LPAR.

8.4.2 Problem determination aid considerations

You must allocate unique PDOX and PDOY data set names for each DFSMShsm host if the data sets are allocated to a catalog that is shared between the DFSMShsm hosts. Allocating the data sets on a volume that is managed by another DFSMShsm host can cause performance degradation or a lockout.

We recommend that you create the problem determination aid (PDA) data set names to reflect the DFSMShsm host that they correspond to. For example, if the host ID is 1, the PDA data set names are allocated in the following manner:

- *qualifier*.HSM1.HSMPDOX
- *qualifier*.HSM1.HSMPDOY

8.4.3 Small-data-set-packing data set considerations

You need to define a (2 x) share option for your small-data-set packing (SDSP) data sets. In an HSMplex that uses a cross-system enqueue product, this option prevents any job on the LPARs in the HSMplex from writing to an SDSP while a DFSMShsm host has an SDSP open for output. A (3 x) option allows programs other than DFSMShsm to write to an SDSP while a DFSMShsm host has an SDSP open for output.

8.4.4 Volume considerations

Consider the following factors when you implement DFSMShsm in a multiple DFSMShsm host environment:

- DFSMShsm does not reserve the volume that contains a user's data set if the user issues a request to migrate or back up the data set, but depends upon global serialization of the SYSDSN resource.
- While DFSMShsm calculates the free space of a volume, it reserves the volume. This task can momentarily interfere with the response time for the functions that require access to that volume.
- Run automatic primary space management, backup, and dump during periods of low system use and low interactive user activity to reduce contention for data sets among DFSMShsm hosts. When a volume that is managed by DFSMShsm is being processed by space management or backup in a DFHSMDATASETSERIALIZATION environment, other DFSMShsm hosts can experience performance problems if they attempt to access the volume. To eliminate these performance problems, consider the use of USERDATASETSERIALIZATION as an alternative.

8.4.5 JES3 considerations

In a JES3 environment, the storage management subsystem (SMS) volumes and the non-SMS volumes that are managed by DFSMShsm in the DFSMShsm general pool must be shared by all processors. Furthermore, if several of the volumes in a data set or volume pool are also in the general pool, all volumes in both pools must be shared.

8.4.6 Multitasking considerations

When you run multiple tasks in a multiple DFSMShsm host configuration, consider the most effective use of your LPARs and the number of DFSMShsm hosts that run within each of those LPARs. For example, you might ask whether it is more efficient to perform eight tasks with one LPAR or four tasks with two LPARs. For space management and backup, it is generally better to perform eight tasks with one LPAR and distribute those tasks across multiple hosts that run in that LPAR. One LPAR offers better performance than two LPARs as a result of the DFSMShsm CDS sharing protocol.

For performance reasons, run several migration or backup tasks in one LPAR versus running a few tasks in each LPAR of an HSMplex; however, for an RLS environment (CDSSHR=RLS), it is better to spread out the tasks across multiple LPARs.

8.5 DFSMShsm in a sysplex environment

A *sysplex* is a collection of LPARs that work together by using certain hardware and software products to process workloads. The products that make up a sysplex provide greater availability, easier system management, and improved growth potential over a conventional computer system of comparable processing power. Two types of sysplexes exist: base and parallel. A *base sysplex* is a sysplex implementation without a coupling facility. A *parallel sysplex* is a sysplex implementation with a coupling facility.

Systems in a *base* sysplex communicate by using channel-to-channel (CTC) communications. In addition to CTC communications, systems in a *parallel* sysplex use a coupling facility (CF), which is a microprocessor unit that enables high performance sysplex data sharing. Because parallel systems allow faster data sharing, workloads are processed more efficiently.

If you run DFSMShsm in a sysplex environment, the following functions can enhance your ability to successfully manage that environment:

- Single GRSplex serialization

  Allows each HSMplex, within a single GRSplex, to operate without interfering with any other HSMplex.

- Secondary host promotion

  Allows one DFSMShsm host to automatically assume the unique functions of another DFSMShsm host that failed.

- Control DS extended addressability

  Allows CDSs to grow beyond the 4 GB size limit.

- Record-level sharing

  Allows CDSs to be accessed in RLS mode. For information about RLS, see 6.1, “Record-level sharing” on page 96.

- Common recall queue

  Balances the recall workload among the hosts in an HSMplex by implementing a common queue.

- ARCCAT release

  Allows a CDS backup host to communicate through cross-system coupling facility (XCF) to other hosts in an HSMplex to tell them to release the necessary resources to allow CDS backup to begin.

8.5.1 Configuring multiple HSMplexes in a sysplex

If you want to define multiple HSMplexes in a sysplex, you must include the following command in each DFSMShsm host's ARCCMD*xx* member:

```
SETSYS PLEXNAME(HSMplex_name_suffix)
```

The **PLEXNAME** keyword distinguishes the separate HSMplexes within a single sysplex. If only one HSMplex is in a sysplex, you do not need to specify this command. By default, the HSMplex name is ARCPLEX0; the suffix is PLEX0, with a prefix of ARC.

If you specify any HSMplex name other than the default name on one host, you must also specify that name on all other DFSMShsm hosts in that HSMplex.

8.5.2 Single GRSplex serialization in a sysplex environment

If two HSMplexes exist within a sysplex (GRSplex) environment, one HSMplex interferes with the other HSMplex whenever DFSMShsm tries to update CDSs in non-RLS mode or when DFSMShsm performs other functions, such as level 1 to level 2 migration. This interference occurs because each HSMplex, even though each HSMplex has unique resources, uses the same resource names for global serialization. This interference can be avoided by using single GRSplex serialization.

Table 8-5 on page 203 shows the resource names that, when enqueued upon, cause interference between HSMplexes in a sysplex.

Table 8-5 Resources that cause interference between HSMplexes in a sysplex

Major (qname) resource name	Minor (rname) resource name	Serialization result
ARCENQG	ARCBMBC	Enqueues during the attach of the ARCBMBC subtask, which moves backup copies from ML1 to backup tapes.
	ARCCDSVF	Serializes a CDS backup function to ensure that only one CDS backup is running within one HSMplex.
	ARCCDSVD	Enqueues while copying CDSVDATA.
	ARCL1L2	Enqueues L1 to L2 migration. L1 to L2 migration is a function of secondary space management.
	ARCMCLN	Enqueues migration cleanup. Migration cleanup is a function of secondary space management.
	RECYC_L2	Prevents two hosts from recycling ML2 tapes concurrently.
	RECYC_SP	Prevents two hosts from recycling backup spill tapes concurrently.
	RECYC_DA	Prevents two hosts from recycling backup daily tapes concurrently.
	ARCBCDS ARCMCDS ARCOCDS	Enqueues CDSs (not obtained in RLS mode). **Note**: If CDSQ is specified, ARCGPA, ARCxCDS, SYSTEMS, SHARE translates to ARCENQG, ARCxCDS, SYSTEMS, EXCLUSIVE.
	ARCCAT	In RLS mode, enqueues change from ARCGPA/ARCCAT STEP to ARCENQG/ARCCAT SYSTEMS to prevent CDS updates during CDS backup.
	HOST||Hostid	Ensures that only one host is started with this host identifier.
	EXPIREBV	Ensures that only one **EXPIREBV** command runs within an HSMplex.
	COPYPOOL||*cpname*	SYSTEMS enqueue **Note:** The scope of the fast replication copy pool extends beyond an HSMplex because a copy pool is defined at the SMSplex level. All DFSMShsm hosts, regardless of which HSMplex they reside in, are prevented from processing the same copy pool. The resource is obtained unconditionally and if the resource is not immediately available, it waits.
	CPDUMP||*cpname*||V*nnn*	SYSTEMS enqueue **Note:** The scope of the fast replication copy pool extends beyond an HSMplex because a copy pool is defined at the SMSplex level. All DFSMShsm hosts, regardless of which HSMplex they reside in, are prevented from processing the same copy pool. The resource is obtained unconditionally and if the resource is not immediately available, it waits.
ARCGPA	ARCRJRN	The volume reserve of the journal volume.
ARCBTAPE	*volser*.TAKEAWAY	Allows recover tape takeaway.
ARCBTAPE	*volser*	Allows recover tape takeaway.

Single GRSplex serialization allows DFSMShsm to translate minor global resource names to unique values within an HSMplex, therefore avoiding interference between the HSMplexes in a sysplex.

To use this new translation technique, you must specify the following setting in your DFSMShsm startup procedure:

```
RNAMEDSN=YES
```

When this setting is specified, DFSMShsm will invoke the new method of translation and will append the CDS and journal data set names that are associated with the function, CDS, or journal that is being serialized to the minor (rname) resource name that is used for that serialization. The result is a unique resource name because the CDS and journal data set names are unique to each HSMplex. Table 8-6 shows the translated minor (rname) resource names when RNAMEDSN=YES is specified.

Table 8-6 Minor (rname) resource translations

Current minor (rname) resource name	Translated minor (rname) resource name
ARCBMBC	ARCBMBC&bcsdsn
ARCCDSVF	ARCCDSVF&mcdsdsn
ARCCDSVD	ARCCDSVD&mcdsdsn
ARCL1L2	ARCL1L2&mcdsdsn
ARCMCLN	ARCMCLN&mcdsdsn
RECYC_L2	RECYC_L2&ocdsdsn
RECYC_SP	RECYC_SP&ocdsdsn
RECYCL2_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

Default

If RNAMEDSN=NO is specified or is not present in the DFSMShsm startup procedure, the new translation method will *not* be invoked.

Note: All DFSMShsm hosts within an HSMplex must use the same translation technique. If a host detects an inconsistency in the translation technique, the detecting host immediately shuts down.

Compatibility considerations

Consider the following coexistence issues before you run DFSMShsm within an HSMplex:

- If all DFSMShsm hosts within an HSMplex are running at DFSMS/MVS Version 1 Release 5 or later, all DFSMShsm hosts must use the same serialization method. If not, at least one of the hosts will shut down (that is, each host that detects a mismatch will shut down).
- If an HSMplex has both Version 1 Release 5 and pre-Version 1 Release 5 running concurrently, the Version 1 Release 5 hosts cannot specify RNAMEDSN=Y. If RNAMEDSN=Y is specified, hosts that detect the mismatched serialization method will shut down.
- If two or more HSMplexes run concurrently in a sysplex, each HSMplex that uses an old serialization method will interfere with other HSMplexes. HSMplexes that use the new serialization method will not interfere with other HSMplexes. However, in a two-HSMplex environment, one HSMplex can use the old method and the other HSMplex can use the new method; neither one will interfere with the other.

8.5.3 Secondary host promotion

DFSMShsm allows secondary hosts to take over functions for a failed primary host or secondary space management (SSM) from a failed host. Secondary host promotion ensures continuous availability of DFSMShsm functions.

The following definitions are key to understanding the concept of secondary host promotion:

- An *original host* is a host that is assigned to perform primary host or SSM responsibilities.
- A *secondary host* is a host that is not assigned to perform primary host or SSM responsibilities.
- A *primary host* is a host that performs primary-level functions, such as the following tasks:
  - Hourly space checks (for interval migration and recall of non-SMS data)
  - During autobackup: Automatic CDS backup
  - During autobackup: Automatic movement of backup versions from ML1 to tape
  - During autobackup: Automatic backup of migrated data sets on ML1
  - During autodump: Expiration of dump copies
  - During autodump: Deletion of excess dump VTOC copy data sets
- An *SSM host* is generally the only host that performs SSM functions.
- A host is said to be *promoted* when that host takes over the primary or SSM (or both) host responsibilities from an original host.
- A host is said to be *demoted* when its primary or SSM (or both) host responsibilities are taken over by another host. Each promoted host always has a corresponding demoted host, and vice versa.

When a host is eligible for demotion

For either a base or parallel sysplex, DFSMShsm, by using XCF, can enable secondary hosts to take over any unique functions that are performed by a primary host or SSM host when one of the following conditions occurs:

- The primary or SSM host goes into emergency mode.
- The primary or SSM host is stopped with the **DUMP** or **PROMOTE** keyword.
- The primary or SSM host is stopped while in emergency mode.
- The primary or SSM host is canceled, DFSMShsm fails, or the system fails.
- A promoted host is stopped or failed by any means.

Notes: If an active primary or SSM host was demoted and it did not take back its responsibilities (for example, it is in emergency mode), you can invoke any type of shutdown and the host remains in its demoted status.

Do not change the host ID of a host that was demoted.

Enabling secondary host promotion

To enable secondary host promotion, specify the **SETSYS PROMOTE** command:

```
SETSYS PROMOTE(PRIMARYHOST(YES | NO) SSM(YES | NO))
```

The following options are valid for the **SETSYS PROMOTE** command:

PRIMARYHOST(YES | NO)
Indicates whether you want this host to take over primary host responsibilities for a failed host. The default value is `NO`.

SSM(YES | NO)
Indicates whether you want this host to take over the SSM responsibilities for a failed host. The default value is `NO`.

Consider the following factors when you enable secondary host promotion:

- Only those DFSMShsm hosts that run on DFSMS/MVS Version 1 Release 5 and later are eligible to use secondary host promotion functions.
- This parameter is ignored when the system runs in LOCAL mode. If the system runs in MONOPLEX mode, the secondary host promotion function is active, but it is unable to perform actions because cross-host connections are not enabled.
- Cross-system coupling facility (XCF) must be configured on the system that DFSMShsm is active on and must be running in multisystem mode.
- An SSM host is not eligible to be promoted for another SSM host.
- The **SETSYS** command does not trigger promotion. That is, a host can be eligible to be promoted only for hosts that fail after the **SETSYS** command is issued.
- Do not make a host eligible for promotion if its workload conflicts with responsibilities of the original host or if it is active on a significantly slower processor.
- If the ARCCBEXT exit is used by the primary host, it must be available for use on all hosts that are eligible to be promoted for the primary host. If the ARCMMEXT exit is used by the SSM host, it must be available for use on all hosts that are eligible to be promoted for the SSM host.
- The CDS backup data sets must be cataloged on all systems that are eligible to be promoted for primary host responsibilities.
- In a multisystem environment, DFSMShsm always sets the SETSYS NOSWAP option.

How secondary host promotion works

When a primary or SSM host becomes disabled, all DFSMShsm hosts in the HSMplex are notified through XCF. Any host that is eligible to perform the functions of the failed host will attempt to take over for the failed host. The first host that successfully takes over for the failed host becomes the promoted host. No method exists to assign an order to which hosts take over the functions of a failed host.

If an original host is both a primary host and an SSM host, its responsibilities can be taken over by two separate hosts.

If the promoted host itself fails, any remaining host that is eligible for promotion takes over. If additional failures occur, promotion continues until no hosts remain that are eligible for promotion.

If a secondary host fails while it is promoted for an original host and no remaining active hosts are eligible for promotion, out of any of the secondary hosts that become reenabled before the original host, only the host that was last promoted for the original host can become the promoted host.

> **Note:** For secondary host promotion to work at its highest potential, do not use system affinity. All systems must have connectivity to all storage groups. If system affinity is used, storage groups that are only associated with one system are not processed when that system is not available.

Configuring automatic backup hosts in an HSMplex

If you use secondary host promotion, be careful when you configure automatic backup in an HSMplex:

- More than one automatic backup host needs to be available to ensure that volume backups of managed volumes are performed even when the primary host is disabled.

  > **Note:** Promoted hosts take over only the unique functions of the original host. They do not take over functions that can be performed by other hosts.

- If a secondary automatic backup host is eligible to be promoted for the primary host, its backup window must be offset from the original primary host's window in a way that it can take over from where the original primary host left off. For example, its start time can correspond with the average time that the primary host finishes its unique automatic backup functions.

  > **Note:** These scenarios assume that the primary host is always an automatic backup host.

Configuring automatic dump hosts in an HSMplex

We recommend to always have more than one automatic dump host in an HSMplex to ensure that volume dumps are performed even if the primary host is disabled.

How the take-back function works

When an original host is reenabled to perform its unique responsibilities (through a restart or by leaving emergency mode), the take-back process begins. The take-back process involves the following procedures:

- The promoted host recognizes that the original host is enabled and gives up the responsibilities that it took over (by resetting its windows, its cycles, and its exit settings to the values that existed before it was promoted).

  > **Note:** Any changes that are made to the window and the cycle while the host was promoted will be lost.

- Until the promoted hosts give back the responsibilities, the original host does not perform any of the responsibilities that were taken over by the promoted hosts.

Considerations for implementing XCF for secondary host promotion

The cross system coupling facility (XCF) component of MVS provides simplified multisystem management. XCF services allow authorized programs on one system to communicate with programs on the same system or on different systems. If a system ever fails, XCF services allow the restart of applications on this system or on any other eligible system in the sysplex.

Before you configure XCF in support of the secondary host promotion function, consider the following information:

- Only one DFSMShsm XCF group exists for each HSMplex. The XCF group name is the HSMplex name, with the default name of `ARCPLEX0`.
- One XCF group member exists for each DFSMShsm host in the HSMplex.
- DFSMShsm does not use the XCF messaging facilities.

8.5.4 Control data set extended addressability in a sysplex

As HSMplexes and sysplexes grow larger to accommodate the amount of data that can exist in an environment, the CDS will likely grow beyond the 16 GB size limit for multicluster migration control data sets (MCDSs) and backup control data sets (BCDSs) (each with four clusters with 4 GB for each cluster) and the 4 GB size limit for offline control data sets (OCDSs). With VSAM extended addressability, you can define each CDS so that it can grow beyond these size limitations.

DFSMShsm supports the VSAM key-sequenced data set (KSDS) extended addressability (EA) capability that uses RLS, CDSQ, or CDSR to access its CDSs.

Extended addressability considerations

The following considerations or requirements can affect extended addressability for your CDSs:

- Mixing extended function (EF) clusters and non-EF clusters is permissible because each cluster is treated as a separate entity. However, if any cluster is accessed in RLS mode, all clusters must be accessed in RLS mode.
- Because EF data sets might contain compressed data, DFSMShsm issues warning message ARC0130I RC16 whenever it detects this condition. RC16 means that a certain CDS contains compressed data, which can affect performance.

8.5.5 Common recall queue

The main purpose of using common recall queue (CRQ) in an HSMplex is to balance the recall workload among all of the hosts in an HSMplex. The use of CRQ enables the following alternative configurations:

- Recall servers: Certain hosts can be configured to place recall requests on the CRQ and select recall requests from the CRQ while other hosts are configured to place only recall requests on the CRQ. Issue the **`HOLD COMMONQUEUE(RECALL(SELECTION))`** command on those hosts that you do not want to select recall requests from the CRQ.

  When used in an HSMplex that contains multiple DFSMShsm hosts on the same LPAR, the use of a CRQ can increase the total number of concurrent recall tasks in an LPAR. Without a CRQ, only the MAIN host in an LPAR can process recalls. With CRQ, the total number of recalls that can be processed concurrently on an LPAR is 15x the number of DFSMShsm hosts on that LPAR.

- Non-ML2 tape only: If certain hosts are not connected to tape drives, they can be configured to select only recalls that do not require ML2 tape. Issue the `HOLD RECALL(TAPE)` command on those hosts.
- Nonparticipating hosts: If a host within an HSMplex is not connected to the CRQ in that HSMplex, it will be able to process only the recalls that are submitted on that system. It will not be able to process the recalls that are submitted on other hosts.

CRQ considerations

If the hosts within an HSMplex are on systems that cannot share data between them, they must not share the same CRQ. If sets of hosts cannot share data, each of those sets of hosts can share a unique CRQ in the HSMplex. When more than one CRQ is used in an HSMplex, each CRQ and the set of hosts that connect to it are referred to as a *CRQplex*.

> **Note:** Although it is possible for multiple CRQplexes among hosts in an HSMplex to share data, this type of configuration is discouraged because most of the benefits of CRQ are achieved as the number of participating hosts increases.

ARCCAT release

The CDS backup function requires an exclusive enqueue on the ARCGPA/ARCCAT resource when it accesses the CDSs in a non-RLS environment. The CDS backup function requires an exclusive enqueue on ARCEQNG/ARCCAT when it accesses the CDSs by using RLS. All other functions that use these resources require only a shared enqueue on the resource.

When a long-running function holds this resource (for example, during the migration of a large data set to tape), it prevents CDS backup from obtaining the exclusive enqueue on the resource until that long-running function completes and dequeues the resource. Because CDS backup requires an exclusive enqueue, any function that is queued after CDS backup is unable to process until CDS backup completes.

To minimize this delay, DFSMShsm hosts in the same HSMplex (RLS environment) or within the same LPAR (non-RLS environment) allow the CDS backup process to begin immediately after all pending CDS updates complete. The host that performs the CDS backup sends a notification to the other hosts by using XCF services. All hosts processing functions and tasks that might delay CDS backup complete pending CDS updates, suspend new CDS updates, and release the ARCCAT resource to allow CDS backup to begin. After CDS backup completes, suspended functions and tasks resume.

This functionality does not apply when you run multiple instances of DFSMShsm in the same LPAR where XCF services are not available.

> **Note:** This functionality was introduced with z/OS 1.13. If any pre-z/OS 1.13 hosts are in the HSMplex, this functionality will not occur for those hosts.

9

Space management

DFSMShsm manages and maintains a multitiered environment to ensure that data availability requirements are met at any time in a data set's lifecycle. When data is created, it exists on a primary disk. As specified in the rules that are set in the DFSMS management class, data sets eventually move on to cheaper media, such as compressed disks or a compressed tape media. These moves are based on the data that is no longer referenced.

This functionality is part of *space management* along with the cleanup, release of unused space, reduction of extents, and recalls (retrieval of data sets from Level 1 or Level 2).

The space management function automatically maintains the environment as outlined, but it can be customized to meet your individual requirements. This chapter explains in detail how the function is logically split into subfunctions and how these subfunctions can be influenced and adjusted by parameter settings in DFSMShsm.

The overall description assumes Data Facility Storage Management Subsystem (DFSMS) management of data where storage groups and storage group settings together with management class settings decide how data is managed.

The space management of non-storage management subsystem (SMS) data is covered and the difference between SMS-managed data and non-SMS-managed data is explained.

Migration subtasking, which was introduced in z/OS 2.1, is also explained.

9.1 Functions in primary space management

Functions in primary space management are described.

9.1.1 Tiers used in DFSMShsm space management

The way DFSMShsm moves data between tiers can seem simple to the more experienced user. For new users, Table 9-1 might be helpful. When DFSMShsm moves data off primary volumes, the term *migrate* (or *archive*) is used. When data is retrieved from a lower tier, the operation is called *recall*.

Table 9-1 describes the DFSMShsm tiers that are used in the space management function.

Table 9-1 DFSMShsm tier levels

Tier levels in DFSMShsm	Description
Primary disk: Migration level 0 (also called *ML0*)	Disk that holds online data
Migration-level 1 (also called *ML1*)	Disk that holds DFSMShsm compressed data, which must be recalled to ML0 and unpacked to be readable
Migration-level 2 (also called *ML2*)	Tapes that hold DFSMShsm migrated, compressed data (must be recalled to ML0 and unpacked to be readable)
Small-data-set packing (SDSP)	A VSAM file on the ML1 disk that holds small data sets, which are under a specified size, to save DASD space on ML1

Space management can be split into three main functions with individual subfunctions.

Migration: Movement of data to a lower level

Migration has the following subfunctions:

- Migration from level 0 to level 1
- Migration from level 0 to level 2
- Migration from level 1 to level 2
- Reconnect unchanged data sets that are recalled from level 2 (Fast Subsequent Migration function)

Cleanup-related tasks

Cleanup has the following subfunctions:

- Extent reduction
- Release extents
- Release unused space
- Scratch obsolete and expired data sets (in control data sets (CDSs) and on volumes)

Recall subfunctions

Recall has the following subfunctions:

- Recall data sets from level 1 to level 0
- Recall data sets from level 2 to level 0

9.2 Components in space management

The components in space management are described.

9.2.1 Automatic primary space management function

The automatic primary space management function focuses on migrating data off the primary volumes in a storage group (that is managed by DFSMShsm) that is based on the management class setting (Primary Days value). If this value is exceeded for the individual data set, it is a candidate to migrate to a lower level in the hierarchy (ML1 or ML2). The following tasks are also performed in the primary space management function:

- Deletion of temporary and expired data sets from the volumes (that are managed by DFSMShsm) that are being processed. The management class that is assigned to all data sets that are managed by DFSMS will control this deletion based on the expiration attributes in the assigned management class. Or potentially, the management class that is assigned to all data sets that are managed by DFSMS will control this deletion based on an expiration date that can be assigned to the data set during the define process and stored in the volume table of contents (VTOC).
- The release of unused space that is based on the management class value that was set for this attribute occurs on physical sequential, partitioned, and extended format Virtual Storage Access Method (VSAM) data sets.
- Reconnections of eligible data sets are based on the Fast Subsequent Migration (FSM) function (must be active) and occur to the volumes where the recall occurred previously.
- The Extent Reduction function will be based on the SETSYS value for this function, reducing the number of extents on physical sequential, partitioned, and direct access data sets if they exceeded the specified number of extents. During the process of extent reduction, they also release any unused space in the data sets and compress partitioned data sets.

Primary space management continues until the SMS-managed volumes contain the specified amount of free space, which is under the high threshold on the volume. If deletion of expired data sets, Fast Subsequent Migration, and space reduction (phase one) of the remaining data sets achieve the specified volume free space low threshold, no actual data sets are moved. This cleanup is a two-phase approach, where data set move (phase two) occurs only if phase one does not bring the volume free space below the high threshold.

If data sets are to be migrated during primary space management, they will be migrated in compacted form, if possible. Data sets that are expected to be smaller than 110 KB (where 1 KB is 1,024 bytes) after they are compacted are candidates for small-data-set packing (SDSP), if enabled.

Controlling primary space management

The window for primary space management is controlled through the DFSMShsm **SETSYS** command. You need to consider the planned start and planned end of the function. Use **CYCLESTARTDATE** to define a start date and the execution during the week by setting Y and N values. The following example shows how to code the start cycle for primary space management:

```
DEFINE PRIMARYSPMGMTCYCLE (YYYYYYY CYCLESTARTDATE(1998/03/02))
```

Next, set the window for primary space management. Use the following example of starting the primary space management at 17:00, and running it for 4 hours:

```
SETSYS PRIMARYSPACEMANAGEMENTSTART(1700 2100)
```

Bandwidth for MIGRATION at a DFSMShsm level can be influenced by setting the number of parallel tasks for MIGRATION, which also can be for INTERVAL MIGRATION. Example 9-1 shows how to set this value through the **SETSYS** command.

Example 9-1 Specify the number of migration tasks

```
SETSYS MAXMIGRATIONTASKS(nn)
SETSYS MAXINTERVALTASKS(nn)
```

Other **SETSYS** commands relate to primary space management and migration. In the following example, the extent reduction value is set:

```
SETSYS MAXEXTENTS(12)
```

This value must be exceeded before *extent reduction* occurs.

To understand how data sets are handled through primary (and secondary) space management, see the management class examples (Example 9-2 and Example 9-3 on page 215). The first example shows attributes that decide expiration (based on the creation date or the last used date), release, and how many days a data set will stay on primary DASD.

Example 9-2 Example of management class setting as related to expiration and primary days

```
LINE       MGMTCLAS EXPIRE     EXPIRE        RET    PARTIAL      PRIMARY
 OPERATOR   NAME     NON-USAGE  DATE/DAYS    LIMIT   RELEASE      DAYS
---(1)----  --(2)--- ---(3)---  ---(4)----  --(5)--  ----(6)----  ---(7)--
            FKSMF      NOLIMIT     NOLIMIT  NOLIMIT  YES_IMMED           7
            MCOBTAPE        30          30       30  NO               ----
            MC365          365     NOLIMIT  NOLIMIT  YES_IMMED          10
            MC54NMIG   NOLIMIT     NOLIMIT  NOLIMIT  CONDITIONAL         0
            MC54PRIM   NOLIMIT     NOLIMIT  NOLIMIT  YES                 3
            MC54WORK         7     NOLIMIT  NOLIMIT  YES                 3
            MHLTST         365     NOLIMIT  NOLIMIT  CONDITIONAL        30
            TOGRP1T3         1           1       10  NO                  0
```

Data sets can expire based on creation date (EXPIRE DATE/DAYS) or usage (EXPIRE NON-USAGE) and are captured in primary space management based on this factor. For the Partial Release attribute, the following values can occur:

YES Release unused space automatically during the space management cycle.

CONDITIONAL Unused space can be released automatically only if a secondary allocation exists for the data set.

YES IMMED Release unused space when a data set is closed or during the space management cycle, whichever comes first.

COND IMMED Unused space for data sets with secondary allocation is released when a data set is closed, or during the space management cycle, whichever comes first.

Based on these attributes, primary space management will release the data sets that are candidates.

The last value that is shown in Example 9-2 on page 214 is “Primary Days”, which decides how many days a data set stays on primary DASD. This value can be from *0 days* to *permanently* remaining on the primary disk.

Example 9-3 is an example of “level 1 days” and “generation data group (GDG) action”. The number of days on ML1 is based on the management class value that is determined when space management moves these data sets on to ML2. A value of `0` causes a migration from ML0 to ML2 directly.

GDG cleanup is also part of primary space management. In Example 9-3, when this parameter is set, only two GDG versions will exist. When a new GDG is created, the oldest is rolled off the GDG base and in this case EXPIRED.

Example 9-3 Example of management class setting as related to lifetime on ML1 and GDG handling

```
MGMTCLAS LEVEL 1   GDG ON  ROLLED-OFF
NAME     DAYS     PRIMARY   GDS ACTION
--(2)--- --(8)--  --(10)--  ---(11)---
FKSMF         60       ---  -------
MCOBTAPE -------       ---  -------
MC365         30         2  EXPIRE
MC54NMIG    9999       ---  -------
MC54PRIM    9999       ---  -------
MC54WORK    9999       ---  -------
MHLTST        60       ---  -------
TOGRP1T3       0       ---  -------
```

Definitions that are needed on a storage group level

In addition to the settings in the DFSMShsm PARMLIB, and the management class attributes, you will need settings at the storage group level to activate primary space management.

On the Interactive System Productivity Facility (ISPF) Primary Option menu, choose **ISMF**. Choose option **6** and enter the storage group name in which you want to activate primary space management. The Auto Migrate option must have a value that differs from `N` (none) to activate primary space management.

The possible settings are `Y`, `N`, `I`, or `P`. These settings apply to the following functions:

Y Data sets are eligible for primary space management migration. If SETSYS INTERVALMIGRATION was specified in DFSMShsm, the data sets are also eligible for interval migration. If on demand migration is activated, it will replace interval migration.

N Data sets are not eligible for automatic migration.

I Data sets are eligible for primary space management and interval migration.

P Data sets are eligible for primary space management but not interval migration.

Example 9-4 shows storage group settings for automatic migration.

Example 9-4 ISMF Storage Group LIST view

```
LINE       STORGRP  VIO   AUTO      MIGRATE SYSTEM
 OPERATOR   NAME     UNIT  MIGRATE   OR SYS GROUP
---(1)----  --(2)--- (5)-  --(6)---  -----(7)------
            DB8XL    ----  NO        --------
            DB9A     ----  NO        --------
            DB9ADET  ----  NO        --------
            DB9AML1  ----  INTERVAL  --------
            FKSMF    ----  NO        --------
            SGDB20   ----  YES       SC65TS
```

You either set a value for Migrate System or leave this field empty. For workload balancing, you can leave this field blank and all systems in the DFSMShsm plex share the primary space management workload. DFSMShsm is aware across the HSMplex of volumes that were already processed by another system. This approach is advantageous because you can take out systems from the HSMplex and still process the workload.

The manual workload balancing of automatic migration is also possible by specifying the system name on the individual storage group. Example 9-5 shows output from DFSMShsm primary space management.

Example 9-5 DFSMShsm log from primary space management processing

```
ARC0520I PRIMARY SPACE MANAGEMENT STARTING
ARC0522I SPACE MANAGEMENT STARTING ON VOLUME  279
ARC0522I (CONT.) MHL001 (SMS) AT 16:30:11 ON 2012/08/26, SYSTEM SC64

....... multiple volumes being processed .......

ARC0421I MIGRATION VOLUME X11111 IS NOW MARKED FULL
ARC0521I PRIMARY SPACE MANAGEMENT ENDED SUCCESSFULLY
```

Using DSSXMMODE for migration

For better performance and to offload the DFSMShsm main task for various workloads, migration and other DFSMShsm auto tasks can run in cross-memory mode, starting an individual task outside of DFSMShsm. For DFSMShsm MIGRATION, the task name is in this form: ARC*x*MIGR where *x* indicates the host ID that started the cross-memory task. The following **SETSYS** command shows how to activate cross-memory mode for MIGRATION:

```
SETSYS DSSXMMODE(MIGRATION(Y) )
```

For more information about DSSXMMODE, see 10.1.5, “Using DSSXMMODE” on page 262.

Using ML1 overflow volumes

An optional keyword on the DFSMShsm **ADDVOL** command to add ML1 volumes is **OVERFLOW**. Overflow volumes can be used for both backup and migration.

If you specify SETSYS ML1OVERFLOW(DATASETSIZE(*dssize*)), the ML1 OVERFLOW volumes are reserved for the following migration types:

- Migration of a data set that exceeds the *dssize* value.
- Migration of data sets to ML1 that cannot find sufficient space elsewhere on the ML1 volumes that are added as NOOVERFLOW.

Overflow volumes are also used for the following functions:

- Inline backup
- **HBACKDS** and **BACKDS** commands
- ARCHBACK macro for data sets that are larger than *dssize* K bytes

The following example shows adding ML1 volumes to DFSMShsm:

```
ADDVOL ML1003 UNIT(3390) -
MIGRATION(MIGRATIONLEVEL1 OVERFLOW)
```

The default for the **ADDVOL** command is `NOOVERFLOW`.

Previously, DFSMShsm did not migrate data sets that were larger than 64 K. Now, you can migrate data sets that are larger than 64 K (large format sequential data sets) and use the overflow volumes for migration and backup copies that are based on the installation-specific setting.

The advantage of this configuration is that these data sets no longer must go directly to ML2.

9.2.2 Secondary space management

The automatic secondary space management (SSM) function manages the migration and cleanup from DFSMShsm ML1 volumes. The automatic SSM function performs these functions:

- Migrates data
- Schedules TAPECOPY commands for migration tape copy-needed (TCN) records
- Deletes expired data sets from the *migration volumes*
- Deletes obsolete migration control data sets (MCDs), volume statistics records (VSRs), and data set readys (DSRs) during migration cleanup
- Moves data sets (based on the management class attribute for ML1 days) from ML1 to ML2 volumes

The automatic SSM function determines whether to perform Level 1 to Level 2 migration by checking to see whether any ML1 volume is filled equal to or greater than its threshold. DFSMShsm migrates all eligible data sets from all ML1 volumes to ML 2 volumes.

Note: SSM can have up to 15 concurrent cleanup tasks, which can be a combination of disk and tape; however, still only one task migrates data from ML1 to ML2.

The SSM window and tasks are controlled by the **SETSYS** command, as shown in Example 9-6.

Example 9-6 Example of commands that control SSM

```
SETSYS SECONDARYSPACEMANAGEMENTCYCLE (YYYYYYY) CYCLESTARTDATE(yyyy/mm/dd))
SETSYS SECONDARYSPMGMTSTART(1400)
SETSYS MAXSSMTASKS (CLEANUP(2) TAPEMOVEMENT(1))
```

The number of cleanup tasks can be 0 - 15. The default is `2`. Tape movement tasks can be 0 - 15, as well. The default for this parameter is `1`. If the number of tasks is set to `0`, the function is not running.

If tape duplexing is used, two tapes are used for each task in this function.

Small-data-set packing

Small-data-set packing (SDSP) is a DFSMShsm function to reduce the space requirement when you migrate smaller data sets to ML1. Instead of migrating as individual data sets, these data sets, which are based on a decided maximum size limit, are moved inside a VSAM file on ML1 volumes. These ML1 volumes are called *SDSP*, and packed efficiently. A number of SDSPs can exist so that DFSMShsm can go to a new SDSP when and SDSP fills up. The following example shows how to set this limit:

```
SETSYS SMALLDATASETPACKING(KB(110))
```

For a usable SDSP on an ML1 volume, you must also add this volume with the **SMALLDATASETPACKING** keyword. The manual *z/OS DFSMShsm Implementation and Customization Guide,* SC35-0418*,* explains in detail how to create an SDSP.

DFSMS V1.13 performance improvements for SDSP

The performance improvements for SDSP that were introduced in DFSMS V1.13 are described.

ARCMMEXT overhead removal

The ARCMMEXT exit looks at already migrated data sets to determine whether they need to be moved on to another device or migration level. When the ARCMMEXT exit scans the SDSP, it evaluates all of the data sets in the SDSP. The exit serializes all of the data sets in the SDSP even though they might not be candidates for migration.

Now, secondary space management executes an initial MCD scan to identify candidates to move on to a migration queue (SMQE), which is processed by the ARCMMEXT exit later. SSM obtains improved performance through this process.

Balanced SDSP selection algorithm

Another new tuning option affects the way that DFSMShsm selects the SDSPs. Today, the target SDSP is selected based on an ML1 volume ADDVOL sequence, which means that the same SDSPs are always selected first. These SDSPs are always filled, so they must be reorganized and the other SDSPs are often idle.

DFSMS V1R13 introduces a change in this approach. A new SDSP selection algorithm ensures that the SDSPs are used evenly. This algorithm is based on the SDSP free-space status. Free space on the SDSP is calculated in two ways:

- During ADDVOL processing
- Every time that an SDSP is closed (at the completion of data processing)

We suggest that you allocate several SDSPs to DFSMShsm so that the small data sets that are candidates for SDSP processing are distributed across multiple SDSPs. Also, SDSPs now are deallocated if they are not in use.

> **Note:** If you are in a multiple DFSMShsm host environment and you use the secondary host promotion function, a secondary host can take over the unique functions that were performed by a primary host that failed. Secondary host promotion also allows another host to take over the secondary space management functions from either a primary or secondary host that failed. You must issue the **SETSYS PROMOTE** command in advance to enable any host to support secondary host promotion.

9.2.3 Event-driven space management

Event-driven migration performs space management functions similarly to primary space management functions, except based on events instead of a schedule.

Two types of event-driven migrations are possible:

- Interval migration
- On demand migration

Interval migration and space management

Interval migration occurs at an hourly interval, if it is enabled. Candidate volumes are volumes over the volume threshold. If AUTOMIGRATE(I) is specified, DFSMShsm checks whether the allocation exceeded the midpoint between Migration High and Migration Low. If AUTOMIGRATE(Y) is specified, the Migration High threshold must be exceeded for the volume to be a candidate for migration. In either case, migration does not occur until allocation on the volume falls under the Migration High threshold or no more candidates exist on the volume for migration.

Interval migration tasks differ from automatic space management tasks. For interval migration, the following tasks are completed:

- The following information about the SMS configuration is collected:
  - The volumes that belong to the DFSMS storage groups
  - The management classes that are identified
  - Allocated space on all volumes
- Unmatched data sets are notified through a message.
- Temporary data sets are deleted.
- Migration on candidate data sets occurs.

Unlike primary space management, these functions are not part of an interval migration:

- Partial release
- Extent reduction

Parameter settings at the volume level and data set level further affect the eligibility requirements on both of these functions.

How to enable interval migration

You can activate Interval migration either by a PARMLIB setting in the ARCCMDxx member or by using the **SETSYS INTERVALMIGRATION** parameter. Defining interval migration in PARMLIB requires you either to restart DFSMShsm or to execute the command.

A **QUERY SETSYS** command informs you whether interval migration is active. In addition to the **SETSYS** activation, you also must set the eligibility requirements.

How to disable interval migration

The interval migration function is easily disabled. To disable it, issue the **SETSYS NOINTERVALMIGRATION** command. To disable the function permanently, also remove the definition in PARMLIB or change it to NOINTERVALMIGRATION.

Eligibility requirements for interval migration

The eligibility requirements for interval migration must be met on both the volume and the data set levels.

At the volume level, the following requirements must be met:

- The volume must be either SMS or non-SMS managed.
- The volume is in a storage group with a setting of AM=`I` (Interval Migration) or AM=`Y` (Automigration YES):
  - If the storage group is defined with AM=`I`, the allocation on the volume must exceed the midpoint between the low and high thresholds to be an eligible candidate.
  - If the storage group is defined with AM=`Y`, the volume must exceed the high threshold to be a candidate.
  - If AM=`Y` is defined on the storage group, on demand migration must not be activated because on demand migration will be performed instead of interval migration.
- The SETSYS INTERVAL MIGRATION setting must be in effect.

At the data set level, the following requirements must be met:

- The management class must allow both *command* and *auto migration*.
- The Primary Days Non Usage value in the management class must be exceeded.
- If the data set needs a backup, it fails to be a candidate.

Considerations for interval migration are listed:

- Neither interval migration or on demand migration is performed on volumes in a storage group with a definition of AM=`P` (Automigrate Primary) or AM=`N` (Automigrate NO).
- Interval migration occurs for volumes in a storage group with the attribute AM=`I`, regardless of whether interval migration is active.
- With on demand migration active, on demand migration occurs instead of interval migration if the AM setting on the storage group is `Y`.
- Interval migration can be set up to require an operator's approval before it starts. Set this requirement by issuing a **`SETSYS REQUEST`** command.
- The **`SETSYS MONITOR(SPACE)`** command can be issued to enable notifications to the operator about the space allocation on volumes during interval migration.
- The order in which data sets are migrated is based on size. The largest data set is migrated first, followed by the next largest data set.

9.2.4 DFSMShsm on demand migration

DFSMS V1.13 introduced an alternative for interval migration that is called *on demand migration*. The current interval migration runs at the top of every hour and spends many resources to create an updated space check of every volume before it starts the space management processing of all volumes over the high threshold.

The new on demand migration takes another approach by running continuously. Activating on demand migration enables a new DFSMShsm listener that waits for a new DFSMS signal. This signal is issued whenever an individual volume exceeds its high threshold due to new allocations and extents. Whenever an individual volume exceeds its high threshold, on demand migration is notified to process these volumes immediately, instead of waiting for the top of the hour, as in interval migration.

> **Note:** Interval migration is still available, and it is the default option for DFSMShsm.

The ENF notification occurs through an ENF72 from DFSMS every time the volume high threshold is exceeded. The notification is sent to all of the members in the SMSplex that share the actual COMMDS, through which the communication occurs. Even though all systems are notified, only one system will perform space management. Space management on individual volumes continues until the low threshold is reached or no more data sets need to be migrated, deleted, or expired.

To avoid too many notifications, DFSMS tracks the previous trigger value and issues a new notification on the same volume only if it grows 25% of capacity above the threshold. This notification occurs four times: at 25%, 50%, 75%, and 100%. If the growth is more than 60% of the high threshold capacity, the first two notifications are skipped, but the next notification is issued at 75%. If the capacity reaches a critical value (97% that is used), an ENF notification will be issued on all volume checks. At 100% usage of the volume, another notification is issued.

For extended address volumes (EAVs), both high thresholds (track-managed and cylinder-managed) are monitored. If either high threshold is exceeded, an ENF notification is issued to activate on demand migration for this volume.

Activating on demand migration

To activate on demand migration, you use a new **SETSYS** command: **ONDEMANDMIGRATION (Y|N)**. When set to Y, on demand migration replaces interval migration for all DFSMS-managed volumes. On demand migration performs the space management based on ENF notifications from volumes that exceed thresholds. DFSMS limits this space management to volumes in storage groups with an AM setting of Y. Interval migration no longer runs on DFSMS-managed volumes in these storage groups hourly. Interval migration runs only on non-SMS managed volumes and on volumes in storage groups with an AM setting of I. Consider storage groups with an AM value of I for migration to on demand migration.

By setting the **ONDEMANDMIGRATION** command to N, interval migration acts as it did before DFSMS V1.13. ONDEMANDMIGRATION (N) is the default. The abbreviation for ONDEMANDMIGRATION is ODM.

Note: On demand migration is valid for DFSMS-managed volumes only.

You can set a new notification limit to alert the number of volumes that is selected for on demand migration. If this limit is exceeded, a message is displayed. This example shows the notification that is based on ODMNOTIFICATIONLIMIT:

```
ARC1901E number of volumes eligible for on-demand migration has reached nnnn.
```

The limit is set by the new **SETSYS ODMNOTIFICATIONLIMIT(limit)** command. The default value for this number is 100. Issuing the **QUERY SETSYS** command also displays this value.

DFSMShsm performs the space management of a volume in two phases. The first phase tries to bring the volume under the threshold without moving data. The second phase (if needed) tries to get the volume under the threshold by moving around data.

When you migrate to ONDEMANDMIGRATION, consider your current setting of MAXMIGRATIONTASKS to see whether ONDEMANDMIGRATION might a conflict with potential parallel tasks that are initiated by the interval migration. The SETSYS(REQUEST/NOREQUEST) setting is not valid for ONDEMANDMIGRATION because the operator acceptance is not needed as it might be for interval migration.

ONDEMANDMIGRATION is regarded as part of AUTOMIGRATION. Therefore, a HOLD AUTOMIGRATION also puts a hold on ONDEMANDMIGRATION.

Compatibility and coexistence

When you migrate to DFSMS V1.13, consider the implementation of on demand migration. In a mixed environment, the prior level systems continue their previous approach with interval migration, and the on demand migration support performs space management on the volumes that are outside of the hourly window that is used by interval migration.

The messages that relate to the new function are updated to reflect the status of on demand migration.

9.2.5 Storage group processing priority for space management functions

The DFSMS storage group application in Interactive Storage Management Facility (ISMF) offers a processing priority value option. The priority is used by DFSMShsm space management functions and autobackup for selecting the order in which to process the storage groups to meet individual client requirements. If the processing priority value is not set, the default is a priority of 50 and DFSMShsm processes the storage groups in normal processing order. See the ISMF storage group application with the priority setting in Figure 9-1.

```
                       POOL STORAGE GROUP DISPLAY                  Page 2 of 2
Command ===>

CDS Name  . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : BIGSTUFF

Dump Class . . . . . . . . . . . . . . . . :
Dump Class . . . . . . . . . . . . . . . . :
Dump Class . . . . . . . . . . . . . . . . :
Allocation/Migration Threshold - High  . : 99
                                 Low . . : 1
Alloc/Migr Threshold Track-Managed - High: 99
                                     Low : 1
Guaranteed Backup Frequency  . . . . . . :
BreakPointValue  . . . . . . . . . . . . :
Processing Priority  . . . . . . . . . . : 50
```

Figure 9-1 ISMF Storage Application display for processing priority

If individual priorities are set (values 1 - 100), DFSMShsm then processes the space management functions based on priority from highest to lowest.

This approach can help ensure that more important storage groups are processed before less important storage groups during interval migration, on demand migration, or primary space management.

An example of how to use a prioritized processing order is in a class transition environment, where you want to offload certain storage groups before they become populated from class transition data movement.

9.2.6 Automatic dynamic volume expansion space management

A control unit, such as the DS8000, supports dynamic volume expansion (DVE). With DVE, you can schedule an automatic reformat of the volume. DFSMShsm senses this scheduled automatic reformat and updates the volume information for future use.

Note: This automatic DVE occurs only within the same sysplex. If any DFSMShsm that is outside of the sysplex manages the same volumes, this information is not updated until this DFSMShsm is restarted.

9.2.7 Command space management

With automatic space management, you can also execute the individual functions through commands.

DELETE command

Deletes can occur in multiple ways:

- DFSMShsm **DELETE** command
- Access method services (AMS) **DELETE** command
- TSO **DELETE** command
- IEHPROGM utility

Deleting a DFSMS-managed migrated data set cleans up the migration records in the CDS without recalling the data set. Any discrete profile in RACF that relates to this data set is cleaned up by RACF.

Note: The deletion of a DFSMS-managed data set does not include DFSMShsm backups. The DFSMShsm backups are retained.

Deleting data sets that are not expired

If the data set has an expiration date and the date is not reached, DFSMShsm deletes the data set only if the **PURGE** parameter of the **DELETE** command is specified. If a data set does not have an expiration date, the **PURGE** parameter does not apply, and DFSMShsm deletes the data set.

When users delete data sets with DFSMShsm, they can now bypass unnecessary recalls for PGM=IEFBR14 job steps by updating the ALLOCxx PARMLIB member to include a new statement SYSTEM IEFBR14_DELMIGDS(NORECALL), or by using the new **SETALLOC** command.

Migrating a specific data set

A data set can be command-migrated by using the **MIGRATE** or **HMIGRATE** commands. Migration can occur from a primary volume or an ML1 volume. The values in the management class setting for the number of days on a primary or a level 1 volume will be overwritten by a command migration. To migrate a data set by command, the management class must have migration set to either COMMAND or BOTH. See Example 9-7.

Example 9-7 Command migration of individual data set

```
MIGRATE DATASETNAME(SAMPLE.DATASET)
MIGRATE DATASETNAME(SAMPLE.DATASET1) MIGRATIONLEVEL2
```

Migrating data sets by using wildcards

Command-migrating data to DFSMShsm level 1 or level 2 can also be invoked through a wildcard specification instead of providing a fully qualified data set name.

If you specify **HMIGRATE **.DATA**, DFSMShsm migrates the USERID.HLQ.DATA and USERID.DATA data sets but not USERID.HLQ.LIST.

For easier data set migrations, DFSMShsm bypasses already-migrated data sets when you specify the **HMIGRATE** command with a wildcard filter (*). In releases before DFSMS V1.13, HMIGRATE processing resulted in numerous error messages if your data set name specification included migrated data sets.

Migrating a volume

The **MIGRATE(VOLUME)** command invokes migration at the volume level. Only the data sets with a management class value of BOTH in COMMAND OR AUTO MIGRATE are included.

The following functions are executed:

- Migration occurs from ML0 to ML1 or ML2 only.
- The volume is considered managed by DFSMShsm.
- A list of management classes is obtained.

Recalling DFSMS-managed data sets by command

Recall can occur by specifying RECALL or HRECALL with the specific data set name as a parameter. This process is the same as the automatic recall process. The volume selection routine for this specific type of recall is the same as the volume selection routine for automatic recalls. See 9.2.9, “Automatic recall” on page 226.

The **FORCENONSMS** parameter can be used to direct a recall to a specific non-SMS-managed volume.

Recalls of extended format data sets are implemented differently for VSAM and sequential data sets. For VSAM extended format, data set recall can occur only to the same data set type. For sequential extended format data sets, recall can occur to an extended format data set or to a non-extended format data set.

If the **FORCENONSMS** parameter is used to RECOVER an extended format data set to a *non*-SMS-managed volume, it is recovered as a NON EXTENDED format data set. Extended format data sets can exist only on DFSMS-managed volumes.

Recalling a large data set might cause a problem. If so, use RECALL DFDSSOPTION(VOLCOUNT(N(*nn*) | ANY)) to raise the number of candidate volumes.

Recalling direct access data sets

When you recall basic direct access method (BDAM) data sets to non-SMS-managed volumes, you can use the DAOPTION to specify the device type and how access occurs (relative track or block). For example, the use of RELBLK allows the data set to be recalled to a device type with a smaller track size than the previous primary volume. DAOPTION is used for non-SMS-managed volumes only.

9.2.8 Fast Subsequent Migration optional feature

Fast Subsequent Migration (FSM) is an optional feature that is available in DFSMShsm. When FSM is enabled, it optimizes the migration from the primary disk to ML2. Without FSM enabled, a recall from ML2 to the primary volume invalidates the CDS record that points to the ML2 volume. If the data set is then later migrated back to ML2, migration occurs to a different ML2 tape volume. With FSM enabled, the ML2 recalls and later migration of the same data set to ML2 will change for those data sets that are not being updated during the time they resided on the primary volume. At the time of the migration of these data sets, a reconnection can be made to the previous CDS record that points to the copy on ML2 from where the recall occurred earlier. This capability is not possible if this tape was recycled or if the migration cleanup removed the CDS record.

The FSM function optimizes migration to ML2 processing for the following reasons:

- It reduces data movement and tape mounts that are caused by migrating a data set again that is unchanged.
- Only CDS records are updated.
- It reduces the need for recycling ML2 tape.

Enabling Fast Subsequent Migration

FSM can be enabled in two ways:

- Add the **SETSYS TAPEMIGRATION** command with the **RECONNECT(ALL)** or **RECONNECT(ML2DIRECTEDONLY)** parameter to the ARCCMDxx PARMLIB member of SYS1.PARMLIB. Then, restart DFSMShsm.
- Issue the **SETSYS TAPEMIGRATION** command with the **RECONNECT(ALL)** or **RECONNECT(ML2DIRECTEDONLY)** parameter after DFSMShsm is started.

How to disable FSM

You can disable the FSM function by issuing the **SETSYS TAPEMIGRATION RECONNECT(NONE)** command or by adding the same command in the ARCCMDxx member in SYS1.PARMLIB. DFSMShsm must be restarted.

Exceptions where FSM is valid:

- HFS files (see APAR - OA22497) and VSAM files with an alternate index or path is not supported.
- Data set was updated since it was recalled.
- The data set occupies multiple ML2 tape volumes.
- The **MIGRATE** command was issued with the **CONVERT** parameter.
- The ARCHMIG macro is used with the **FORCML1=YES** parameter to direct the data set to an ML1 volume.
- The **SETSYS TAPEMIGRATION** command with the RECONNECT(NONE) setting is in effect.

The following conditions exist for FSM at recall time:

- The data set is recalled from a single ML2 tape volume.
- The SETSYS USERDATASETSERIALIZATION setting must be in effect.
- The **SETSYS TAPEMIGRATION** command with the RECONNECT(ALL) or RECONNECT(ML2DIRECTEDONLY) was issued.

Migration cleanup is managed through the **SETSYS MIGRATIONCLEANUPDAYS** command. This setting can affect the data sets that benefit from FSM. The FSM pointers are cleaned up on a calculated remigration date plus the value in RECONNECTDAYS (either a specified value or the default). For more details about the calculation, see the *z/OS V1R13.0 DFSMShsm Storage Administration Guide*, SC26-0421. Specify a RECONNECTDAYS value that provides the optimum use of FSM in your environment. The following example shows the **MIGRATIONCLEANUPDAYS** command:

```
MIGRATIONCLEANUPDAYS(recalldays statdays reconnectdays)
```

FSM no longer relies on the change bit in the format 1 data set control block (DSCB) to determine the eligibility to reconnect. Now, FSM looks into two new bits in the format 1 DSCB: one bit to determine that the data set was recalled and another bit to determine whether the data set was opened for output since being recalled. The two bits are DS1IND08 and DS1RECAL.

9.2.9 Automatic recall

Automatic recall is explained.

Volume selection for recalls

When automatic recall is invoked through a reference from a batch job, a TSO user, or a program, DFSMS passes the following information to the automatic class selection (ACS) routines:

- The name of the data set.
- The name of the storage class that is associated with the data set when the data set was migrated. If the data set was not SMS-managed when it was migrated, no storage class is passed.
- The name of the management class that is associated with the data set when the data set was migrated. If the data set used the default management class or was not SMS-managed when it was migrated, no management class is passed.
- The name of the data class that is associated with the data set when the data set was migrated. If the data set did not use a data class or was not SMS-managed when it was migrated, no data class is passed.
- The volume serial number and unit type of the volume where the data set originally resided or of the volume that is specified by the request.
- An environment indicator of RECALL.
- The RACF user ID and group ID for non-DFSMShsm authorized users.
- The data set size in KB.
- The data set expiration date.
- A generation data group (GDG) indication.
- The data set organization.
- The record organization for VSAM data sets.

If DFSMS is active, Data Facility Product (DFP) will return a storage class and manage the allocation. The volume will be assigned by DFSMSdfp selection and not the volume that DFSMShsm bypassed. If DFSMS is not installed, or volume selection was unable to select a volume, the allocation is treated as a non-SMS allocation.

Bandwidth in recall

Set the requirements that relate to recall for your environment. The requirements can depend on factors, such as the volume and the number of concurrently available tape drives. See Example 9-8.

Example 9-8 Example of setting the number of parallel recall tasks

```
SETSYS MAXRECALLTASKS(nn)
SETSYS TAPEMAXRECALLTASKS(nn)
```

This value can be set on each DFSMShsm task in the sysplex. Specify a number `0 - 15` for this value.

Using the ARCCATGP group

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

The following tasks are used to enable DFSMShsm to bypass automatic recall during catalog operations:

- Define the RACF group ARCCATGP by using the RACF **ADDGROUP** command (ARCCATGP).
- Connect users who need to perform catalog operations without automatic recall to ARCCATGP by using the following RACF command: **CONNECT (*userid1*,. . .,*useridn*) GROUP(ARCCATGP) AUTHORITY(USE)**.
- Each user who needs to perform catalog operations without automatic recall must log on to TSO and specify the **GROUP(ARCCATGP)** parameter on the TSO logon window or the GROUP=ARCCATGP parameter on the JOB statement of a batch job. See Example 9-9.

Example 9-9 Jobcard pointing to ARCCATGP group

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

9.2.10 Common recall queue

The common recall queue is described.

Advantages of using the common recall queue

Activating the *common recall queue* (CRQ) feature in DFSMShsm provides an optimized and flexible way to queue and process recalls. The use of CRQ primarily results in putting recalls on a shared queue across the sysplex, which means that all logical partitions (LPARs) can share the recall workload and therefore increase the recall bandwidth.

Additionally, CRQ enables all LPARs to schedule recalls to devices that they cannot access (both disk and tape). For example, if a data set is migrated to ML2, recall can occur from an LPAR without access to a tape device by processing the actual recall by another LPAR in the CRQplex.

CRQ processing of the individual recalls is based on priority across the sysplex instead of by taking the LPAR approach. Also, when CRQ is enabled, DFSMShsm recalls all active requests that are scheduled against the same ML2 tape volume, which increases the effectiveness of the recall process by mounting fewer tape volumes.

CRQ removes affinity to a single system in the sysplex in case the system is down for maintenance or other reasons.

Preparing for common recall queue

CRQ needs a list structure in the coupling facility to be able to share information across the sysplex. This structure must be defined to the coupling facility resource management (CFRM) by the z/OS systems programmer. Before you create this structure, you must estimate the number of recalls, which determines the size that is set in the list structure allocation. Use the coupling facility (CF) sizer to calculate this size. The CF sizer is available at this link:

`http://www.ibm.com/systems/support/z`

The size of the CRQ structure depends on the highest number of projected concurrent recall requests that is expected to be on the queue at any point in time.

The CF sizer recommends that you configure the CRQ to hold no fewer than 3,900 recalls. For the default, the CF sizer configures a size that will be able to hold 1,700 concurrent recalls on the CRQ.

Figure 9-2 shows an example of the CF sizer input panel.

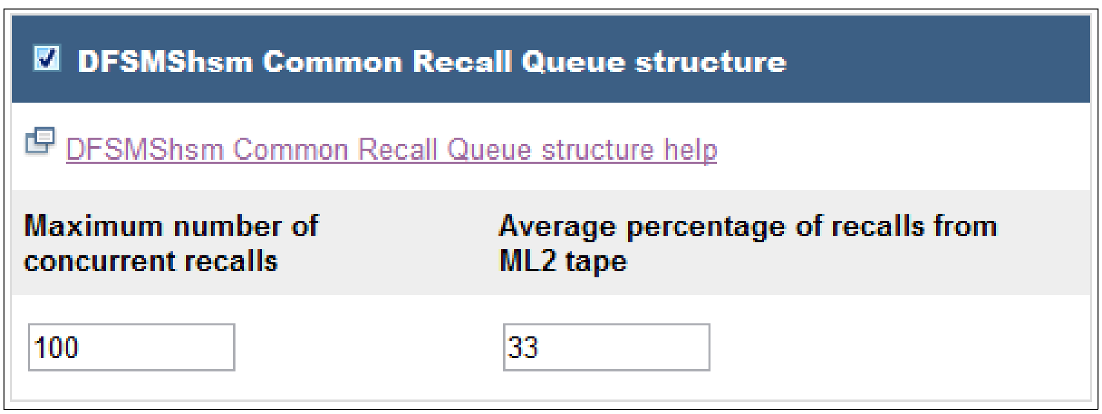

Figure 9-2   Example of CF sizer input panel

The estimated number of concurrent recalls is provided, plus the expected percentage of recalls from ML2. Clicking **Submit** at the bottom of the panel provides the recommended size for the `INITSIZE` and `SIZE` parameters, and also the necessary JCL to create the list structure. Figure 9-3 shows an example of the CF sizer output.

Function	Type	Structure name	INITSIZE	SIZE
DFSMShsm	LIST	SYSARC_bname_RCL	5120	5120

Figure 9-3   CF sizer result panel

DFSMShsm knows about filling the CRQ and stops adding recall requests to the CRQ when the queue reaches 95% of its capacity. Rerouting to the CRQ starts again when the percent full falls under 85%. In the meantime, all scheduled recall requests are routed to the local queue.

If needed, the structure size can be altered dynamically with the standard nondisruptive CF function through the SETXCF **START,ALTER** command.

Enabling and using the common recall queue

You enable the CRQ function in DFSMShsm by adding the **SETSYS COMMONQUEUE(RECALL(CONNECT(*basename*)))** command to the ARCCMDxx member in SYS1.PARMLIB. Next, you must restart DFSMShsm or issue the **SETSYS** command to the DFSMShsm task. The *basename* is a selection of your own. It must be five characters in length. It must be referred to by all of the DHSM tasks that share this CRQ.

By executing the previous command, DFSMShsm connects to the structure and starts to use CRQ. If DFSMShsm is already active and recalls are on the local queue, these recalls are transferred to the CRQ shared queue.

The CRQ eligibility requirements must be met to pass recalls to the CRQ and process them from the CRQ. The main eligibility requirements are that DFSMShsm functions are not held, the CRQ is not over 95% used, and at least one host can process the CRQ. Read more about eligibility requirements in the *DFSMShsm Storage Administration Guide*, SC26-0421.

Note: A CRQ is limited to work in one HSMplex and it cannot be shared across multiple HSMplexes. The CRQ is associated with the HSMplex that created it. DFSMShsm cannot connect to a CRQ that is not associated.

Recall requests on the CRQ are processed in priority sequence. The priority is based on the wait type request with the highest priority. Requests with the same priority are additionally prioritized in first-in first-out sequence.

The host that starts the recall request invokes ARCRPEXT (if enabled). The processing host calls ARCRDEXT (if enabled). ARCRPEXT can be used in the individual DFSMShsm tasks to assign a priority other than the default (50) and process those tasks.

CRQ requests are processsed based on recall tasks available on individual DFSMShsm hosts, and recall prioritization. If requests are on the local queue, local queue requests are prioritized higher than the CRQ requests with the same priority.

When any host selects an ML2 recall request, any other ML2 requests to the same tape volume are also processed. All of these recalls are prioritized. After a wait period, the ML2 tape that is used is unmounted if no more requests to this tape appear on the queue.

Use the **TAPERECALLLIMITS** command if you need to tune this process. This command sets the limit for the amount of time that DFHSM holds the tape:

```
TAPERECALLLIMITS(TASK(time) TAPE(time))
```

The time values for the subparameters **TASK** and **TAPE** specify in minutes how long a recall request can continue (**TASK**) and how long a tape can be allocated (**TAPE**) before DFSMShsm looks for higher priority recalls. If higher priority recalls exist, a tape takeaway occurs.

When the recall processing stops for a specific recall task on a DFSMShsm host, all queued recall requests on that host that require the unmounted tape volume are excluded from consideration for five minutes. This period allows other hosts to try their delayed requests again for the unmounted tape volume.

Read more about the **TAPERECALLLIMITS** command in *DFSMShsm Storage Administration Reference,* SC26-0422.

Two types of messages that relate to the individual recall are sent back to the originating host: the notification to TSO and the message that posts the wait request. All other messages go to the processing host.

Requests must be canceled before they become active on the queue. The cancellation must occur on the originating host. In DFSMS V1.13, the **QUERY** command provides both the originating host and the processing host in the query output.

The CRQ status can be monitored through the **QUERY** command to the CRQ function. See Example 9-10.

Example 9-10 Example of output from a query on CRQ

```
QUERY COMMONQUEUE(RECALL)
ARC1545I COMMON QUEUE STRUCTURE FULLNESS: COMMON
 ARC1545I (CONT.) RECALL QUEUE:STRUCTURE ENTRIES=000% FULL, STRUCTURE
 ARC1545I (CONT.) ELEMENTS=000% FULL
```

Disconnecting from the CRQ

To disconnect from the CRQ feature, you must issue a **SETSYS** command or change the DFSMShsm PARMLIB with the following **SETSYS** command:

```
SETSYS COMMONQUEUE(RECALL(DISCONNECT))
```

Disconnecting from the CRQ function sends all upcoming recall requests in this DFSMShsm task to the local queue. These recall requests are processed in the same manner that they were processed before the CRQ was activated. DFSMShsm will process active CRQ requests that originated from other hosts, but new requests will no longer be picked up from the CRQ.

Disconnection from the CRQ structure occurs during DFSMShsm shutdown, so no action must be taken that relates to CRQ then.

Handling common recall queue failure

If the failing host is processing a request from ML2 tape, all requests from that tape cannot be selected until the failing host is restarted. The tape is marked as “in-use” by the failing host, and that indication is not reset until the failing host restarts. If the failing host is not restarted for a short time, enter the **LIST HOST(*hostid*) RESET** command, where ***hostid*** is the DFSMShsm host ID of the host that failed.

Example 9-11 shows a list and reset by using this command.

Example 9-11 Example of how to reset a serialization from a DFSMShsm host

```
LIST HOST(hostid)
HOST(hostid) RESET
```

You can issue the **LIST HOST** command to receive information about SMS-managed volumes with MCV records that are serialized with a host ID. For *hostid*, substitute the identification character of the DFSMShsm host whose serialization information you want listed or reset.

Recall by FBID

APAR OA24053 introduced the new **SETSYS TAPEDATASETORDER** command in DFSMShsm. The new command provides additional flexibility to handle situations where ascending FBID order offers a significant reduction in tape repositioning, such as when many data sets are recalled or recovered from a single tape. You can specify the **SETSYS** command with the general **TAPEDATASETORDER(PRIORITY | FBID)** parameter and the command will apply to both recall and recover or you can specify a specific functional parameter.

The following examples demonstrate how to issue the command for a specific function:

- SETSYS TAPEDATASETORDER(RECALL(PRIORITY | FBID))
- SETSYS TAPEDATASETORDER(RECOVER(PRIORITY | FBID))

> **Note:** If the **ALTERPRI** command is issued to reprioritize requests that reside on the Recall or Recover queue, the **ALTERPRI** command takes priority over FBID. If you are enabling the new function in a CRQ environment, it is recommended that you apply all PTFs to all systems that connect to the CRQ.

Extended TTOC

When the tape table of contents (TTOC) reaches the limit of 330,000 data sets, future attempts to migrate or back up a data set to that volume will result in spanning to a new migration or backup tape even though the tape still has available capacity.

EXTENDEDTTOC(Y|N) is an optional parameter that specifies whether DFSMShsm uses extended TTOCs to use tapes with higher capacity volumes in your installation more efficiently. Extended TTOCs allow DFSMShsm to write 1,060,000 data sets (potentially) to a migration tape or backup tape.

To use extended TTOCs in your installation, you must define your offline control data set (OCDS) with a maximum record size of 6,144 bytes, as described in the *zDFSMShsm Implementation and Customization Guide*, SC35-0418. Then, enable the support by entering the **SETSYS** command with the optional parameter **EXTENDEDTTOC(Y)** or its shortened form, **EXTTC(Y)**.

> **Note:** If you attempt to specify the **SETSYS EXTENDEDTTOC(Y)** command and the OCDS was not defined with a record size of 6,144, DFSMShsm issues an error message and forces the value of **EXTENDEDTTOC** to N.

To return to using non-extended TTOCs, enter the **SETSYS** command with **EXTENDEDTTOC(N)** or its shortened form, **EXTTC(N)**.

Users must not specify the **SETSYS EXTENDEDTTOC(Y)** command on any host in an HSMplex until the OCDS is redefined with a record length of 6,144 bytes and all hosts in an HSMplex are prepared to issue the **SETSYS EXTENDEDTTOC(Y)** command.

The default value is **EXTENDEDTTOC(N)**.

9.3 Space management of non-SMS-managed volumes

Space management for non-SMS-managed storage occurs at the volume level. These volumes are known as *primary volumes*. Management occurs at the same level as for DFSMS-managed data, but you need to use command settings to specify the space management rules for the individual volumes and associated data sets.

The following functions are performed in space management for non-SMS-managed volumes:

- Automatic space management
- Interval migration
- Automatic recall
- Command space management

The differences between processing for SMS-managed and non-SMS-managed data sets are listed:

- Two additional space management functions are available for non-SMS-managed volumes:
  - Deletion (delete by age)
  - Retirement (delete if backed up)
- Deletion of list and utility data sets.
- Level 1 to level 2 migration is controlled on a processing-unit-wide basis rather than on a data set management class basis.
- Target volumes for recall are selected by DFSMShsm rather than by DFSMSdfp.
- Volumes are identified individually to DFSMShsm by the use of the **ADDVOL** command.

9.3.1 Primary space management

During primary space management on non-SMS managed volumes, DFSMShsm deletes LIST, UTILITY, and expired data sets on the volumes. Partial release is not performed. Only extent reduction is performed.

Fast Subsequent Migration is also supported for non-SMS managed data.

For non-SMS managed data be part of space management, ensure that the following tasks were completed:

- Specify the age for deleting LIST and UTILITY data sets.
- Specify the minimum age for data sets to migrate if no age is specified when the volume was defined to DFSMShsm.
- Specify when data sets become eligible for level 1 to level 2 migration.
- Specify the recall characteristics for primary volumes.
- Define the primary volumes to DFSMShsm.
- Define the pools of volumes.

LIST data set settings

LIST data sets are data sets with the following suffixes:

- LIST
- LINKLIST
- OUTLIST

To clean up these data sets, SETSYS SCRATCHFREQUENCY(*n*) must be defined in the DFSMShsm PARMLIB. The value of *n* in parentheses determines how many days to retain these data sets. A SCRATCHFREQUENCY(999) setting ensures that these data sets are kept for 999 days.

Specifying a minimum age for migration

To decide when to migrate data off the primary volumes, the **`SETSYS DAYS(n)`** setting must be in effect. The value in the **`DAYS(n)`** parameter determines the minimum number of days on a primary disk.

This value is an overall setting that is used if no setting was set on the **`MIGRATE`** command or on the individual **`ADDVOL`** of a primary volume.

Migrating data from ML1 to ML2

Migration from ML1 to ML2 in a non-SMS-managed environment is the same for all data, and it is determined by the SETSYS MIGRATIONLEVEL1DAYS(*n*) setting. This value sets the minimum number of days that data sets resides on an ML1 volume.

> **Note:** You must also specify the threshold on the ML1 **`ADDVOL`** command for DFSMShsm to automatically migrate data from an ML1 volume to an ML2 volume.

Recall in a non-SMS-managed environment

In a non-SMS environment, volume selection is performed by DFSMShsm on recall. The **`SETSYS RECALL`** command specifies the general volume destination to use for recalls. The following command shows the DFSMShsm setting for the target volume for recalls:

```
SETSYS RECALL(ANYSTORAGEVOLUME(LIKE))
```

Setting this parameter enables recalls to go to storage-mounted volumes. Another parameter, **`PRIVATEVOLUME`**, enables recalls to PUBLIC, PRIVATE, or STORAGE-mounted volumes if these volumes were added by an **`ADDVOL`** command with the **`AUTORECALL`** parameter. The use of the **`LIKE`** parameter causes DFSMShsm to only use a volume with the same attributes as the volume from which this data set was migrated. The **`UNLIKE`** parameter causes DFSMShsm to choose a volume with different attributes.

Defining primary volumes to DFSMShsm

The automatic primary space management function occurs in the DFSMShsm host. First, this volume must be in the list.

You can limit processing to only occur in one host by specifying **`AUTOMIGRATION`** on the **`ADDVOL`** command in the selected host. You can also specify **`NOAUTOMIGRATION`** on the other DFSMShsm hosts in the HSMplex.

When you set the ADDVOL of a primary volume to DFSMShsm, consider the following characteristics:

- Do you want automatic space management to run on this volume?
- Do you want this volume to be a candidate for receiving recalls?
- Which space management technique do you want to use?

The potential settings are shown for a sample ADDVOL setting for a primary volume:

```
ADDVOL vvvvvv UNIT(3390) PRIMARY(AUTOMIGRATION AUTORECALL MIGRATE(12)) -
THRESHOLD(95 80)
```

You must add the volumes that you want managed one by one. You must specify the **UNIT** parameter and the volume type (Primary or Migration Level 1). AUTOMIGRATION specifies that DFSMShsm is expected to perform space management on this volume. AUTORECALL allows recalls to target this volume. DELETEBYAGE, DELETEIFBACKEDUP, or MIGRATE specifies the management technique for this volume. In the previous example, data sets will be migrated after a minimum of 12 days on this primary volume.

When you specify DELETEBYAGE(*nnn*) instead, DFSMShsm deletes the data sets without the EXPIRATION DATE set when the data sets are not opened for a number of days that are greater than the value that is specified in the **DELETEBYAGE** parameter. You can specify up to 999 for this value.

DELETEIFBACKEDUP(*nnn*) is another parameter. DFSMShsm deletes data sets with a valid backup and that were not opened in the number of specified days. You can specify up to 999 for this value.

Attributes on primary volumes

In total, you can specify the following attributes to a primary volume:

- AUTOMIGRATION or NOAUTOMIGRATION
- AUTORECALL or NOAUTORECALL
- AUTOBACKUP or NOAUTOBACKUP
- BACKUPDEVICECATEGORY
- AUTODUMP or NOAUTODUMP

Defining pools for non-SMS-managed data sets

You can organize your data in non-SMS-managed pools. You can set up these pools to manage data set names that were defined by you or to receive data sets that were migrated away from these pools earlier.

To define a pool, use the **DEFINE POOL** command and specify one or more volumes to this pool and at the same time give this pool an ID (*name*). Add a statement that is similar to the following statement to define a pool and provide an ID:

```
DEFINE POOL(GROUPA VOLUMES(V11111 V22222 V33333))
```

A volume pool can be defined as shown:

```
DEFINE VOLUMEPOOL (VPOOL1 VOLUMES(V44444 V55555 V66666))
```

Keeping data sets in a volume pool will ensure that data sets that are migrated off this volume pool will be associated with all volumes in the pool so that these volumes later can be target volumes for a recall of these data sets.

You can specify non-resident volumes to the pool to ensure the recall of data sets that were on these primary volumes. Through defining these volumes, the entire pool is associated with the data sets during recall. You can define SMS-managed volumes to the volume pool, but they will not be candidates for the non-SMS data sets that are recalled to this pool. To change the volumes that belong to a volume pool, define the volumes that are already in the pool and add new volumes to the list based on your requirements. A volume pool can hold 140 volumes in total.

One restriction is that all volumes in a pool must have the same geometry (3390, for example). Volumes with different sizes (3390-3 and 3390-9) are allowed to go into the same pool.

A primary volume can appear in multiple pools and it can be added with the **AUTORECALL** parameter on the **ADDVOL** command.

In a multihost DFSMShsm environment, we recommend the following settings:

- A data set or volume pool must be the same in all DFSMShsm hosts.
- Volumes that are available for general recall must be accessible to all DFSMShsm hosts.
- A volume that is removed from the environment must be removed from DFSMShsm by using the **DELVOL** command on *all* DFSMShsm hosts, and from any **ADDVOL** commands that might be in the startup member. This volume can remain in the pool definition. In this way, DFSMShsm can translate a recall request to a currently active volume in the pool.

Note: To define pools in a JES3 environment, see the *z/OS V1R13 DFSMShsm Storage Administration Guide*, SC35-0421.

9.3.2 Automatic secondary space management

Migration cleanup and Level 1 to Level 2 migration for non-SMS-managed data sets work the same way as for SMS-managed data sets.

During migration cleanup for non-SMS-managed data sets, DFSMShsm uses the **SETSYS** parameter **EXPIREDDATASETS(SCRATCH)** to determine whether to delete expired data sets. The **STATDAYS** and **RECALLDAYS** subparameters of the SETSYS MIGRATIONLEVEL1DAYS setting influence when to delete statistics records and MCD records for recalled data sets. The SETSYS MIGRATIONLEVEL1DAYS setting determines when to migrate from Ml1 to ML2.

9.3.3 Automatic primary space management

For automatic primary space management, DFSMShsm processes all primary volumes with the AUTOMIGRATION attribute set. Whether to delete, retire, or migrate a primary volume is determined by the specification on the **ADDVOL** command.

The space management function for non-SMS-managed data occurs in two passes.

Pass one

In pass one, DFSMShsm investigates all data sets on the volumes, independently of filling the volume. A check of the integrity age ensures that the data set can be processed (for environments with USERDATASETSERIALIZATION set, integrity age is 0). The inactive age (number of days unreferenced) must be larger than the integrity age for the data set to be a candidate for migration.

The following actions are not performed for non-SMS-managed data sets:

- Issue a message for unmatched data sets
- Delete temporary data sets
- Release unused space in non-VSAM data sets
- Expire data sets by inactive age or age since creation

For non-SMS-managed data sets, the following actions are performed:

- Delete LIST and UTILITY data sets
- Delete data sets that passed their expiration dates
- Determine eligible data sets for migration, including Fast Subsequent Migration
- Determine eligible data sets to back up
- Determine eligible data sets for extent reduction

Remember, the SCRATCHFREQUENCY determines the cleanup of LIST and UTILITY data sets. EXPIREDDATASETS determines the deletion of data sets that are past expiration, and migration is determined by the DAYS value.

Pass two

In pass two, the following actions are taken:

- Sorts the candidates for migration into priority order
- Performs space management on the volume

Sorting the migration candidate into priority order is based on age and size. Space management potentially performs the following functions, depending on the technique setting on ADDVOL:

- Delete by age
- Delete if backed up
- Migrate

Space management for VSAM data sets

Migration of non-SMS-managed VSAM data sets occurs at the sphere level. The base cluster and index do not need to be defined individually. PATHs are also included automatically.

When DFSMShsm migrates a VSAM sphere, it tracks up to eight components for one data set.

For a non-SMS-managed component, only the components that reside in *one* volume can be migrated. DFSMShsm will not migrate VSAM data sets with a component that needs verification. A VSAM data set with more than one alternate index or path will not be selected for automigration. DFSMShsm will not migrate a data set that is marked as “forward recovery”.

After the migration of the non-SMS-managed volumes, DFSMShsm performs extent reduction in the same manner that it performs extent reduction for SMS-managed data sets.

Automatic interval migration

For non-SMS-managed volumes, interval migration from primary volumes occurs only when the following requirements are met:

- You must define a threshold of occupancy.
- The high threshold of occupancy must be exceeded.
- The primary volume has an attribute of automatic migration.

- The volume was added for automatic migration in a DFSMShsm host with interval migration specified
- The space management technique of migration is set.

Recall volume selection

Recalls from non-SMS-managed volumes differ from recalls of SMS-managed data sets. For non-SMS-managed data sets, DFSMShsm selects the target volume based on the following pools:

- Data set pools
- Volume pools
- General pool (JES3) or a single pool that is configured by DFSMShsm

When a non-SMS-managed recall occurs, DFSMShsm builds a list of up to five candidate volumes for the recall. The space information in DFSMShsm is based on the latest space check that was performed on the primary volumes.

DFSMShsm selects a volume based on the following guidelines:

- If the data set that is recalled has a high-level qualifier (HLQ) that matches the HLQ of a data set pool, the five volumes come from the data set pool.
- If the volume from which this data set was migrated is part of a volume pool, the candidate volumes come from the volume pool.
- If the data set is not restricted to be recalled to a user-defined pool, DFSMShsm selects volumes that were assigned to the default pool. Whether a volume is selected is based on the following factors:
  - The subparameters that were selected for the **RECALL** parameter of the **SETSYS** command.
  - The volumes with the **AUTORECALL** subparameter specified in the **ADDVOL** command and with MIGRATE as the space management technique.
  - The volumes are online.
  - The use attribute of the volume.
  - An attribute of storage for the **RECALL(ANYSTORAGEVOLUME)** parameter.
  - An attribute of storage, public, or private for the **RECALL(PRIVATEVOLUME)** parameter.

When DFSMShsm builds the candidate list of volumes, DFSMShsm considers **LIKE** and **UNLIKE** parameters. If LIKE is specified, DFSMShsm builds a list of volumes with the same attributes as this volume, from where migration occurred. If the number of volumes with these attributes is fewer than five, the list consists of fewer than five candidates.

If UNLIKE was specified, DFSMShsm builds the candidate list based on all of the volumes in the pool in the following order:

- If five or more volumes exist with like recall attributes, all of the volumes in the candidate list have recall attributes that match those recall attributes of the volume from which the data set was migrated.
- If not enough like volumes exist, DFSMShsm selects as many volumes with recall attributes that match the recall attributes of the volume from which the data set was migrated as possible, and selects unlike volumes in the following order:
  - Volumes with matching backup attributes (AUTOBACKUP|NOAUTOBACKUP and BACKUPDEVICECATEGORY)
  - Volumes with matching migration attributes (AUTOMIGRATION|NOAUTOMIGRATION)
  - Volumes with no matching attributes

In the following cases, DFSMShsm does not require matching attributes:

- A data set that is being recalled to a user-defined pool
- A data set that was migrated by a command from a volume with the DELETEBYAGE management technique
- A data set that was migrated by a command from a volume that was not managed by DFSMShsm
- A data set that was migrated from a primary volume that was added to only one host in a multiple DFSMShsm-host environment

When the candidate volume selection is complete, allocation can start. Allocation occurs by selecting the first volume in the list. If this volume cannot meet the space requirements, the next volume is selected. This process continues until allocation is successful or the volume list is exhausted.

If DSS is the data mover, it might not be able to allocate a data set on the volume if the volumes are too fragmented (requires the allocation to occur within five extents).

If the user tries to open a non-VSAM and non-SMS-managed data set by referring both to unit and volser, automatic recall is initiated by DFSMShsm.

For a VSAM data set, DFSMShsm also captures the catalog locate and initiates a recall.

Note: A scratch request to a data set with a volser of MIGRAT is converted to a deletion of a migrated data set and occurs without a recall. A delete PURGE of a data set with an expiration date that was not met results in a recall and deletion.

Command space management

Command space management can be executed in the same way for non-SMS-managed storage as for SMS-managed storage.

Migration from ML1 volumes differs compared to the same function on SMS-managed volumes. For non-SMS-managed volumes, you can also perform the following tasks:

- Migrate from a single primary volume all data sets that are not used within the number of days you specify
- Delete by age from a single primary volume all data sets that are not used within the number of days you specify

- Delete if backed up from a single primary volume all data sets that are not used within the number of days you specify
- Migrate data sets from all non-SMS-managed primary volumes
- Control migration of individual data sets, groups of data sets, or volumes
- Recall a data set to a specific volume
- Recall an SMS-managed data set to a non-SMS-managed volume

Volume space management by command

Volume space management in a non-SMS-managed environment for primary volumes can be performed in two ways. In the first method, space management is performed as it is in automatic primary space management, which is based on the settings on the **ADDVOL** command. The following example shows a MIGRATE VOLUME:

```
MIGRATE VOLUME(V11111)
```

In Example 9-12, the processing in automatic primary space management is still followed, but the technique is taken from the setting on these commands.

Example 9-12 MIGRATE VOLUME example of setting the technique

```
MMIGRATE VOLUME(V22222 DELETEBYAGE(days))
MIGRATE VOLUME(V22222 DELETEIFBACKEDUP(days)
MIGRATE VOLUME(V22222 MIGRATE(days)
```

You can use the **SETMIG** command to manage deviations to the general rules for migration. The **SETMIG** command works at the data set name level, qualifier level, or volume level. The following settings are valid:

- COMMANDMIGRATION
- MIGRATION
- NOMIGRATION

Example 9-13 shows examples of the **SETMIG** command.

Example 9-13 SETMIG examples on exceptions

```
SETMIG VOLUME(V33333) NOMIGRATION
SETMIG DATASETNAME(A.B.C) NOMIGRATION
SETMIG LEVEL(HLG) MIGRATION
```

Recalling non-SMS-managed data sets can occur to a specific volume and can also be forced to a non-SMS-managed volume, even if it was not previously managed by DFSMS.

This action is shown in Example 9-14. The first recall command recalls to a specific volume. If the allocation is captured in the ACS routines, the data set is converted to DFSMS.

The second example recalls a DFSMS-managed data set and converts this data set into a non-SMS-managed data set.

Example 9-14 Recall commands

```
RECALL A.B.C UNIT(3390) VOLUME(V44444)
RECALL A.B.C UNIT(3390) VOLUME(V44444) FORCENONSMS
```

9.3.4 Migration subtasking

DFSMS V2.1 introduced a significant throughput improvement in auto migration by enabling multiple migration subtasks that run in parallel under each migration task. This new function, which is called *migration subtasking*, increases bandwidth in auto migration and offers new possibilities for managing migration requirements.

To understand how this new function changes the behavior of auto migration, it is important to understand how DFSMShsm performed auto migration before DFSMS V2.1.

DFSMShsm auto migration processing was performed in three major serialized steps:

- A preprocessing step that checks eligibility, enqueues for serialization, and updates the CDS and catalog
- A data movement step, which can be class transition (moving data between ML0 tiers) or a migration hierarchy movement (Ml0 to Ml1 or ML2)
- A postprocessing step that performs the dequeue, the updates to the CDS and catalog, and scratch cleanup

For small data sets, the overhead of premigration and postmigration processing is two/thirds of the total processing time.

With the introduction of migration subtasking, parallel subtasks in each migration task enable DFSMShsm to process more data sets in parallel in the same migration task. If the new function is not enabled, migration processing continues as it did before.

How migration subtasking changes auto migration

The DFSMShsm performed auto migration before DFSMS V2.1 as a single-threaded process for each migration task as outlined in the three steps. The parallelism that you can obtain with DFSMShsm V2.1 adds more migration tasks and therefore more data sets at a time.

Figure 9-4 on page 241 shows migration processing before and after DFSMShsm V2.1. The top part shows how the three data sets are processed before DFSMS V2.1 and the lower part shows the migration subtasking implementation as introduced in DFSMS V2.1.

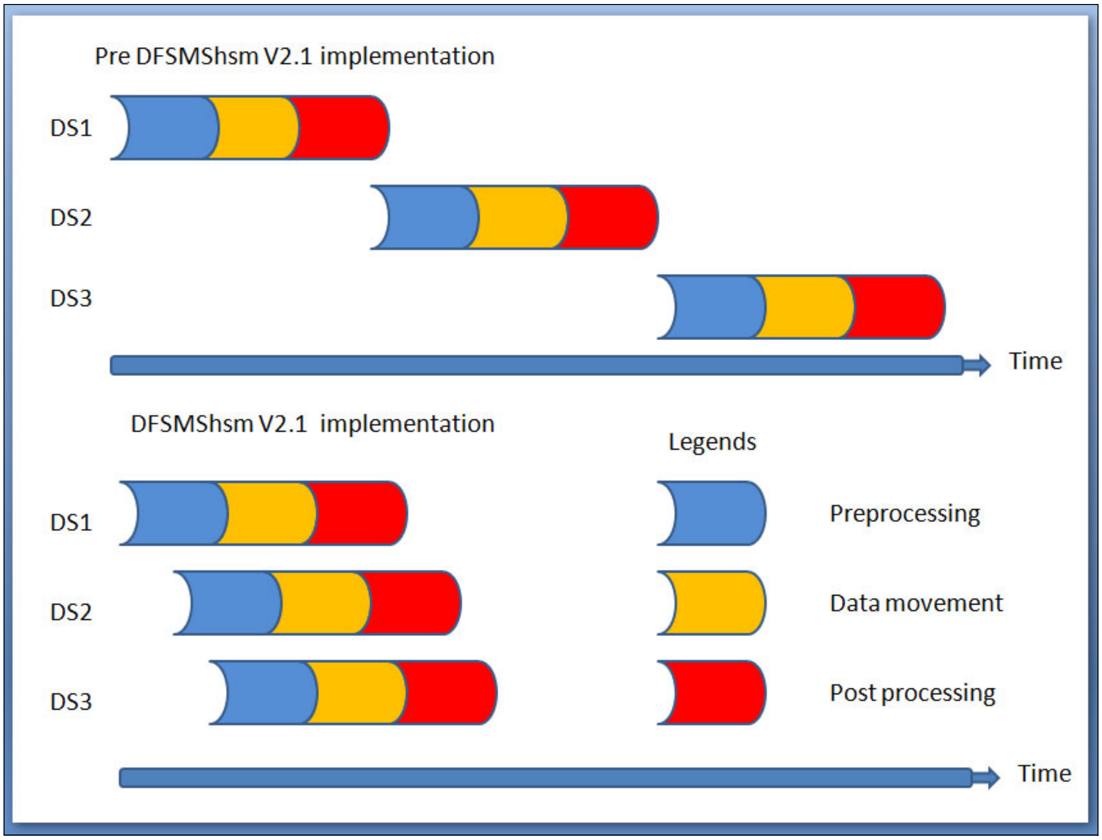

Figure 9-4 DFSMShsm subtasking approach before and after DFSMS V2.1

In DFSMS V2.1, you can start multiple migration subtasks under each migration task to enable parallel processing. The number of subtasks defaults to `5`. The processing for each data set is still serialized in three major processing steps, but multiple data sets can be processed in parallel due to the availability of multiple subtasks. The new feature increases migration bandwidth, which provides relief in large environments where the migration window is a bottleneck.

> **Note:** If auto migration targets ML2 (tape), only one concurrent data movement can occur at any time.

Implementation of migration subtasking

You can implement migration subtasking for all types of auto migration: primary space management, on demand migration, and interval migration.

You activate migration subtasking through the **`SETSYS`** command only in the DFSMShsm startup member. You cannot dynamically activate migration subtasking through a **`SETSYS`** command. The following example shows how migration subtasking is activated:

```
SETSYS MIGRATIONSUBTASKS(YES)
```

The default for the **`MIGRATIONSUBTASKS`** parameter is `NO`. When the function is activated, the default number of tasks is `5` (which can be patched). Testing of this function showed that five tasks are ideal for CPU and memory consumption. The starting default value can be patched up to a maximum value of `105` subtasks.

You can also change the number of subtasks through the **SETSYS** command but you must update the DFSMShsm PARMLIB and restart it to activate the update. Use the **ADDITIONALSUBTASKS** parameter, which is a subparameter of the **MIGRATIONSUBTASKS** parameter, to change the number of subtasks. The following example shows adding two additional subtasks to the default number of subtasks:

```
SETSYS MIGRATIONSUBTASKS(YES ADDITIONALSUBTASKS(2))
```

The command adds the specified number to the current number of active subtasks, up to the maximum limit of 105 subtasks. A **SETSYS QUERY** command displays the actual settings for migration subtasking, as shown in Example 9-15.

Example 9-15 Display of MIGRATIONSUBTASKS settings after adding additional tasks

```
ARC0267I MIGRATIONSUBTASKS=YES,  369
ARC0267I (CONT.) ADDITIONALMIGSUBTASKS=02
```

10

Availability management means backup

DFSMShsm *availability management* is a function to ensure that a recent copy of your direct access storage device (DASD) data set exists. Any lost or damaged data sets can be retrieved at the most recent level. Availability management consists of the following functions:

- Backup
- Dump
- Aggregate backup and recovery support (ABARS)
- Recovery
- Fast replication backup and recovery

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

In this chapter, we describe the preceding functions in detail.

10.1 Backup availability management

One of the ways in which DFSMShsm ensures data availability is by automatically copying new and changed user data sets to a backup volume. The copy of your data set is called a *backup version*. The backup version ensures that your data is still available if your original data set is damaged or accidentally deleted.

Another way in which DFSMShsm ensures data availability is by automatically dumping DASD volumes to tape.

DFSMShsm uses the following functions to ensure that your data is available:

- Backup:
  - Automatic backup
  - Command backup
  - Inline backup
- Recovery:
  - Data set recovery
  - Volume recovery

10.1.1 Automatic backup

The automatic backup function, which is also referred to as *incremental backup,* ensures that current copies of new and changed data sets exist in case the original data sets are damaged or accidentally deleted. The automatic incremental backup function is a data-set-level function when DFSMShsm is processing level 0 volumes or a level 1 migration volume to a backup volume during the automatic backup window.

DFSMShsm relies on the data set's management class attributes to determine how DFSMShsm manages the data set. For DFSMShsm to perform automatic backup, you must set up automatic backup requirements for both the DFSMShsm environment and the data set level.

Setting up automatic backup in DFSMShsm environment

You must define a backup window and backup cycle for DFSMShsm to perform the automatic backup process by specifying the **SETSYS** command in ARCCMDxx PARMLIB:

```
SETSYS AUTOBACKUPSTART(StartTime LatestStartTime QuiesceTime)
```

The variables are described:

- *StartTime*: Automatic backup start time
- *LatestStartTime*: Start automatic backup no later than the specified time
- *QuiesceTime*: Additional volumes for backup are not allowed after this time

Depending on the number of data sets that are modified in a certain day and the number of available tape drives for the DFSMShsm automatic backup function, you can map out the automatic backup window. Assign the number of MAXBACKUPTASKS to be equal to the number of available tape drives that DFSMShsm can use.

If you omit the quiesce time, automatic backup processes for all eligible volumes, which might lengthen the backup window and affect batch or online processes.

Automatic backup is performed in four phases:

1. Backing up the control data sets (CDSs): This process is called *CDS version backup*, and it is controlled by the **CDSVERSIONBACKUP** parameter of the **SETSYS** command. Before it backs up the CDSs and the journal data set, DFSMShsm issues an exclusive enqueue against the CDSs. In a multiple DFSMShsm-host environment, if the CDSs are *not* accessed with Virtual Storage Access Method (VSAM) record-level sharing (RLS), DFSMShsm reserves the volumes that contain the journal and the CDSs. If the CDSs *are* accessed by using RLS, DFSMShsm performs an exclusive enqueue. If cross-system coupling facility (XCF) is available in the multihost environment, the CDS backup host notifies other hosts through XCF that the enqueue is needed.

   Optional: DFSMShsm invokes the CDS backup installation exit (ARCCBEXT) so that you can access or validate the backup versions after DFSMShsm creates them.

   When the CDS backup is complete, DFSMShsm releases the volumes that contain the journal data set and each CDS, and removes the exclusive enqueue against the CDSs, which allows other DFSMShsm functions to proceed.

2. Moving backup versions from ML1 to tape: Backup versions that are created by data set backup commands can be stored temporarily on migration-level 1 (ML1) volumes. After DFSMShsm backs up the CDSs, the primary host moves any backup versions that are created by data set backup commands that temporarily reside on ML1 volumes to daily backup volumes.

   The primary host is the only host that moves backup versions. Backup versions are moved only once a day and normally during automatic backup. However, if automatic backup is not scheduled to run on a particular day, you can use the **ML1BACKUPVERSIONS** parameter of the **FREEVOL** command to move backup versions.

3. Backing up migrated data sets: A migrated data set might not be backed up because it was migrated after it was created or changed but before it was backed up. This condition can occur if a **MIGRATE** command is issued for the data set shortly after the data set is created or changed. The migration can occur to either an ML1 volume or a DASD migration-level 2 (ML2) volume.

   A data set needs to be backed up when all of the following conditions apply:

   – The data-set-changed indicator is on.

   – The value of the Auto Backup attribute in the management class is Y.

   – The value of the ADMIN OR USER COMMAND BACKUP attribute in the management class is not NONE.

   – The value of the Auto Backup attribute in the storage group is Y.

   If DFSMShsm recalls the data set before it is backed up, DFSMShsm does not back up the data set from the migration volume. Instead, DFSMShsm backs up the data set when it backs up the volume that is managed by DFSMShsm that now contains the data set.

   If you specify the **BACKDS** command to back up a migrated data set, DFSMShsm does not back up the data set again during this phase.

   The primary host is the only host that backs up migrated data sets. Migrated data sets are backed up once a day and normally during automatic backup. However, if automatic backup is not scheduled to run on a particular day, you can use the **BACKUP** parameter of the **RELEASE** command to back up migrated data sets.

4. Backing up volumes that are managed by DFSMShsm from a storage group with the auto backup attribute: Before backup processing begins for SMS-managed volumes, DFSMShsm requests from SMS a list of management classes that are defined to the configuration in which DFSMShsm is running. The successful return of the list of management classes indicates that SMS is installed and active. If DFSMShsm does not receive the list of management classes, it issues a message to indicate that backup cannot be performed for SMS-managed volumes.

   If SMS is active, DFSMShsm retrieves the list of volumes that are associated with storage groups that have the AUTO BACKUP attribute value of `Y`. After it retrieves the list of volumes, DFSMShsm starts the backup tasks to the maximum that is specified by the **`MAXBACKUPTASKS`** parameter of the **`SETSYS`** command in the particular host. If the maximum number of backup tasks is reached and still more volumes must be processed, DFSMShsm waits for any already-active volume backup task to complete backing up its current volume. When a volume backup task completes backing up its current volume, it begins backing up the next SMS-managed volume with the automatic backup attribute. This process continues until all SMS-managed volumes are backed up or the quiesce time passes.

   Backup processing examines each data set on the volume individually. Backup processing for each data set performs the following functions:

   – Checks for unmatched data sets (missing catalog or volume table of contents (VTOC) entry or both) and issues a message
   – Deletes temporary data sets
   – Creates a VTOC copy data set entry for the volume
   – Backs up the eligible data sets
   – Resets the data-set-changed indicator and the last backup date

Always start automatic backup on the primary host processor about 10 - 30 minutes before you start backup on any other hosts, depending on the expected time to accomplish the first three phases of automatic backup. This delay allows the CDSs to be backed up, and the other processors do not have to wait for the CDSs. Another tip is to run automatic backup before space management so that backup versions that are created with the **`BACKDS`** command are moved off the ML1 volumes where they temporarily reside.

You also need to define the backup cycle and the date that it will start the backup process. This command is also in ARCCMDxx and the syntax is shown:

```
DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(2012/08/06))
```

The preceding command establishes a 7-day backup cycle that starts on Monday, 6 August 2012. Sunday is a day off for automatic backup. Specifying CYCLESTARTDATE means that the cycle will stay the same through each initialization of DFSMShsm.

We recommend that you code your DEFINE similar to the example. You might find that it is easier to work with a 7-day cycle than a long string of Ys and Ns.

Your string of alphabetic Ys and Ns can represent up to 31 days in the cycle.

If you are running automatic backup every day, we recommend that you use a 1-day backup cycle to reduce the number of partial tapes in use.

DFSMShsm can use either DASD or tape as target volumes. To direct your backup versions either to tape or DASD, the syntax is shown:

```
SETSYS BACKUP(DASD | TAPE)
```

If you are backing up to DASD and the volume becomes full, DFSMShsm moves older backup versions of data sets to other volumes that are known as *spill volumes*. These volumes are usually tape. We recommend that you back up to tape, so spill processing is available. To prevent spill processing, use the **SETSYS SPILL** command. The SETSYS SPILL command syntax is shown:

```
SETSYS SPILL | NOSPILL
```

The following command shows how to use DFSMShsm to back up a data set with a different prefix (change the DFSMShsm backup prefix):

```
SETSYS BACKUPPREFIX(prefix)
```

If you want to create two copies concurrently, you can use the DUPLEX feature. The DUPLEX tape function is an alternative to the TAPECOPY function. The intent of the DUPLEX function is to keep one copy onsite and to store the other copy remotely. The command syntax for creating the duplex for the BACKUP function is shown:

```
SETSYS DUPLEX(BACKUP(Y | N))
```

You also need to consider the number of DFSMShsm concurrent backup tasks to define to DFSMShsm. Normally, the number of backup tasks is equivalent to the number of tape drives that are available to DFSMShsm for backup. The maximum number of backup tasks is 64. The command syntax is shown:

```
SETSYS MAXBACKUPTASKS(n)
```

The data set management class attributes define how the data set is treated for the creation and retention of backup versions. DFSMShsm makes copies (*versions*) of changed data sets that are on level 0 volumes on backup volumes. DFSMShsm uses the management class attributes and the guaranteed backup frequency attribute of the storage group for each data set to determine whether to copy the data set. After the data sets are backed up, DFSMShsm determines from the management class attributes for each data set how many backup versions to keep and how long to keep them.

You specify backup versions to either DASD or tape. If backup is performed to DASD, your DASD volumes can become filled with the current backup versions and the earlier backup versions that were not discarded. When DASD backup volumes are full, DFSMShsm transfers the old (non-current but valid) backup versions to spill backup volumes. The transfer of valid versions from DASD backup volumes can be either to tape or to DASD.

If the backup or the spill processes are performed to tape, the tapes eventually contain many invalid versions that were superseded by the more current versions that were made. When tape volumes contain many invalid versions, they are selected by the recycle process, which moves valid versions to spill backup volumes.

Setting up the automatic backup process at the data set level

For DFSMShsm to automatically back up the SMS-managed data set, the data set management class of AUTO BACKUP must be Y. The data set must be on a volume that belongs to a storage group that is eligible for AUTO BACKUP, as shown in Figure 10-1 on page 248 and Figure 10-2 on page 250.

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

SCDS Name . . . . . . : SYS1.SMS.MHLRES3.SCDS
Management Class Name : MC54PRIM

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

Figure 10-1 Management class backup attributes

The meanings of the attributes in Figure 10-1 are explained for the management class MC54PRIM:

- The BACKUP FREQUENCY attribute determines how frequently changed data sets are backed up automatically. It specifies the number of days that must elapse since the last backup. In this case, one day must elapse before it is eligible for backup again and it must change since the last backup.
- The NUMBER of BACKUP VERS (Data Set Exists) attribute tells DFSMShsm to keep two backup versions while the data set exists.

  The correct use of the BACKUP FREQUENCY and NUMBER OF BACKUP VERSIONS (Data Set Exists) attributes can provide added assurance that you can recover data sets.

  Assume that you specify a backup frequency of one day and a maximum number of versions of three. If the data set changes daily, the user has only two days to detect any errors that occur in the data set before the oldest backup version contains the error. If the error is not detected in two days, the user cannot use a backup version to recover the data set because all backup versions now contain error data.

  Now, assume that you specify a frequency of two days and a maximum number of versions of three. If the data set changes daily, the user has six days to detect the error before the last backup version contains the error. However, each successive backup version contains two days' worth of daily changes, which might not provide enough granularity for your needs.

  You can also control the number of days in which a user can detect errors by increasing the number of retained backup versions.
- The NUMBER of BACKUP VERS (Data Set Deleted) attribute tells DFSMShsm to keep one backup version and retain it for 30 days after the data set is deleted.
- The RETAIN DAYS EXTRA BACKUP VERS: Extra backup versions are retained for five days after they are created.
- ADMIN or USER COMMAND BACKUP: Apart from automatic backup, users can issue commands to back up data sets.

- AUTO BACKUP: Process the eligible data sets.
- BACKUP COPY TECHNIQUE: Use the standard copy process. The Concurrent Copy process is not used.

  If you want to back up data sets by using Concurrent Copy, follow these conditions. The data set must reside on level 0 volumes. The Concurrent Copy will not process either integrated catalog facility (ICF) catalogs or partitioned data sets (PDSs). Use of Concurrent Copy for data set backup is justified only if the data set is database-related and significant value exists in serializing the data set for as short a duration as possible.

  However, if your hardware is sufficient to support Concurrent Copy, seven options are available to you through the management class BACKUP COPY TECHNIQUE attribute:

  R CONCURRENT REQUIRED

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

  P CONCURRENT PREFERRED

  Concurrent Copy needs to be used for backup. A data set is backed up without Concurrent Copy if it does not reside on a volume that is supported by Concurrent Copy or otherwise unavailable for Concurrent Copy.

  S STANDARD

  Data sets are backed up without the Concurrent Copy technique.

  VP VIRTUAL PREFERRED

  Data is processed with virtual Concurrent Copy, if possible. Otherwise, the data is processed by using standard I/O as though CONCURRENT is not specified.

  VR VIRTUAL REQUIRED

  Data is processed with virtual Concurrent Copy, if possible. Otherwise, the data is not processed.

  CP CACHE PREFERRED

  Data is processed with cache-based Concurrent Copy, if possible. Otherwise, the data is processed by using standard I/O as though CONCURRENT is not specified.

  CR CACHE REQUIRED

  Data is processed with cache-based Concurrent Copy, if possible. Otherwise, the data is not processed.

  When DFSMShsm performs either automatic, volume, or command data set backup processing, the management class that is associated with a data set specifies to use a Concurrent Copy BACKUP COPY TECHNIQUE and DFSMSdss as the data mover, and then DFSMShsm invokes DFSMSdss by using the appropriate `CONCURRENT` parameter.

Figure 10-2 shows how to specify the automatic backup (Auto Backup) attribute to a storage group.

```
                      POOL STORAGE GROUP DEFINE           Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : SCTEST
To DEFINE Storage Group, Specify:
 Description ==>
             ==>
 Auto Migrate . . Y  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . Y  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . N  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . . . .
                                      Copy Pool Backup SG Name . . .

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

 DEFINE    SMS Storage Group Status . . . N   (Y or N)
 DEFINE    SMA Attributes . . . . . . . . N   (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-2 Define the Auto Backup attribute for a storage group

If your DFSMShsm environment consists of more than one DFSMShsm host, you can select the system to run the backup process by specifying the system name in the BACKUP SYS/SYS GROUP NAME field.

The value of SETSYS INCREMENTALBACKUP(ORIGINAL | CHANGEDONLY) is defined in ARCCMDxx. If you specify ORIGINAL, DFSMShsm backs up the data sets with no backup copies regardless of the change bit status or data sets with the change bit set to `on`. Specifying CHANGEDONLY means that only data sets with the data set changed indicator set to `on` are backed up.

We recommend that you specify ORIGINAL from time to time (monthly or quarterly) to ensure that all data sets that require backup are backed up.

You can also use the GUARANTEED BACKUP FREQUENCY storage group attribute to control the maximum period of elapsed time before a data set is backed up regardless of its change status, as shown in Figure 10-3 on page 251.

```
                             POOL STORAGE GROUP ALTER          Page 2 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : SGTEST

To ALTER Storage Group, Specify:

 Allocation/migration Threshold :     High    85   (1-100)  Low . .     (0-99)
 Alloc/Migr Threshold Track-Managed:  High    85   (1-100)  Low . . 1   (0-99)
 Guaranteed Backup Frequency  . . . . . . 7        (1 to 9999 or NOLIMIT)
 BreakPointValue  . . . . . . . . . . . .          (0-65520 or blank)
```

Figure 10-3 Guaranteed backup frequency attribute

GUARANTEED BACKUP FREQUENCY provides an additional criterion to create a backup version and guarantees that a backup version of the data set is created after the specified number of days elapses, whether the data set changed or not. The backup version is created either during the DFSMShsm automatic backup or command volume backup incremental processing.

If the data set did not change and the backup version is created because the GUARANTEED BACKUP FREQUENCY was met, that backup version replaces the most recent backup version. (The replacement is not performed for the **(H)BACKDS** and **BACKVOL TOTAL** commands.)

GUARANTEED BACKUP FREQUENCY (as shown in Figure 10-3) takes precedence over the management class attribute BACKUP FREQUENCY. For example, if the BACKUP FREQUENCY is set to six days and GUARANTEED BACKUP FREQUENCY is set to four days, the data set is backed up after four days, whether the data set changed or not. If the attribute values are reversed, the data set is backed up after four days if changed, or after six days if not changed.

GUARANTEED BACKUP FREQUENCY provides a way to reduce the necessary number of tape mounts to recover a volume when a few data sets are not changed. Volume dumps copy all data, even if nearly all of the data is already backed up by incremental backup. Having all of the necessary data for a recovery on a few tapes can reduce the number of necessary tape mounts to recover a volume entirely from incremental backups.

If the data set storage group attribute of AUTO BACKUP is NO, no data sets are backed up by the automatic backup process, regardless of the setting of GUARANTEED BACKUP FREQUENCY. However, if an SMS volume is backed up by the **BACKVOL** command, GUARANTEED BACKUP FREQUENCY is honored for data sets that are not changed.

Note: No interaction occurs between GUARANTEED BACKUP FREQUENCY and the command backup of a single data set or of a total volume.

Storage group processing priority for automatic backup

The ISMF Storage Group application has a processing priority setting, which is set to 50, by default. If the client does not change this setting, DFSMShsm uses the normal processing order.

If the client changes the setting to a different, individual value on the storage groups that are backed up, DFSMShsm uses the processing priority (possible value is 100 - 1) and processes automatic backup on these storage groups based on priority. DFSMShsm selects the storage group with the highest priority first and then lower priority storage groups, always selecting the highest prioritized of the remaining storage groups next. This way, DFSMShsm backs up the most mission-critical storage groups first.

10.1.2 Backing up data sets with a command

You can use a DFSMShsm command to back up data sets.

The data set backup by command function provides the following capabilities:

- Up to 64 data sets for each host can be backed up at one time.
- Data sets can be backed up directly to DASD or to tape.
- If Concurrent Copy is specified, users are notified when the data set backup is either logically or physically complete. In addition, Concurrent Copy supports non-SMS data sets, or if specified on the command, Concurrent Copy overrides the management class for an SMS data set.
- Users can unmount continuously mounted backup tapes.
- Users can tailor the times when DFSMShsm unmounts a tape.
- Users can specify the number of days to retain a specific backup copy of a data set. See 10.1.4, “Backing up a data set manually and the RETAINDAYS keyword” on page 257.
- Users can create a data set backup copy and store it under a different, specified data set name.

You can use the data set backup command to back up an individual data set:

(H)BACKDS *dsname* | *dsn_patterns*

You can direct your backup copies to either DASD or tape:

(H)BACKDS *dsname* TARGET(DASD) | TARGET(TAPE)

You can also back up the data set to a new name:

(H)BACKDS *dsname* NEWNAME(*newdsname*)

When DFSMShsm processes these commands, it first checks the management class for the data set to determine the value of the ADMIN OR USER COMMAND BACKUP attribute. If the value of the attribute is BOTH, a DFSMShsm-authorized user can use either of the commands, and a non-DFSMShsm-authorized user can use the **(H)BACKDS** command to back up the data set. If the value of the attribute is ADMIN, a DFSMShsm-authorized user can use either of the commands to back up the data set, but a non-DFSMShsm-authorized user cannot back up the data set. If the value of the attribute is NONE, the command backup cannot be performed.

Because the data set backup function allows users to target their backups directly to tape, this backup might increase the number of required tape drives and mounts. The **SETSYS DSBACKUP** and the **HOLD BACKUP(DSCOMMAND)** commands manage the task levels of both tape and ML1 DASD.

The **DSBACKUP** command has three parameters, **DASDSELECTIONSIZE**, **DASD**, and **TAPE**, so that you can control the command data set backup environment:

SETSYS DSBACKUP(DASDSELECTIONSIZE(*max standard*) DASD(TASK(*n*)) TAPE(TASK(*n*)))

With the **DASDSELECTIONSIZE** parameter, you can balance the workload between DASD and tape tasks for all WAIT requests that do not target an output device. This parameter is applicable if both tape and ML1 DASD are used for data set backup requests. The defaults for this parameter are 3,000 KB for the *maximum* size and 250 KB for the *standard* size.

With the **DASD** parameter, you can specify the number of concurrent data set backup tasks that can direct backups to ML1 DASD. The value *n* for the **TASK** subparameter is `0 - 64` tasks.

With the **TAPE** parameter, you can specify the number of concurrent data set backup tasks that can direct backups to tape. The value *n* for the **TASK** subparameter is `0 - 64` tasks. Another parameter, **DEMOUNTDELAY**, is available with the **TAPE** parameter:

```
SETSYS DSBACKUP(TAPE(TASK(n) DEMOUNTDELAY(MAXIDLETASKS(drives)- MINUTES(minutes)))
```

With **DEMOUNTDELAY**, you can tailor the DFSMShsm tape mounting and unmounting for command data set backup. **MINUTES(*minutes*)** is the number of minutes that you want DFSMShsm to wait before it deallocates the tape that is associated with continuously inactive (idle) command data set backup tasks. *Minutes* is a value `0 - 1440`. A value of `1440` indicates that DFSMShsm will not unmount the tape until command data set backup tasks are held, a SWITCHTAPES event occurs, or DFSMShsm shuts down, so this value extends across 24 hours.

If you specify **DEMOUNTDELAY(MINUTES(0))**, a tape that is mounted for command data set backup remains mounted long enough to support a continuous stream of WAIT-type backup requests to tape from a single job stream even if the queue of work momentarily becomes empty. The tape remains mounted for up to 5 seconds if **MINUTES(0)** is specified.

A change to the **DEMOUNTDELAY(MINUTES)** value results in setting a new delay time. This new time is calculated from the current time of day. The new delay does not include any time that the tape was idle up to the issuance of this change.

MAXIDLETASKS(drives) is the maximum number of tape drives that **DEMOUNTDELAY** can accommodate. *Drives* is a value `0 - 64`. You can specify a maximum of 64 drives for **MAXIDLETASKS**, regardless of the number of **TAPE** tasks specified. However, the effective number of tasks for **MAXIDLETASKS** is bound by the number of **TAPE** tasks. So, if you specify **MAXIDLETASKS(64)** and **TAPE(TASKS(5))**, the effective **MAXIDLETASKS** is `5`. If later you specify **TAPE(TASKS(7))** again, the effective **MAXIDLETASKS** becomes `7`.

When a command data set backup task that is writing to tape completes, DFSMShsm does not deallocate tape units until all queued command data set backup requests are processed. This design enables all consecutive WAIT-type requests by the same batch job to be performed without an unmount and mount sequence.

If a data set backup command does not specify a target, the data set is directed to either tape or DASD based on the size of the data set and the availability of tasks. If the data set is greater than a specified threshold, the data set is directed to tape. Any data set less than or equal to a specified threshold is directed to the first available task, DASD or tape.

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

Notes:

- A change in **TAPE(TASKS(*n*))** requires DFSMShsm to start additional tape tasks or stop existing tape tasks (at the end of processing the current command data set).
- Be careful when you change the **DEMOUNTDELAY** parameter. For instance, if a backup to tape task completes its work, the value for **MAXIDLETASKS** is greater than zero, and **DEMOUNTDELAY(10)** is in effect, DFSMShsm sets an internal timer for 10 minutes:
  - If no additional work is received after 10 minutes, the task deallocates the drive and ends.
  - If during this 10-minute idle time, the task is selected to perform a backup, the timer is canceled. After the task completes, DFSMShsm again sets a timer value of 10 minutes.
  - If during this 10-minute delay, the time is decreased to 5 minutes by using **DEMOUNTDELAY(MINUTES(5))**, DFSMShsm cancels the current timer and a new timer is established for 5 minutes (not considering any previous time).
  - If **DEMOUNTDELAY(MINUTES(3))** is in effect and you issue an increase, such as **DEMOUNTDELAY(MINUTES(10))**, DFSMShsm cancels the current timer value and uses the new value. This increase resets the countdown to 10 minutes from the time the command is received.
- If a parsing error occurs, DFSMShsm fails the **SETSYS** command with the ARC1605I message. For non-parsing errors but contextual errors, DFSMShsm fails the parameter in error but continues to process all other parameters in the **SETSYS DSBACKUP** command. DFSMShsm treats the **TAPE(TASKS)** and **DASD(TASKS)** parameters as a single entity. For example, **SETSYS DSBACKUP(DASDSELECTIONSIZE(5000 300) DASD(TASKS(45)) TAPE(TASKS(20)))** results in setting **DASDSELECTIONSIZE**, but **DASD(TASKS)** and **TAPE(TASKS)** fail because the sum of both tasks exceeds 64.

During the automatic backup process, DFSMShsm recognizes two situations that involve backup and a data set in use. First, certain data sets exist that you do not expect to be open for update, but that might be in use when backup is attempted. For this situation, the **SETSYS BACKUP** parameter has an **INUSE** subparameter to control how DFSMShsm can try a second time, after a specified delay, to serialize on a data set before backing it up, and whether successful serialization for a retry is required or merely preferable. The backups that the **INUSE** subparameter allows can be placed temporarily on ML1 DASD.

In the example system, it is presumed that you want DFSMShsm to attempt a retry if needed, after a delay of 10 minutes, and to back up the data set even if it is still (or again) in use on the second try. The following command shows a data set that is used with the SERIALIZATION preferred method. The following command is added to the example system:

```
SETSYS BACKUP(INUSE(RETRY(Y) DELAY(10) SERIALIZATION(PREFERRED)))
```

For the second situation, certain data sets (typically a database) are expected to be open for update most of the time. You can use the data set backup installation exit ARCBDEXT (which is invoked at the data set backup level for **BACKDS** command) for additional control to direct DFSMShsm either not to serialize before making a backup or to back up the data set even if serialization fails. The **SETSYS INUSE** subparameters can also apply in this situation, but the exit can override them for a certain data set.

For more information about the installation exit, see *DFSMS Installation Exits,* SC26-7396.

Back up data set in a batch environment

DFSMShsm supplies a program to perform an *inline backup*. The program that you execute is called ARCINBAK. With ARCINBAK, you can back up data sets in the middle of a job by the addition of a new step to the job. In Example 10-1, the DD names to identify the data sets that you want backed up are identified as `BACK`*nn*. The data set names that are associated with these DDs will be backed up. A DD name prefix of anything other than `BACK` is not allowed.

Example 10-1 Sample ARCINBAK JCL

```
//JOBNAME JOB . . . ,USER=USERID,PASSWORD=USERPSWD
//STEP1 EXEC PGM=USERPGM
//SYSPRINT DD SYSOUT=A
//DSET1 DD DSN=USERID.NO3.GDG(-1),DISP=OLD
//DSET2 DD DSN=USERID.NO3.PSFB,DISP=OLD
//DSET3 DD DSN=USERID.NO4.PSFB,DISP=OLD
//DSET4 DD DSN=USERID.NO3.KSDS,DISP=OLD
/*
//STEP2 EXEC PGM=ARCINBAK
//ARCPRINT DD SYSOUT=A
//ARCSNAP DD SYSOUT=A
//* -----------------------------------------------------------
//* BACKUP OF GDG DATA SET SHOULD BE SUCCESSFUL.
//* -----------------------------------------------------------
//BACK01 DD DSN=*.STEP1.DSET1,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//BACK02 DD DSN=*.STEP1.DSET2,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----------------------------------------------------------
//BACK03 DD DSN=*.STEP1.DSET4,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF GDG DATA SET SHOULD BE SUCCESSFUL.
//* -----------------------------------------------------------
//BACK04 DD DSN=USERID.NO1.GDG.G0001V00,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----------------------------------------------------------
//BACK05 DD DSN=USERID.NO1.PSFB,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF UNCATALOGED DATA SET SHOULD FAIL.
//* -----------------------------------------------------------
//BACK06 DD DSN=USERID.NO2.UNCAT,VOL=SER=VOL003,UNIT=3390,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----------------------------------------------------------
//* -----------------------------------------------------------
//BACK07 DD DSN=USERID.NO1.KSDS,DISP=SHR
//* -----------------------------------------------------------
//* BACKUP OF OPEN IN-USE VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----------------------------------------------------------
```

```
//BACK08 DD DSN=USERID.NO2.KSDS,DISP=SHR
//* ---------------------------------------------------------------
//* BACKUP OF RACF PROTECTED NON-VSAM DATA SET
//* BY AN UNAUTHORIZED USER SHOULD FAIL.
//* ---------------------------------------------------------------
//BACK09 DD DSN=USERXX.NO2.PSFB,DISP=SHR
/*
```

Notes:

- Only cataloged data sets are supported for inline backup. If volume and unit information is specified on the DD statement, an uncataloged data set is assumed, and the data set is not processed.
- ARCINBAK does not support data sets that are allocated with any of the following dynamic allocation options, XTIOT, UCB NOCAPTURE, and DSAB above the line, except when the calling program supplies an open data control block (DCB).

10.1.3 Backup non-SMS-managed data sets

You can also use automatic backup for data sets that reside on non-SMS-managed volumes by using the **ADDVOL** command and defining it in ARCCMDxx. The syntax is shown:

```
ADDVOL volser UNIT(unittype) PRIMARY(AUTOBACKUP)
```

The primary differences in processing for SMS-managed storage and non-SMS-managed storage occur in the following processes:

- Uncataloged data sets can be backed up and recovered.
- Users can specify the volume to which a data set is to be recovered.
- Volumes to be dumped or backed up are individually identified to DFSMShsm.

Because storage groups or management classes are not available to define how to manage the volumes and which volumes to manage for non-SMS-managed storage, the following subtasks provide a way to manage non-SMS-managed storage:

- Defining how frequently data sets that are managed by a system will be backed up. The **FREQUENCY** parameter of the **SETSYS** command controls the backup frequency for the data sets that are processed by each DFSMShsm host. The command that is added to the ARCCMDxx member for each host is shown:

  ```
  SETSYS FREQUENCY(days)
  ```

- Defining how many versions to keep for each backed up data set. For non-SMS-managed storage, the number of backup versions to retain is a DFSMShsm host-wide specification.

  As with SMS-managed storage, depending on the record length that is used to define the backup control data set (BCDS), DFSMShsm can maintain up to 29, or up to 100, backup versions of any data set. Within that upper limit, the **VERSIONS** parameter of the **SETSYS** command controls the number of backup versions to retain. The command syntax to add to the ARCCMD*xx* member for each DFSMShsm host is shown:

  ```
  SETSYS VERSIONS(limit)
  ```

  If you use the **RETAINDAYS** keyword when you create data set backup copies, and active backup copies with unmet RETAINDAYS values are rolled off, DFSMShsm maintains them as retained backup copies. DFSMShsm can maintain more than enough retained copies for each data set to meet all expected requirements.

The **FREQUENCY** and **VERSION** parameters of the **SETSYS** command control how often data sets are backed up and how many versions are kept for all of the non-SMS-managed data sets.

You can control the frequency and versions for individual data sets with the **(H)ALTERDS** command. You can also control the frequency and the versions independently of each other.

The command syntax is shown:

(H)ALTERDS *dsname* FREQUENCY(*days*) VERSIONS(*limit*)

10.1.4 Backing up a data set manually and the RETAINDAYS keyword

If the data set's management class allows users to issue a backup command, users can back up their data sets whenever they want, regardless of the change bit status, by using the **(H)BACKDS** command. The **RETAINDAYS** keyword in the **(H)BACKDS** command allows users to specify the number of days that a backup version will be retained for cataloged data sets. This **RETAINDAYS** value is applied when a backup copy is rolled off.

The value of **RETAINDAYS** can be in the range of 0 - 50000, which corresponds to the maximum of 136 years. If you specify 99999, the data set backup version is treated as never expiring. Any value greater than 50000 (and other than 99999) causes a failure with an error message ARC1605I. A **RETAINDAYS** value of 0 indicates the following conditions:

- The backup version expires when the next backup copy is created.
- The backup version might expire within the same day that it was created if EXPIREBV processing takes place.
- The backup version is kept as an active copy before roll-off occurs.
- The backup version is not managed as a retained copy.

The command syntax for **BACKDS** with the **RETAINDAYS** keyword is shown:

(H)BACKDS *dsname* RETAINDAYS(*days*)

By using this feature, DFSMShsm backup copies are managed as either active copies or retained copies:

- Active copies are a set of backup copies that are not yet rolled off. Active copies are determined by either the SMS management class or SETSYS value. Depending on the record length that is used to define the BCDS, DFSMShsm can keep up to 29, or up to 100, versions of any data set.
- Retained copies are backup copies that rolled off from a set of active copies and did not reach their retention period. A data set can have an unlimited number of retained backup copies.

Backup versions roll-off process

The roll-off processing checks all of the active copies to see whether any active copy has specified retained days and if so, if the retained days were met:

- If more than one of the active backup copies met their retain days, the oldest active backup copy is rolled off, and the rest of the active backup copies are maintained as active copies even though their retain days were already met.
- If one or more of the active backup copies specified retain days, but none of them met their retain days, one of the following actions occurs:
  - If the oldest active backup copy has a RETAINDAYS value, it is rolled off and replaced by the new backup copy but it is still maintained as a retained backup copy.
  - If the oldest active backup copy does not have a RETAINDAYS value, it is rolled off as normal.
- If none of the active backup copies have RETAINDAYS values, roll off processing is processed as normal.

Roll-off example

By using the **HLIST DSNAME(MHLRES4.JCL.CNTL) BCDS TERMINAL** DFSMShsm command, the MHLRES4.JCL.CNTL data set shows five backup versions and the maximum number of active copies. The NUMBER of BACKUP VERS (Data Set Exists) field in ISMF is 5. See Example 10-2.

Example 10-2 Output from the HLIST command before the creation of another backup version

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T180320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:18  CAT=YES  GEN=000  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T140320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:14  CAT=YES  GEN=001  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T080320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:08  CAT=YES  GEN=002  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

  BDSN=HSM.BACK.T040320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:04  CAT=YES  GEN=003  VER=002  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T280220.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:02:28  CAT=YES  GEN=004  VER=001  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00001

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000078 SENT TO DFSMSHSM
 ***
```

The next day after the creation of the new backup version (VER=006), Example 10-3 shows that the backup copy version 001 (VER=001) is rolled off because its RETAINDAYS value is 1 day and it is no longer shown in the output.

Example 10-3 Output from HLIST command after the new backup version is created

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T222712.MHLRES4.JCL.A2234          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:27:22  CAT=YES  GEN=000  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T180320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:18  CAT=YES  GEN=001  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T140320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:14  CAT=YES  GEN=002  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T080320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:08  CAT=YES  GEN=003  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

  BDSN=HSM.BACK.T040320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:04  CAT=YES  GEN=004  VER=002  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000085 SENT TO DFSMSHSM
 ***
```

The next creation of a new backup version (VER=007) has backup copy version 2 (VER=002), which has no RETAINDAYS value, to be rolled off as shown in Example 10-4.

Example 10-4 Output from HLIST command

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T025012.MHLRES4.JCL.A2234          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:50:02  CAT=YES  GEN=001  VER=007  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T014812.MHLRES4.JCL.A2234          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:48:01  CAT=YES  GEN=002  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T180320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:18  CAT=YES  GEN=001  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T140320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
```

```
  BACKDATE=12/08/20  BACKTIME=20:03:14  CAT=YES  GEN=002  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T080320.MHLRES4.JCL.A2233         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:08  CAT=YES  GEN=003  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000085 SENT TO DFSMSHSM
 ***
```

The oldest active copy is VER=003 and its RETAINDAYS (RETDAYS=00005) values are not met. After the creation of a new backup version, VER=003 is rolled off but it is kept as the RETAINED COPIES version. This rolled-off backup version has no GEN and VER values anymore, as shown in Example 10-5. Also, the TOTAL BACKUP VERSIONS field is 6.

Example 10-5 Output from (H)LIST command

```
DSN=MHLRES4.JCL.CNTL                               BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T275912.MHLRES4.JCL.A2234         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:59:27  CAT=YES  GEN=000  VER=008  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T025012.MHLRES4.JCL.A2234         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:50:02  CAT=YES  GEN=001  VER=007  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T014812.MHLRES4.JCL.A2234         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:48:01  CAT=YES  GEN=002  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T180320.MHLRES4.JCL.A2233         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:18  CAT=YES  GEN=001  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T140320.MHLRES4.JCL.A2233         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:14  CAT=YES  GEN=002  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T080320.MHLRES4.JCL.A2233         BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:08  CAT=YES  GEN=*** VER=*** UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

TOTAL BACKUP VERSIONS = 0000000006

ARC0140I LIST COMPLETED,       14 LINE(S) OF DATA OUTPUT
***
```

A new **SELECT(ACTIVE/RETAINDAYS)** parameter was added to the DFSMShsm **(H)LIST** command to list either active copies or only retained copies as shown:

(H)LIST DSNAME(*dsn*) BOTH | BCDS SELECT(ACTIVE | RETAINDAYS)

The output of the **LIST DSNAME(MHLRES4.JCL.CNTL) BCDS SELECT(ACTIVE) TERMINAL** command is shown in Example 10-6.

Example 10-6 Output from (H)LIST command for ACTIVE copies

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T275912.MHLRES4.JCL.A2234          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:59:27  CAT=YES  GEN=000  VER=008  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T025012.MHLRES4.JCL.A2234          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:50:02  CAT=YES  GEN=001  VER=007  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T014812.MHLRES4.JCL.A2234          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/21  BACKTIME=12:48:01  CAT=YES  GEN=002  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T180320.MHLRES4.JCL.A22333          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20 BACKTIME=20:03:18 CAT=YES  GEN=003  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T140320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20  BACKTIME=20:03:14  CAT=YES  GEN=002  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

TOTAL BACKUP VERSIONS = 0000000005

ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
***
```

The output of the **LIST DSNAME(MHLRES4.JCL.CNTL) BCDS SELECT(RETAINDAYS) TERMINAL** command is shown in Example 10-7.

Example 10-7 Output from the (H)LIST command for RETAIN copies

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T180320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20 BACKTIME=20:03:18 CAT=YES  GEN=000  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T080320.MHLRES4.JCL.A2233          BACKVOL=SBXHS6   FRVOL=MHLS1A
  BACKDATE=12/08/20 BACKTIME=20:03:08 CAT=YES  GEN=***  VER=***  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

 TOTAL BACKUP VERSIONS = 0000000002
 ARC0140I LIST COMPLETED,       6 LINE(S) OF DATA OUTPUT
```

10.1.5 Using DSSXMMODE

The backup and migration throughput can be maximized by using the DSSXMMODE feature. In this mode, DFSMSdss is loaded in its own address space, or in the DFSMShsm address space, for backup, CDS backup, dump, migration, and full-volume and data set recovery functions. This parameter can be issued only from the ARCCMDxx PARMLIB member. This parameter does not affect the FRBACKUP and FRRECOV use of the DFSMSdss cross-memory interface.

The syntax command is shown:

```
SETSYS DSSXMMODE(Y|N BACKUP(Y|N) CDSBACKUP(Y|N) DUMP(Y|N) MIGRATION(Y|N)
RECOVER(Y|N))
```

The following parameters are optional parameters of the **DSSXMMODE** parameter:

N
DFSMSdss is not loaded into its own address space for the DFSMShsm function that is invoked. Instead, DFSMSdss is loaded in the DFSMShsm address space, which is the default.

Y
DFSMSdss is loaded into a new address space for the DFSMShsm function that is invoked and retained for future invocations of that function until DFSMShsm is shut down or restarted.

BACKUP
This value controls where DFSMSdss for the DFSMShsm backup function is loaded. It can be in its own address space (Y) or in the DFSMShsm address space (N). The default is N for no.

CDSBACKUP
This value controls where DFSMSdss for the DFSMShsm CDS backup function is loaded. It can be in its own address space (Y) or in the DFSMShsm address space (N). The default is N for no.

DUMP
This value controls where DFSMSdss for the DFSMShsm dump function is loaded. It can be in its own address space (Y) or in the DFSMShsm address space (N). The default is N for no.

MIGRATION
This value controls where DFSMSdss for the DFSMShsm migration function is loaded. It can be in its own address space (Y) or in the DFSMShsm address space (N). The default is N for no.

RECOVERY
This value controls where DFSMSdss for the DFSMShsm full-volume and data set recovery functions are loaded. They can be in their own address space (Y) or in the DFSMShsm address space (N). The default is N for no.

Table 10-1 on page 263 defines the DFSMSdss address space identifiers for address spaces that are started by DFSMShsm.

Table 10-1 DFSMSdss address space identifiers that are started by DFSMShsm functions

Function	Address space identifier
Backup	ARCnBKUP
CDS backup	ARCnCDSB
Dump	ARCnDUMP
Fast replication backup	DSSFRBxx
Fast replication recovery (copy pool and volume)	DSSFRRxx
Fast replication recovery (data set)	DSSFRDSR
Migration	ARCnMIGR
Recovery from backup (data set and full volume)	ARCnRCVR
Recovery from a dump tape (data set)	ARCnREST
Recovery from a dump tape (full volume)	ARCnRSTy

DFSMSdss requires that the ARC* and DFSSFR* address spaces run at the same or higher priority as DFSMShsm. You can create Workload Manager (WLM) profiles so that these address spaces are assigned to the correct service classes.

Notes:

- Each DFSMSdss address space requires the correct setup in the same manner as DFSMShsm started task control (STC).
- Use of the DFSMSdss cross-memory API for these functions might result in an increase in CPU time.
- When DFSMShsm starts one or more of these address spaces, the address spaces remain active until DFSMShsm is terminated. When DFSMShsm terminates, all of the started DFSMSdss address spaces automatically terminate.
- The **FRBACKUP** and **FRRECOV** commands always use DFSMSdss cross-memory support.
- The DFSMSdss address space identifiers that are started for the dump, data set recovery from dump, data set recovery from backup, migration, backup, and CDS backup functions are optional and are controlled by using the SETSYS DSSXMMODE.

10.1.6 Interrupting and restarting backup processing

To stop the first phase of automatic backup, which is the backup of the CDSs, you must use the **STOP** command.

You can use the following methods to interrupt automatic backup processing during its last three phases. The last three phases of automatic backup processing consist of the following functions:

- Movement of backup versions
- Backup of migrated data sets
- Backup of volumes that are managed by DFSMShsm that have the automatic backup attribute

Four methods exist that you can use to interrupt automatic backup processing during its last three phases:

- Holding backup processing: You can hold backup processing by specifying the **HOLD** command for **BACKUP** or **BACKUP(AUTO)**, or **ALL** parameters. If you hold backup processing with the **ENDOFDATASET** parameter while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after DFSMShsm finishes processing the current backup version.

  If you hold backup processing with the **ENDOFVOLUME** parameter while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after the current phase of automatic backup completes. If you hold backup processing with the **ENDOFVOLUME** parameter while DFSMShsm is backing up managed volumes, DFSMShsm continues processing the current volumes and does not process any new volumes. Otherwise, DFSMShsm finishes processing the current data set. DFSMShsm does not start any new volume backup tasks.

- Disabling backup processing: You can disable backup processing by specifying the **SETSYS NOBACKUP** command. If you disable backup processing while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after DFSMShsm finishes processing the current backup version. If you disable backup processing while DFSMShsm is backing up volumes that are managed by DFSMShsm, DFSMShsm finishes processing the current data set and does not start any new volume backup tasks.

- Placing DFSMShsm in emergency mode: You can place DFSMShsm in emergency mode by specifying the **SETSY EMERGENCY** command. If you place DFSMShsm in emergency mode while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after DFSMShsm finishes processing the current data set.

  If you are in a multiple DFSMShsm-host environment and enabled the secondary host promotion function, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you put in emergency mode. Those functions include the automatic movement of backup versions from ML1 to tape, and the automatic backup of migrated data sets. If promotion occurs during an automatic backup window (as defined on the primary host), and it is not past the latest start time, the promoted host takes over where the primary host left off.

  If the promoted host was an automatic backup host before being promoted, it performs backups of DFSMShsm-owned volumes after performing the automatic movement of backup versions from ML1 to tape, and the automatic backup of migrated data sets. If the promoted host was not an automatic backup host before being promoted, it does not back up DFSMShsm-owned volumes.

  If you place DFSMShsm in emergency mode while DFSMShsm is backing up volumes that are managed by DFSMShsm, DFSMShsm finishes processing the current data set before ending the volume backup tasks that are in progress and it does not start any new volume backup tasks.

- Stopping DFSMShsm: You can stop DFSMShsm by entering the MVS or DFSMShsm **STOP** command. If you shut down DFSMShsm while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after DFSMShsm finishes processing the current data set.

  If you shut down DFSMShsm while DFSMShsm is backing up volumes that are managed by DFSMShsm, DFSMShsm finishes processing the current data set before ending the volume backup tasks that are in progress and does not start any new volume backup tasks.

  If you issue the DFSMShsm **STOP** command while DFSMShsm is backing up the CDSs or the journal data set, DFSMShsm finishes processing the current data set and does not back up any other CDSs or the journal data set.

If you are in a multiple DFSMShsm-host environment and you enabled secondary host promotion and issued a **STOP PROMOTE** command, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you stopped. Those functions include the automatic movement of backup versions from ML1 to tape, and the automatic backup of migrated data sets.

If promotion occurs during an automatic backup window (as defined on the primary host), and it is not past the latest start time, the promoted host takes over where the primary host left off. If the promoted host was an automatic backup host before being promoted, it performs backups of DFSMShsm-owned volumes after performing the automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If the promoted host was not an automatic backup host before being promoted, it does not back up DFSMShsm-owned volumes.

You can restart automatic backup processing by entering the corresponding commands if certain conditions are met:

- RELEASE BACKUP or RELEASE ALL
- SETSYS BACKUP
- SETSYS NOEMERGENCY
- START DFSMShsm (MVS command)

If secondary host promotion did not continue backup processing where the primary DFSMShsm host left off, the automatic backup restarts at the point of interruption if the date of the defined start time for the current automatic backup window is the same as the date that automatic backup last started from the beginning (same cycle date). When automatic backup restarts, it uses the same set of daily backup volumes it was using when it first started. Those daily backup volumes are the volumes that are assigned to the day in the backup cycle that was in effect when the automatic backup started.

After you restart automatic backup processing, DFSMShsm continues with the next volume that is managed by DFSMShsm under automatic backup control. However, a few considerations exist:

- If you issued a DFSMShsm **STOP** command and then issued a **START** DFSMShsm command, DFSMShsm can start automatic backup at the point of interruption if the following conditions are met:
  - If you add the primary volumes under automatic backup control in the same order in each startup of DFSMShsm
  - If you do not change the volumes that are associated with the storage groups
- If you interrupted a **BACKVOL PRIMARY** command, DFSMShsm can resume backup at the start of the next volume only if you issued HOLD BACKUP or SETSYS NOBACKUP. You must issue RELEASE BACKUP, RELEASE ALL, or SETSYS BACKUP during the same startup of DFSMShsm if you want the **BACKVOL PRIMARY** command to resume processing.
- DFSMShsm does not restart a **BACKVOL** command that is issued for one volume.

10.2 Dump availability management

The process of copying all data from a DASD volume to a tape volume is called *volume dump*. The dump of the entire volume is performed by DFSMShsm invoking DFSMSdss through an application interface. Volumes that are not in storage groups or not set as ADDVOL to DFSMShsm are dumped only by command.

10.2.1 Setting up automatic full volume dump in the DFSMShsm environment

You must define the full volume dump window and the full volume dump cycle for DFSMShsm to perform the auto dump process. You specify the **SETSYS AUTODUMPSTART** and **DEFINE** commands in the ARCCMDxx PARMLIB, as shown:

```
SETSYS AUTODUMPSTART(StartTime LatestStartTime QuiesceTime)
```

The command syntax is defined:

- *StartTime*: Automatic dump start time.
- *LatestStartTime*: Start automatic dump no later than the specified time.
- *QuiesceTime*: Additional volumes for full volume dump are not allowed after this time.

If you omit the quiesce time, auto dump processes all eligible volumes.

DFSMShsm always allocates the source and dump volumes for the full-volume dump before it invokes DFSMSdss. DFSMSdss performs only volume-level serialization during full-volume dump processing. Because DFSMShsm does not perform any data set serialization during full-volume dump processing, activity against a volume that is being dumped needs to be kept at an absolute minimum during the full-volume dump process, or errors occur.

You also need to define the backup cycle and the date that it will start the backup process. This command is also in ARCCMDxx and the syntax is shown:

```
DEFINE DUMPCYCLE(NNNNNNY CYCLESTARTDATE(2012/08/06))
```

The preceding command establishes a seven-day dump cycle that starts on Monday, 6 August 2012. Automatic dump will run on Sunday only. Specifying CYCLESTARTDATE means that the cycle stays the same through each initialization of DFSMShsm. We recommend that you code your DEFINE similar to the example. You might find that it is easier to work with a seven-day cycle than a long string of Ys and Ns.

Notes:

- The date that is specified for the cycle start date cannot be a date in the future. The date must always fall on or before the date that the **DEFINE** command is issued.
- When you redefine the dump class, be careful if you use the **DAY** parameter to define the days that a certain dump class is active. The day that is specified with the **DAY** parameter needs to be a `Y` day in the DUMPCYCLE. If the day that is specified is an `N` day in the DUMPCYCLE, automatic dump processing does not target the dump class.
- Dump cleanups run on `N` days in the dump cycle. Cleanup functions include the expiration of expired dump copies and the deletion of excess dump VTOC copy data sets.

DFSMShsm can run up to 64 concurrent dump tasks for each host, which does not limit the number of dump copies because each dump task can create up to five dump copies. The syntax for this command is shown. You need to add it to your ARCCMDxx member.

```
SETSYS MAXDUMPTASKS(nn)
```

You must consider the number of tape units that you expect to have available during automatic dump processing. To determine the number of necessary tape drives, multiply the number of dump classes for each volume by the number of dump tasks. If this number exceeds the number of tape drives that you expect to have available, you must lower your **MAXDUMPTASKS** parameter. If this number is under the number of tape drives that you expect to have available, you can raise your **MAXDUMPTASKS** parameter.

You also need to specify the DFSMSdss DASD I/O buffering technique to use for dump. With a single START I/O instruction, DFSMSdss can read one, two, or five tracks at a time, or a complete cylinder from the DASD that is being dumped. The syntax to set up the I/O buffer is shown:

SETSYS DUMPIO(*n*,*m*)

The *n* variable indicates the DFSMSdss DASD I/O buffering technique for the physical volume dump (by using the **BACKVOL** command with the **DUMP** parameter or during automatic dump). The *m* variable indicates the value that is used for the DFSMSdss logical dump (with DFSMSdss specified as the data mover on the **SETSYS** command).

The values that are used for the *n* and *m* variables are shown in Table 10-2.

Table 10-2 SETSYS DUMPIO command parameter values

Value	Meaning
1	DFSMSdss reads one track at a time
2	DFSMSdss reads two tracks at a time
3	DFSMSdss reads five tracks at a time
4	DFSMSdss reads one cylinder at a time

If you specify DUMPIO without a value for the *m* variable, the *m* variable defaults to the value that is specified for the *n* variable. If you do not specify the **DUMPIO** parameter on any **SETSYS** command, the DFSMShsm default for the *n* variable is 1 and for the *m* variable is 4.

The full volume dump process for each volume is based on information from DUMPCLASS, which is a predefined DFSMShsm-named set of characteristics that describes how volume dumps are managed. At least one DUMPCLASS must be defined in ARCCMDxx so that automatic dump uses that information in the full volume dump process. Each storage group that is eligible for full volume dump is assigned with a valid DUMPCLASS and this DUMPCLASS is passed to all volumes that belong to that storage group.

DUMPCLASS is created by using the DFSMShsm **DEFINE DUMPCLASS** command in ARCCMDxx. Because many considerations exist, we briefly explain what you can achieve with the dump class definition, the parameters that you can specify, and the DFSMShsm default (highlighted in bold) on the **DEFINE DUMPCLASS** command.

We show a sample definition that we use throughout this book.

For our examples, we define a dump class called *ONSITE*. With the **DEFINE DUMPCLASS** command, you can specify the following definitions:

- You define the name of the DUMPCLASS:

  DEFINE DUMPCLASS(ONSITE)

- You define whether dump volumes are available for reuse when they are invalidated:

  DEFINE DUMPCLASS(ONSITE AUTOREUSE | **NOAUTOREUSE**)

- You define whether data set restore is allowed from a full-volume dump copy:

  DEFINE DUMPCLASS(ONSITE DATASETRESTORE | **NODATASETRESTORE**)

- You define that a dump will be taken only on a particular day in the dump cycle:

  `DEFINE DUMPCLASS(ONSITE DAY(`*`day`*`))`

  If DAY is not specified, the **FREQUENCY** parameter is met, and it is a `Y` day in the dump cycle, the volume is dumped.

- You define the minimum number of days between volume dumps to a class:

  `DEFINE DUMPCLASS(ONSITE FREQUENCY(`*`days`*` | `**`7`**`))`

- You define whether DFSMSdss resets the “data set changed” indicator for each data set after a full-volume dump:

  `DEFINE DUMPCLASS(ONSITE RESET | `**`NORESET`**`)`

  Do not use **RESET** if you want incremental backup to make a backup version. **RESET** is not used if the DASD volume is also managed by DFSMShsm backup.

- You define how long you want to keep this full volume dump version:

  `DEFINE DUMPCLASS(ONSITE RETENTIONPERIOD(`*`days`*` | `**`NOLIMIT`**`))`

- You define the number of dump copies that DFSMShsm places on a dump volume that is assigned during one invocation of automatic dump:

  `DEFINE DUMPCLASS(ONSITE STACK(`*`nn`*`))`

  The *nn* variable is a value `1` - `255`.

  If you plan to use full volume dump with the stacking feature, it is better to group all volumes with the same volume capacity in the same storage group and assign DUMPCLASS with the appropriate stack number.

  For example, if you set up a pool of 100 volumes to have full volume dump, which consists of 80 model 9 volumes and 20 model 27 volumes, the dumpclass is defined with DUMPSTACK=`20` and also MAXDUMPTASKS is `5`. DFSMShsm possibly will assign all or most of 20 volumes of model 27 into one dump task and the others are divided evenly to the remaining four dump tasks. As a result, it takes a longer time to finish the dump for all 20 model 27 volumes. Therefore, auto dump might not complete within its defined window.

  Alternatively, you can put all of the 80 model 9 volumes into one storage group and assign DUMPCLASS=`FDVMOD9` with DUMPSTACK=`20` and put all of the 20 model 27 volumes into another storage group and assign DUMPCLASS=`FDVMOD27` with DUMPSTACK=`5`. Then, each of the dump tasks either gets 20 model 9 volumes or 5 model 27 volumes. Therefore, the dump window is shorter.

- You define the type of tape unit to use for the dump tapes:

  `DEFINE DUMPCLASS(ONSITE UNIT(`*`unittype`*`))`

- You can use a generic name or an esoteric name that is defined with the **SETSYS USERUNITTABLE** command in place of the unit type.

- You define the number of generations for which copies of the VTOC of dumped volumes will be kept:

  `DEFINE DUMPCLASS(ONSITE VTOCCOPIES(`*`copies`*` | `**`2`**`))`

From the **DEFINE DUMPCLASS** parameters, we can define an example **DUMPCLASS** name *ONSITE* in one command:

```
DEFINE DUMPCLASS(ONSITE UNIT(3590-1)
- RETPD(15) AUTOREUSE STACK(13)
- DATASETRESTORE VTOCCOPIES(4) - DAY(7))
```

If Concurrent Copy capability is set up and you want to use it for volume dumps, use this command syntax to make DFSMSdss use Concurrent Copy when it performs full-volume dumps:

```
SETSYS VOLUMEDUMP(CC)
```

Contention can occur between DFSMShsm full-volume dump and DFP scratch, rename, and allocate services. Major resource name ADRLOCK and minor resource name NONSPEC are allocated as exclusive (EXC) when the volume is not known.

Because this process can affect dump performance, you can bypass it by using the patch:

```
PATCH .MCVT.+38F X'10' VERIFY(.MCVT.+38F X'00')
```

10.2.2 Setting up automatic full-volume dump at the volume level

For SMS-managed volumes, only volumes that belong to a storage group with an AUTO DUMP value of `Y` and that are assigned a valid DUMPCLASS are eligible for full-volume dumps.

Figure 10-4 shows the Pool Storage Group Alter panel to specify whether you want to perform an automatic dump. Entering `Y` tells DFSMShsm to perform an automatic dump for all volumes in the storage group. You assign the dump class on this panel.

```
                           POOL STORAGE GROUP ALTER         Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : SGTEST
To ALTER Storage Group, Specify:
 Description ==>
             ==>
 Auto Migrate . . N  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . N  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . Y  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . . . .
                                      Copy Pool Backup SG Name . . .

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

 ALTER     SMS Storage Group Status . . . N   (Y or N)
 ALTER     SMA Attributes . . . . . . . . N   (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-4 Pool Storage Group Alter panel for auto dump and assigning dump class

For a non-SMS-managed volume, you can use the **ADDVOL** command:

ADDVOL *volser* UNIT(*unittype*) PRIMARY(AUTODUMP (*dumpclass*))

If your DFSMShsm environment consists of more than one DFSMShsm host, you can choose the system to run the full-volume dump process by entering the system name in the Dump Sys/Sys Group Name field in Figure 10-4.

10.2.3 Dumping volumes by command

Issue one of the commands that are shown in Example 10-8 to cause a command dump of one or more volumes.

Example 10-8 Dumping volumes by command

```
BACKVOL VOLUMES(volser,...,volser) DUMP

BACKVOL VOLUMES(volser,...,volser) -
DUMP(DUMPCLASS(class,class,class,class,class))

BACKVOL VOLUMES(volser,...,volser) -
DUMP(DUMPCLASS(class, . . .,class) -
RETENTIONPERIOD(days|*|NOLIMIT, . . .,days|*|NOLIMIT))
```

To dump all of the volumes in one or more storage groups, use the following command:

```
BACKVOL STORAGEGROUP(sgname) DUMP(DUMPCLASS(class) STACK(nn))
```

10.2.4 Interrupting and restarting automatic dump processing

You can use the following methods to interrupt automatic dump processing during the three phases of automatic dump:

- Automatic deletion of expired dump copies
- Automatic dump of volumes that are managed or owned by DFSMShsm
- Deletion of excess VTOC copy data sets

You can use the following four methods to interrupt automatic dump processing:

- Hold dump processing: You can hold dump processing by specifying the **HOLD** command for the **DUMP**, **DUMP(AUTO)**, or **ALL** parameter. If you hold dump processing with the **ENDOFDATASET** parameter while DFSMShsm is automatically deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMShsm finishes processing the current dump generation.

  If you hold dump processing with the **ENDOFVOLUME** parameter while DFSMShsm is automatically deleting expired dump copies or excess VTOC copies, processing ends after the current phase of automatic dump completes. If you hold dump processing with the **ENDOFVOLUME** parameter, DFSMShsm continues processing the current volumes and does not process any new volumes. Otherwise, DFSMShsm dump processing ends the next time that DFSMSdss reads or writes a record. DFSMShsm does not start any new volume dump tasks.

- Disable dump processing: You can disable dump processing by specifying the **SETSYS NOBACKUP** command. If you disable dump processing while DFSMShsm is automatically deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMShsm finishes processing the current dump generation. If you disable dump processing while DFSMShsm is dumping volumes that are managed by DFSMShsm, DFSMShsm dump processing ends the next time that DFSMSdss reads or writes a record. DFSMShsm does not start any new volume dump tasks.

- Place DFSMShsm in emergency mode: You can place DFSMShsm in emergency mode by specifying the **SETSYS EMERGENCY** command. If you place DFSMShsm in emergency mode while DFSMShsm is automatically deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMShsm finishes processing the current dump generation.

If you are in a multiple DFSMShsm-host environment and enabled the secondary host promotion function, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you put in emergency mode. Those functions include automatic deletion of expired dump copies and excess dump VTOC copy data sets. If promotion occurs during an automatic dump window (as defined on the primary host), and it is not past the latest start time, the promoted host restarts automatic dump processing.

If you place DFSMShsm in emergency mode while DFSMShsm is dumping volumes that are managed by DFSMShsm, DFSMShsm dump processing ends the next time that DFSMSdss reads or writes a record. DFSMShsm does not start any new volume dump tasks.

- Stop DFSMShsm processing: You can stop DFSMShsm by entering the MVS or DFSMShsm **STOP** command. If you shut down DFSMShsm while DFSMShsm is deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMShsm finishes processing the current dump generation. If you shut down DFSMShsm while DFSMShsm is dumping volumes that are managed by DFSMShsm, DFSMShsm dump processing ends the next time that DFSMSdss reads or writes a record. DFSMShsm does not start any new volume dump tasks.

  If you are in a multiple DFSMShsm-host environment, enabled secondary host promotion, and issued a **STOP PROMOTE** command, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you stopped. Those functions include the automatic movement of backup versions from ML1 to tape, and the automatic backup of migrated data sets.

  If promotion occurs during an automatic backup window (as defined on the primary host), and it is not past the latest start time, the promoted host takes over where the primary host left off. If the promoted host was an automatic backup host before it was promoted, it performs backups of DFSMShsm-owned volumes after it performs the automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If the promoted host was not an automatic backup host before it was promoted, it does not back up DFSMShsm-owned volumes.

You can restart automatic dump processing by entering the corresponding commands:

- RELEASE DUMP or RELEASE ALL
- SETSYS BACKUP
- SETSYS NOEMERGENCY
- START DFSMShsm (MVS command)

After you restart automatic dump processing, DFSMShsm continues with the next volume that is managed by DFSMShsm under automatic dump control. The automatic dump restarts at the point of interruption if less than 24 hours passed since automatic dump was last started from the beginning and if the time is still within the dump start window.

10.3 Recovering data from the DFSMShsm backup

The recovery and restore process, which can be considered the opposite of backup and dump, is not an automatic process. DFSMShsm will not automatically recover or restore data if it becomes damaged. The recover and restore process is driven by commands. You use the recovery and restore process to perform the following tasks:

- Recover a data set that was lost or damaged
- Access an earlier version of the data set without deleting the current version

The recovery process has two restrictions:

- DFSMShsm cannot recover a data set to a migration volume.
- DFSMShsm cannot recover a data set that is marked as migrated in the computing system catalog, *unless* the data set is non-VSAM, a recover data set name is issued with NEWNAME specified, and the NEWNAME data set is not a migrated data set.

In this topic, we show how to perform the following functions:

- Recover the most recent version of a data set or dump volume
- Restore a volume from a dump copy and update it from incremental backup versions
- Restore a volume from a full-volume dump copy
- Recover a volume from DFSMShsm backup versions

You need to determine the maximum number of recovery tasks that your systems can afford, based on the amount of disk storage and the number of available tape drive resources. The actual number of effective tasks might be limited if the number of backup resources is less than the total tasks specified. The command syntax to define this parameter is shown:

`MAXDSRECOVERTASKS(`*nn*`)`

The *nn* variable is a value `0` - `64`.

The value that you specify for **MAXDSRECOVERTASKS** applies to data set recovery tasks for both DASD and tape. To limit tape requests to a subset of this value, use the keyword **MAXDSTAPERECOVERTASKS** instead. The maximum number of tasks from DASD is the value of **MAXDSRECOVERTASKS** minus the current number of tape tasks that are specified by **MAXDSTAPERECOVERTASKS**. If no active tape tasks exist, the number of disk tasks can reach the value that is specified by **MAXDSRECOVERTASKS**. To prevent tape tasks, specify **SETSYS MAXDSTAPERECOVERTASKS(0)**.

10.3.1 Recover a data set from a backup

The data set recovery function refers to the process of recovering a data set to its condition as of a specified date. You can recover individual data sets by entering an HRECOVER line operator on an ISMF panel or by issuing the **HRECOVER** command. DFSMShsm can recover data sets from a DFSMShsm backup version or from a DFSMShsm dump copy. DFSMShsm automatically chooses the most recent copy of the data set unless directed otherwise by options you specify with the **HRECOVER** command.

If the data set is SMS-managed at the time of recovery, the target volume is determined by the data set's storage class and storage group. If the data set is not SMS-managed, the target volume is selected in the following order:

- The specified target volume
- The volume on which the target data set is cataloged
- The volume from which the data set was originally backed up or dumped

There are two ways to recover a data set:

- Recover by using the ISMF panel
- Recover by using the TSO **(H)RECOVER** command

Recover data set by using the ISMF panel

The following steps present an example of how to use the HRECOVER line operator to recover a cataloged data set. In our ISMF panel example, we used MHLRESA.REPORT.JCL as a sample data set name.

From option 1 of the ISMF panel, we generate a list of data set patterns of MHLRESA.** as shown in Figure 10-5.

```
                       DATA SET SELECTION ENTRY PANEL             Page 1 of 5
Command ===>

For a Data Set List, Select Source of Generated List  . . 2  (1 or 2)

  1  Generate from a Saved List          Query Name To
       List Name  . .                    Save or Retrieve

  2  Generate a new list from criteria below
       Data Set Name . . . 'MHLRESA.**'
       Enter "/" to select option       Generate Exclusive list
       Specify Source of the new list  . . 2    (1 - VTOC, 2 - Catalog)
       1 Generate list from VTOC
          Volume Serial Number . . .            (fully or partially specified)
          Storage Group Name . . . .            (fully specified)
       2 Generate list from Catalog
          Catalog Name . . .
          Volume Serial Number . . .            (fully or partially specified)
          Acquire Data from Volume . . . . . . . N  (Y or N)
          Acquire Data if DFSMShsm Migrated  . . N  (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-5 ISMF data set selection panel

Enter the `HRECOVER` line operator in the line operator column next to MHLRESA.REPORT.JCL, as described in Figure 10-6 on page 274.

```
                              DATA SET LIST
Command ===>                                                Scroll ===> HALF
                                                      Entries 1-15 of 15
Enter Line Operators below:                           Data Columns 3-5 of 42

     LINE                                         ALLOC      ALLOC     % NOT
     OPERATOR         DATA SET NAME               SPACE      USED      USED
    ---(1)----  ------------(2)------------ ---(3)---  ---(4)---  -(5)-
                MHLRESA                     ---------  ---------    ---
                MHLRESA.ABARS               ---------  ---------    ---
                MHLRESA.ABARS.INSTRUCT      ---------  ---------    ---
                MHLRESA.ABARS.OUTPUT        ---------  ---------    ---
                MHLRESA.ABARS.SELECT        ---------  ---------    ---
                MHLRESA.AUDIT.MCDS          ---------  ---------    ---
                MHLRESA.BRODCAST            ---------  ---------    ---
                MHLRESA.LOG.MISC            ---------  ---------    ---
    HRECOVER    MHLRESA.REPORT.JCL          ---------  ---------    ---
                MHLRESA.REPORT.LIB          ---------  ---------    ---
                MHLRESA.RMMREP.XMIT         ---------  ---------    ---
                MHLRESA.SC64.ISPF42.ISPPROF ---------  ---------    ---
                MHLRESA.SC64.SPFLOG1.LIST   ---------  ---------    ---
                MHLRESA.SC70.ISPF42.ISPPROF ---------  ---------    ---
                MHLRESA.VALIDATE            ---------  ---------    ---
    ----------  ------  -----------  BOTTOM OF  DATA  -----------  ------  ----
```

Figure 10-6 ISMF Data Set List panel

Pressing Enter opens the next panel, which shows all available backup versions of data set MHLRESA.REPORT.JCL, as shown in Figure 10-7.

```
                              HRECOVER ENTRY PANEL               Page 1 of 6
Command ===>

Specify Y to recover a Backup Version for
 Data Set: MHLRESA.REPORT.JCL
                               Recover                                  Recover
Version  Date    TIME   Gen#   (Y/N)  |  Version  Date    TIME   Gen#   (Y/N)
--------------------------------------|---------------------------------------
008     2012237  141952  00       y   |  007     2012234  162156  01       N
006     2012234  134813  02       N   |  005     2012233  193354  03       N
004     2012233  193350  04       N   |
                                      |

Use ENTER to Continue; Use DOWN to select other versions;
Use HELP Command for Help; Use END Command to Cancel the HRecover.
```

Figure 10-7 HRECOVER

In Figure 10-7, we chose to recover from backup version 008. Therefore, we enter `Y` in the RECOVER column.

With the next panel, you can specify several options, including whether you want to recover the data set to a new name or you want to replace the current version on disk by a backup version. We chose to replace the current version on disk by the most recent backup version.

Therefore, we entered `Y`, as shown in Figure 10-8. Press Enter to perform the data set recovery.

```
                            HRECOVER ENTRY PANEL                  Page 6 of 6
Command ===>

Optionally specify one or more for
 Data Set: MHLRESA.REPORT.JCL
 For the Backup Data Set, Specify:
   Backup Generation Number . . 00          (0 to 99)
   Backup Date  . . . . . . . .             (yyyy/mm/dd)
   Backup Time  . . . . . . . .             (hhmmss)
   Data Set Password  . . . . .             (if password protected)
   Note: Dataset Password ignored when in ADMIN mode.
 To Recover to a specific volume, Specify:
   Volume Serial Number . . . .             (target volume)
   Device Type  . . . . . . . .             (target device type)
 To rename recovered Data Set, Specify:
   New Data Set Name  . . . . . 'MHLRESA.REPORT.JCL'
   Password of New Data Set . .             (if password protected)
 Replace existing Data Set  . . Y           (Y or N)
 DA Access Option . . . . . . .             (1, 2 or 3)
 Wait for completion  . . . . . N           (Y or N)
Use ENTER to perform HRecover;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-8 HRECOVER ENTRY PANEL

Recover by using the TSO (H)RECOVER command

Another way to recover a data set is to issue the TSO **(H)RECOVER** command. If no data set of the same name exists in the system catalog, you can use the following command:

H)RECOVER *dsname*

The *dsname* is the name of the data set that you want to recover. DFSMShsm chooses the most recently created version of the data set for you. You do not have to identify where the most recent copy is yourself before you put the command together.

The following command recovers a specific data set but gives it a new name:

(H)RECOVER *dsname* NEWNAME(*newdsname*)

The *dsname* is the name of the data set that you want to recover and *newdsname* is the new name for the data set that you want to recover.

You can replace an existing, cataloged data set with the recovered version by using the following command:

(H)RECOVER *dsname* REPLACE

If a backup exists of a data set in full-volume dump version, you can recover it by using the following command to restore the latest dump copy from available dump volumes. If the data set exists, you add the **REPLACE** optional parameter to restore the data set successfully.

(H)RECOVER *dsname* FROMDUMP

You can issue the following additional commands to recover a non-SMS-managed data set:

- To recover an uncataloged data set to the volume from which it was backed up, execute the following command:

  `(H)RECOVER` *`dsname`* `FROMVOLUME(`*`volser`*`)`

- You can also add the **`REPLACE`** parameter to the command to replace the existing uncataloged data set of the same name on that volume. Or, add the **`NEWNAME(`*`newdsname`*`)`** parameter to give it a new name.
- To recover a data set to a specified volume that is different from the original volume, execute the following command:

  `(H)RECOVER` *`dsname`* `TOVOLUME(`*`volser`*`) - UNIT(`*`unittype`*`)`

  You can also use the preceding command to recover an SMS-managed data set to a non-SMS-managed volume by adding **`FORCENONSMS`** parameter.
- To recover a data set to the specified volume and give the data set a new name, execute the following command:

  `(H)RECOVER` *`dsname`* `NEWNAME(`*`newdsname`*`) - TOVOLUME(`*`volser`*`) UNIT(`*`unittype`*`)`

  You can also use the preceding command to recover an SMS-managed data set to a non-SMS-managed volume by adding **`FORCENONSMS`** parameter.

10.3.2 Restoring a volume from a full volume dump

The methods to restore volumes from a full volume restore are described.

Full-volume restore with incremental backup

When you recover a volume with full-volume restore and incremental backup versions, one of your users might use a data set in the interval between the restore and the incremental recovery. Avoid this situation by putting the SMS-managed volume in DISALL status. You need to be aware of two considerations before you place the volume in DISALL status:

- With the DISALL status, data sets might be recovered (as directed by your automatic class selection (ACS) routines) to different volumes in the storage group that contain the volume that is being recovered.
- If you are recovering a catalog, or a volume that belongs to a storage group that uses the guaranteed space attribute, the volume must be in the enabled (ENABLE) status during the recovery process.

Ensure that the SMS-managed volume is in the disabled (DISALL) status before you issue the **`RECOVER`** command, unless you are recovering a catalog, or a volume from a storage group that uses the guaranteed space attribute.

After the recovery process completes, be sure to review the messages for failures or data sets that were not recovered. For example, if the volume that you are recovering contains part of a multivolume data set, that partial data set is not recovered. After the recovery process completes, the data set is listed as not being recovered, which gives you the opportunity to recover the entire data set, including the parts of the data set that reside on volumes that are not being recovered. You can issue individual **`RECOVER`** ***`dsn`*** commands for these data sets.

Example 10-9 shows the commands that you can use to combine a full-volume restore with a full-volume recovery.

Example 10-9 Commands to combine a full-volume restore with a full-volume recovery

```
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPVOLUME(tape_volser) APPLYINCREMENTAL)

RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPCLASS(class) APPLYINCREMENTAL)

RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPGENERATION(dgennum) APPLYINCREMENTAL)
```

When **APPLYINCREMENTAL** is specified on a **RECOVER FROMDUMP** command, DFSMShsm first performs a full-volume restore. After the volume restore completes, DFSMShsm performs the following steps:

- If ICF catalogs reside on the volume, DFSMShsm recovers any catalog for which a backup version exists that was created following the dump. All of these catalogs must be successfully recovered for the APPLYINCREMENTAL process to continue. The recovery of the catalog fails and the APPLYINCREMENTAL process will not continue for either or both of the following conditions:
  - If any catalog was backed up by DFSMShsm operating with a version of MVS/DFP that is earlier than version 2 release 3.0
  - If any catalog was dumped before it was exported or backed up by DFSMShsm, which uses export
- DFSMShsm creates a list of candidate data sets from either or both the dump and backup VTOC copy data sets to control the volume recovery process.
- DFSMShsm applies incremental data set backups from the list of candidate data sets with the following exceptions:
  - Catalogs.
  - Data sets that are cataloged as multiple-volume, including multiple-stripe data sets. If the target volume is SMS-managed, the data set is not currently cataloged on the volume that is being processed, and the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
  - VSAM data sets that are currently cataloged as MIGRAT. Cluster, index, and data components are scratched from the restored volume.
  - Non-VSAM data sets that are currently cataloged as MIGRAT but the dump VTOC copy data set is more recent than the backup VTOC copy data set. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
  - VSAM volume data sets (VVDSs).
  - VTOC index data sets.
  - Data sets that are not on the backup VTOC copy data set but the backup VTOC copy data set is more recent than the dump VTOC copy data set. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
  - Currently uncataloged VSAM data sets. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
  - Currently uncataloged non-VSAM data sets that are targeted to an SMS-managed volume. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.

– ICF VSAM component data sets.
– Data sets that are currently cataloged as MIGRAT, that were cataloged when they were backed up, and that are selected for recovery. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
– Data sets that are currently cataloged as MIGRAT, that were uncataloged when they were backed up, that are selected for recovery, and that were migrated after the backup version of the uncataloged data set was made from the same volume. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
– Data sets that are currently cataloged on another volume but the selected data set was cataloged when it was backed up. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
– Data sets that were last backed up when they were cataloged, but a backup version of a data set that was uncataloged when it was backed up is selected for recovery. If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
– Data sets that are cataloged on the non-SMS-managed target volume but the backup version was made when the data set was uncataloged. (If the data set was uncataloged when it was backed up from the target volume and is now cataloged on the target volume, and the volume is now SMS-managed, recovery is allowed.) If the data set was restored to the volume, DFSMShsm scratches it from the restored volume.
– Data sets that are not SMS-managed that are being recovered to SMS-managed volumes.

Full-volume restore from a dump copy

Full-volume restore is the process of recovering the entire physical contents of a volume. Full-volume restore, an alternative to the DFSMShsm volume recover function, invokes Data Facility Storage Management Subsystem Data Set Services (DFSMSdss) to perform full-volume restores.

DFSMShsm uses the dump volumes of a dump copy as input to a DFSMSdss full-volume restore request. If the dump copy starts in file two or beyond on a stacked dump tape, DFSMShsm supplies the file block ID for high-speed positioning to the start of the dump copy. Example 10-10 shows the commands that can be used to request a full-volume restore.

Example 10-10 Commands to request a full-volume dump

```
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP

RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP DATE(date)

RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPVOLUME(tape_volser))

RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPCLASS(class))
```

```
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPCLASS(class)) DATE(yyyy/mm/dd)

RECOVER * TOVOLUME(original_volser) UNIT(unittype) -
FROMDUMP(DUMPGENERATION(dgennum)) -

TARGETVOLUME(spare_volser)
```

10.3.3 Volume recovery from incremental backup versions

Normally, volume recovery is run when an entire DASD pack is lost or damaged or a significant amount of the data is inaccessible. A copy of the VTOC that was made during the incremental backup of the volume is used to drive the individual incremental recovery of data sets for which DFSMShsm has backup versions. DFSMShsm volume recovery is also referred to as *incremental volume recovery*. After DFSMShsm successfully recovers a volume, each supported data set on the recovered volume is at the level of its latest backup. This process recovers the data set to the most recent level unless someone changed the data set since the last time DFSMShsm backed it up.

The following commands cause volume recovery from incremental backup copies, as shown in Example 10-11.

Example 10-11 Volume recovery from incremental backup copies

```
RECOVER * TOVOLUME(volser) UNIT(unittype)
RECOVER * TOVOLUME(volser) UNIT(unittype) DATE(date)
```

Volume recovery is similar to data set recovery in that DFSMShsm recovers each data set; DFSMShsm does not recover data track by track. First, a single task builds the queue of data sets that need to be recovered, then multitasking is used during the actual recovery of the data sets. If recovering multiple volumes, it is most efficient to first put a hold on tape data set recovers, create the queue of data sets to be recovered, and then release the tape data set recovery processing.

Because data sets from different volumes might reside on one tape, recovering data sets for multiple volumes at the same time, rather than volume-by-volume, reduces the number of required tape mounts, and therefore speeds processing. However, because DFSMShsm recovers each data set and because the backup operation uses backup volumes from different days, volume recovery can require access to many backup volumes only to recover one level 0 volume. As a result, volume recovery can be time-consuming. Consider the use of the DFSMShsm full-volume dump restore facility to reduce the time that is required for recovery.

You can specify that backup versions be at least as recent as a particular date by specifying the **DATE** parameter of the **RECOVER** command. You must use the **TOVOLUME** parameter to identify an entire volume that you want to recover.

DFSMShsm uses the *latest* VTOC copy data set to recover the volume unless an error occurs in allocating or opening the latest VTOC copy data set. If the error occurs, DFSMShsm attempts to recover the volume by using the *next latest* copy of the VTOC copy data set. If a similar error occurs again, the volume recovery fails.

10.4 Aggregate backup and recovery support

Aggregate backup and recovery support (ABARS) is a component of DFSMShsm that performs data backup processes on a predefined set of data that is known as an *aggregate* and recovers at another computer site or at the same site. ABARS can be used to back up and recover both SMS and non-SMS-managed data.

In general, to use aggregate backup, you first specify the data sets to be backed up, and then use the **ABACKUP** command or ISMF panels to back up the data sets to tape files. These tape files can then be physically transported or transmitted by using a transmission program, such as the IBM Tivoli NetView® for z/OS File Transfer Program (FTP), to a recovery site where the data sets can be recovered.

Only a host that is designated with the HOSTMODE=MAIN attribute can execute aggregate backup or aggregate recovery.

The following functions are shown:

- Define an aggregate group
- Define a selection data set
- Define an instruction data set
- Set your ABARS-related **SETSYS** commands
- Perform an aggregate backup (ABACKUP)
- Perform an aggregate recover (ARECOVER)

10.4.1 Define an aggregate group

Before you can run aggregate backup, you need to create one or more selection data sets and define an aggregate group and related management class to specify exactly which data sets to back up. You can also create an instruction data set to contain any information you want conveyed to the recovery site. One instruction data set name and up to five selection data set names can be defined within one aggregate group.

An aggregate group can be created through ISMF with your selections for data set name, instruction data set name, and the management control information that is required to perform ABACKUP.

Figure 10-9 on page 281 shows the ISMF Aggregate Group Application Selection panel that is displayed after you enter option `9` on the ISMF main entry panel.

```
                      AGGREGATE GROUP APPLICATION SELECTION
Command ===>

To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                                 (1 to 44 Character Data Set Name or 'Active')
  Aggregate Group Name . . MHLRESA    (for Aggregate Group List, fully or
                                     Partially Specified or * for All)
Select one of the following Options:
  3  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria

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

Figure 10-9 ISMF Aggregate Group Application Selection panel

We chose to call our aggregate group, *MHLRESA*. After you type a name for your aggregate group and press Enter, you are presented with the panels that are shown in Figure 10-10 and Figure 10-11 on page 282.

```
                           AGGREGATE GROUP DEFINE              Page 1 of 2
 Command ===>

 SCDS Name  . . . . . : SYS1.SMS.MHLRES3.SCDS
 Aggregate Group Name : MHLRESA

 To DEFINE Aggregate Group, Specify:
  Description ==>  ABARS example
              ==>

  Backup Attributes
    Number of Copies  . . . . . . . 1                (1 to 15)
    Management Class Name . . . . . MNDB22         (1 to 8 characters, to be
                                                     defined in current SCDS)

  Output Data Set Name Prefix . . . MHLRESA
                                                     (1 to 33 Characters)
  Account . . . . . . . . . . . . .
                                                     (1 to 32 Characters)

 Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
 Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-10 Aggregate Group Define (Page 1 of 2)

The management class specifies the parameters, such as the maximum number of aggregate versions to keep, the aggregate group retention parameters, the serialization option, and whether Concurrent Copy is requested.

The prefix identifies the output data sets that are created by aggregate backup. These output data sets are created with the following naming conventions:

- `prefix.D.`*Cnn*`V`*nnnn* and `outputdatasetprefix.O.`*Cnn*`V`*nnnn* for the data file
- `prefix.C.`*Cnn*`V`*nnnn* for the control file
- `prefix.I.`*Cnn*`V`*nnnn* for the instruction/activity log file

Because they all share a common output data set prefix and version number, it is easier to identify all of the output data sets from one aggregate backup.

In our example, the prefix is MHLRESA. The data, control, and instruction/activity log files are named `MHLRESA.D.C01V0001` (for the data file), `MHLRESA.O.C01V0001` (for the control file), and `MHLRESA.C.C01V0001` and `MHLRESA.I.C01V0001` (for the instruction/activity files). *Cnn* is the copy number and *Vnnnn* is the version number that are generated during the ABACKUP operation.

Figure 10-11 shows the second panel of the Aggregate Group Define process.

```
                           AGGREGATE GROUP DEFINE               Page 2 of 2
Command ===>

SCDS Name  . . . . . : SYS1.SMS.MHLRES3.SCDS
Aggregate Group Name : MHLRESA
To Edit a Data Set, Specify Number  . . 1   (1, 2, 3, 4, 5, or 6)
  Selection Data Sets:           (1 to 44 characters)
   1  . . 'MHLRESA.ABARS.SELECT'
     Member Name  . . GROUP1      (1 to 8 characters)
   2  . .
     Member Name  . .             (1 to 8 characters)
   3  . .
     Member Name  . .             (1 to 8 characters)
   4  . .
     Member Name  . .             (1 to 8 characters)
   5  . .
     Member Name  . .             (1 to 8 characters)

  Instruction Data Set:          (1 to 44 characters)
   6  . . 'MHLRESA.ABARS.INSTRUCT'
Use ENTER to Perform Verification; Use UP Command to View previous Panel;
```

Figure 10-11 Aggregate Group Define (Page 2 of 2)

In Figure 10-11, we chose to describe our aggregate group, MHLRESA, make one copy of the output files, take our management class definitions from management class MNDB22, and use MHLRESA as the data set prefix for the output data sets that are created by ABACKUP.

10.4.2 Define selection and instruction data set

A *selection data set* lists the names of the data sets to be processed during ABACKUP. You can create a selection data set by using the Aggregate Group Alter panel, or you can predefine one yourself. Your selection data set must have at least one INCLUDE, ALLOCATE, or ACCOMPANY data set. We show how to set up a basic aggregate with an INCLUDE statement.

In Figure 10-11 on page 282, we chose to call our selection data set, *MHLRESA.ABARS.SELECT* and our instruction data set is called *MHLRESA.ABARS.INSTRUCT*.

To create the data set MHLRESA.ABARS.SELECT and edit member GROUP1, enter option 1 in the "To Edit a Data Set, Specify Number" field, as shown in Figure 10-11 on page 282. ISMF creates the data set and displays member GROUP1 for you to edit, as shown in Figure 10-12.

```
EDIT       MHLRESA.ABARS.SELECT(GROUP1) - 01.00            Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** Bottom of Data ****************************
```

Figure 10-12 Editing panel in ISMF

We selected the data set pattern of MHLRESA.REPORT.** as the candidates for the ABARS process. Therefore, we create an INCLUDE statement in that member, as shown in Figure 10-13.

```
EDIT       MHLRESA.ABARS.SELECT(GROUP1) - 01.00            Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000100 INCLUDE(MHLRES%.REPORT.**)
****** **************************** Bottom of Data ****************************
```

Figure 10-13 Editing the member in ISMF

ABACKUP uses this statement to select cataloged data sets with a high-level qualifier (HLQ) of MHLRESA and a second-level qualifier of REPORT. The instruction data set is an optional data set that is free-form text that is meant to include the necessary information to assist with the recovery process at the recovery site.

It might be useful to include the following types of information:

- SMS attributes
- Description of the application
- RACF environment
- Software requirements
- Hardware requirements
- Unique application dependencies

You can create and edit the MHLRESA.ABARS.INSTRUCT data set in the same way as the preceding selection data set.

You can specify other parameters in your selection data set. For more information, see *DFSMShsm Storage Administration Guide,* SC26-0421, for comprehensive details about additional optional parameters, such as the following parameters:

- EXCLUDE
- ACCOMPANY
- ACCOMPANYEXCLUDE
- ALLOCATE
- ALLOCATEEXCLUDE

10.4.3 Defining SETSYS parameters for aggregate backup

We do not go into detail about each option. Example 10-12 shows the considerations if you decide that you want to implement ABARS.

Example 10-12 Example of SETSYS parameters of ABARS

```
SETSYS                       /* START ONLY ONE SECONDARY ADDRESS  */ -
  MAXABARSADDRESSSPACE(1)    /* SPACE FOR BACKING UP AND          */
                             /* RECOVERING AGGREGATED DATA SETS   */

SETSYS                       /* START THE SECONDARY ADDRESS       */ -
  ABARSPROCNAME(DFHSMABR)    /* SPACE WITH THE STARTUP PROCEDURE  */
                             /* NAMED DFHSMABR.                   */

SETSYS                       /* WRITE THE ABARS ACTIVITY LOG TO   */ -
  ABARSACTLOGTYPE(DASD)      /* DASD                              */

SETSYS                       /* LOG ALL ABARS MESSAGES            */ -
  ABARSACTLOGMSGLVL(FULL)

SETSYS                       /* RECOVER ML2 DATA SETS TO TAPE.    */ -
  ARECOVERML2UNIT(3590-1)

SETSYS                       /* USE 90% OF THE AVAILABLE TAPE FOR */ -
  ARECOVERPERCENTUTILIZED(090) /* ARECOVERY TAPES.                */

SETSYS                       /* BACKUP AGGREGATES TO 3590-1       */ -
  ABARSUNITNAME(3590-1)      /* DEVICES.
```

```
SETSYS                         /* BACKUP ABARS DATA SETS WITH TWO   */ -
  ABARSBUFFERS(2)              /* DATA MOVEMENT BUFFERS.            */

SETSYS                         /* SPECIFY ABARS TO STACK THE        */ -
  ABARSTAPES(STACK)            /* ABACKUP OUTPUT ONTO A MINIMUM     */
/* NUMBER OF TAPE VOLUMES              */

SETSYS                         /* ABARS ACTIVITY LOG WILL NOT BE    */ -
  ABARSDELETEACTIVITY(N)       /* AUTOMATICALLY DELETED DURING      */
                               /* ABARS PROCESSING                  */

SETSYS                         /* SET PERFORMANCE OF BACKING UP     */ -
  ABARSOPTIMIZE(3)             /* LEVEL 0 DASD DATASETS             */

SETSYS                         /* TARGET DATASET IS TO BE ASSIGNED  */ -
  ARECOVERTGTGDS(SOURCE)       /* SOURCE STATUS                     */

SETSYS                         /* ALLOWS RECOVERY OF A LEVEL 0      */ -
  ABARSVOLCOUNT(ANY)           /* DASD DATA SET UP TO 59 VOLUMES    */
 /*                                                                 */

SETSYS                         /* SKIP LO DATA SETS PROTECTED       */ -
  ABARSKIP(PPRC XRC)           /* BY PPRC OR XRC                    */
```

For more information about the **SETSYS** parameters for ABARS, see *DFSMShsm Implementation and Customization Guide,* SC35-0418, and *DFSMShsm Storage Administration Guide,* SC26-0421.

You must also define a procedure for DFSMShsm to use. You must add this procedure to SYS1.PROCLIB. It is not necessary to issue an **MVS START** command for the procedure (DFHSMABR in our case). DFSMShsm schedules the start internally when an ABACKUP or ARECOVER task starts. Example 10-13 shows an ABARS procedure JCL.

Example 10-13 ABARS procedure example

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

After you define the data sets that you want to back up in your selection data set, you can use the DFSMShsm **ABACKUP** command or ISMF panels to back up the data sets to tape.

To activate the new definitions, you must activate the newly configured SCDS by copying the SCDS into the ACDS. Enter option 5 from the CDS Application Selection panel, as shown in Figure 10-14 on page 286.

```
                           CDS APPLICATION SELECTION
Command ===>

To Perform Control Data Set Operations, Specify:
  CDS Name . . 'SYS1.SMS.MHLRES3.SCDS'
                            (1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:
  5  1. Display       - Display the Base Configuration
     2. Define        - Define the Base Configuration
     3. Alter         - Alter the Base Configuration
     4. Validate      - Validate the SCDS
     5. Activate      - Activate the CDS
     6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
     7. Cache Update  - Define/Alter/Delete CF Cache Sets
     8. Lock Display  - Display CF Lock Structure Names for all CF Lock Sets
     9. Lock Update   - Define/Alter/Delete CF Lock Sets
If CACHE Display is chosen, Enter CF Cache Set Name . . *
If LOCK Display is chosen, Enter CF Lock Set Name . . . *
                             (1 to 8 character CF cache set name or * for all)
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-14 CDS Application Selection panel

10.4.4 Perform an aggregate backup (ABACKUP)

After you successfully activate the new definition, you can use ABARS functions for the aggregate group MHLRESA.

To perform an aggregate backup, enter option 5, as shown in Figure 10-15 on page 287.

```
                    AGGREGATE GROUP APPLICATION SELECTION
Command ===>

To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . 'ACTIVE'
                                (1 to 44 Character Data Set Name or 'Active')
  Aggregate Group Name . . MHLRESA   (for Aggregate Group List, fully or
                                    Partially Specified or * for All)
Select one of the following Options:
  5  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria

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

Figure 10-15 Perform ABACKUP by using the ISMF panel

The Aggregate Group Backup panel displays, as shown in Figure 10-16.

```
                             AGGREGATE GROUP BACKUP                Page 1 of 2
Command ===>

CDS Name : ACTIVE

  Aggregate Group Name  . . MHLRESA

  Unit Name . . . . . . . . 3590-1
  Processing Option . . . . 2                   (1=Verify, 2=Execute)
  Wait for Completion . . . N                   (Y or N)
  Stack / Nostack . . . . . N                   (S=STACK, N=NOSTACK or blank)
  Optimize  . . . . . . . .                     (1 to 4 or blank)
  Delete Data Sets After Abackup  . . N         (Y or N)

  Filter Output Data Set Name (1 to 44 Characters)
    ===> ABARS

  Enter "/" to select option
  Process only                L0         ML1        ML2        USERTAPE

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

Figure 10-16 Aggregate Group Backup (Page 1 of 2)

The second Aggregate Group Backup panel displays, as shown in Figure 10-17 on page 288.

```
                          AGGREGATE  GROUP  BACKUP           Page  2 of  2
Command ===>


DFSMShsm Command and Processing Option:
NOWAIT ABACKUP MHLRESA UNIT(3590-1) EXECUTE NOSTACK
FILTEROUTPUTDATASET('MHLRESA.ABARS')









Enter 1 to Submit DFSMShsm ABACKUP  COMMAND
Enter 2 to Save Generated ABACKUP  PARAMETERS

  Select Option  . . 1                         (1=SUBMIT, 2=SAVE)

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

Figure 10-17 Aggregate Group Backup (Page 2 of 2)

In Figure 10-17, we chose to run ABACKUP directly by entering option 1 in the Select Option field to submit the DFSMShsm **ABACKUP** command.

After the ABACKUP process completes, you can view the ABACKUP log data set, as shown in Example 10-14.

Example 10-14 ABACKUP log

```
PAGE 0001  Z/OS DFSMSHSM 1.13.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER 12.2
ARC6000I ABACKUP MHLRESA EXECUTE UNIT(3590-1) FILTEROUTPUTDATASET('MHLRESA.ABARS
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP MHLRESA, AT 12:16:39, STA
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP MHLRESA WILL BE ROUTED TO HSMACT.H2.AB
ARC6379I THE MANAGEMENT CLASS CONSTRUCTS USED IN THE AGGREGATE GROUP, MHLRESA, A
CLASS NAME : MCDB22
DESCRIPTION:
MGMT CLASS FOR DB2 TS
EXPIRATION ATTRIBUTES
  EXPIRE AFTER DAYS NON-USAGE: NOLIMIT
  EXPIRE AFTER DATE/DAYS     : NOLIMIT
RETENTION LIMIT: NOLIMIT
PARTIAL RELEASE: N
MIGRATION ATTRIBUTES
  PRIMARY DAYS NON-USAGE: 7
  LEVEL 1 DAYS NON-USAGE: 20
  COMMAND/AUTO MIGRATE  : BOTH
BACKUP ATTRIBUTES
  BACKUP FREQUENCY                 : 1
  # OF BACKUP VERSIONS (DS EXISTS) : 1
  # OF BACKUP VERSIONS (DS DELETED): 0
  RETAIN DAYS ONLY BACKUP VERSION  : 1
```

```
       (DS DELETED)
RETAIN DAYS EXTRA BACKUP VERSIONS: 1
  ADMIN OR USER COMMAND BACKUP     : BOTH
  AUTO BACKUP                      : Y
  AGGREGATE BACKUP COPY TECHNIQUE  : S

CLASS NAME : MC54NMIG
DESCRIPTION:
MANAGEMENT CLASS FOR NO MIGRATION
EXPIRATION ATTRIBUTES
  EXPIRE AFTER DAYS NON-USAGE: NOLIMIT
  EXPIRE AFTER DATE/DAYS     : NOLIMIT
RETENTION LIMIT: NOLIMIT
PARTIAL RELEASE: C
MIGRATION ATTRIBUTES
  PRIMARY DAYS NON-USAGE: 0
  LEVEL 1 DAYS NON-USAGE: 9999
  COMMAND/AUTO MIGRATE  : BOTH
BACKUP ATTRIBUTES
  BACKUP FREQUENCY                 : 1
  # OF BACKUP VERSIONS (DS EXISTS) : 2
  # OF BACKUP VERSIONS (DS DELETED): 1
  RETAIN DAYS ONLY BACKUP VERSION  : 60
       (DS DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS: 30
ADMIN OR USER COMMAND BACKUP     : BOTH
  AUTO BACKUP                      : Y
  AGGREGATE BACKUP COPY TECHNIQUE  : S

ARC6379I THE STORAGE CLASS CONSTRUCTS USED IN THE AGGREGATE GROUP, MHLRESA, ARE:
CLASS NAME : STANDARD
DESCRIPTION:
PAGE 0002  Z/OS DFSMSHSM 1.13.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER 12.2
STORAGE CLASS FOR AVERAGE RESPONSE TIME
AVAILABILITY: STANDARD
ACCESSIBILITY: S
GUARANTEED SPACE: N
GUARANTEED SYNCHRONOUS WRITE: N

ARC6004I 00D0 ABACKUP PAGE 0001     5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVIC
ARC6004I 00D0 ABACKUP ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS
ARC6004I 00D0 ABACKUP  DUMP DATASET(FILTERDD(SYS00005)) -
ARC6004I 00D0 ABACKUP  OUTDDNAME(SYS00004  -
ARC6004I 00D0 ABACKUP                  ) OPTIMIZE(3) SPHERE
ARC6004I 00D0 ABACKUP  ALLDATA(*) FORCECP(0) -
ARC6004I 00D0 ABACKUP  SHARE TOLERATE(ENQFAILURE)
ARC6004I 00D0 ABACKUP ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO C
ARC6004I 00D0 ABACKUP ADR109I (R/I)-RI01 (01), 2012.240 12:16:59 INITIAL SCAN OF
ARC6004I 00D0 ABACKUP ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION
ARC6004I 00D0 ABACKUP ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
ARC6004I 00D0 ABACKUP ADR006I (001)-STEND(01), 2012.240 12:16:59 EXECUTION BEGIN
ARC6004I 00D0 ABACKUP ADR411W (001)-DTDSC(01),
ARC6004I 00D0 ABACKUP DATA SET MHLRESA.SC64.SPFLOG1.LIST IN CATALOG UCAT.VSBOX01
ARC6004I 00D0 ABACKUP REQUEST
ARC6004I 00D0 ABACKUP ADR804W (001)-DTDSC(01),
```

```
ARC6004I 00D0 ABACKUP EOF FOR DATA SET MHLRESA.SC64.SPFLOG1.LIST IN CATALOG UCAT
ARC6004I 00D0 ABACKUP ALL
ARC6004I 00D0 ABACKUP                          ALLOCATED SPACE WILL BE PROCESSED
ARC6004I 00D0 ABACKUP ADR801I (001)-DTDSC(01),
ARC6004I 00D0 ABACKUP 2012.240 12:17:10 DATA SET FILTERING IS COMPLETE. 14 OF 14
ARC6004I 00D0 ABACKUP SERIALIZATION
ARC6004I 00D0 ABACKUP                          AND 0 FAILED FOR OTHER REASONS
ARC6004I 00D0 ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCC
ARC6004I 00D0 ABACKUP                           MHLRESA.AUDIT.MCDS
ARC6004I 00D0 ABACKUP                           MHLRESA.SC70.ISPF42.ISPPROF
ARC6004I 00D0 ABACKUP                           MHLRESA.ABARS.INSTRUCT
ARC6004I 00D0 ABACKUP                           MHLRESA.LOG.MISC
ARC6004I 00D0 ABACKUP                           MHLRESA.BRODCAST
ARC6004I 00D0 ABACKUP                           MHLRESA.REPORT.JCL
ARC6004I 00D0 ABACKUP                           MHLRESA.ABARS.SELECT
ARC6004I 00D0 ABACKUP                           MHLRESA.SC64.ISPF42.ISPPROF
ARC6004I 00D0 ABACKUP                           MHLRESA.REPORT.LIB2
ARC6004I 00D0 ABACKUP                           MHLRESA.SC64.SPFLOG1.LIST
ARC6004I 00D0 ABACKUP                           MHLRESA.ABARS.OUTPUT
ARC6004I 00D0 ABACKUP                           MHLRESA.ABARS
ARC6004I 00D0 ABACKUP                           MHLRESA.RMMREP.XMIT
ARC6004I 00D0 ABACKUP                           MHLRESA.VALIDATE
ARC6004I 00D0 ABACKUP ADR006I (001)-STEND(02), 2012.240 12:17:16 EXECUTION ENDS
ARC6004I 00D0 ABACKUP ADR013I (001)-CLTSK(01), 2012.240 12:17:21 TASK COMPLETED
ARC6004I 00D0 ABACKUP ADR012I (SCH)-DSSU (01),
ARC6004I 00D0 ABACKUP 2012.240 12:17:21 DFSMSDSS PROCESSING COMPLETE. HIGHEST RE
ARC6004I 00D0 ABACKUP                          TASK    001
ARC6004I 00D0 ABACKUP PAGE 0001     5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVIC
ARC6004I 00D0 ABACKUP ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS
ARC6004I 00D0 ABACKUP  DUMP DATASET(INCLUDE( -
PAGE 0003  Z/OS DFSMSHSM 1.13.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER 12.2
ARC6004I 00D0 ABACKUP   MHLRESA.ABARS.INSTRUCT                      ,    -
ARC6004I 00D0 ABACKUP   HSMACT.H2.ABACKUP.MHLRESA.D12240.T121639      )) -
ARC6004I 00D0 ABACKUP  OUTDDNAME(SYS00017  -
ARC6004I 00D0 ABACKUP                ) OPTIMIZE(3) SPHERE
ARC6004I 00D0 ABACKUP  ALLDATA(*) FORCECP(0) -
ARC6004I 00D0 ABACKUP  SHARE TOLERATE(ENQFAILURE)
ARC6004I 00D0 ABACKUP ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO C
ARC6004I 00D0 ABACKUP ADR109I (R/I)-RI01 (01), 2012.240 12:17:41 INITIAL SCAN OF
ARC6004I 00D0 ABACKUP ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION
ARC6004I 00D0 ABACKUP ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
ARC6004I 00D0 ABACKUP ADR006I (001)-STEND(01), 2012.240 12:17:41 EXECUTION BEGIN
ARC6004I 00D0 ABACKUP ADR801I (001)-DTDSC(01),
 2012.240 12:17:48 DATA SET FILTERING IS COMPLETE. 2 OF 2 DATA SETS WERE SELECTE
ARC6004I 00D0 ABACKUP                          AND 0 FAILED FOR OTHER REASONS
ARC6004I 00D0 ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCC
ARC6004I 00D0 ABACKUP                           MHLRESA.ABARS.INSTRUCT
ARC6004I 00D0 ABACKUP                           HSMACT.H2.ABACKUP.MHLRESA.D12240
ARC6004I 00D0 ABACKUP ADR006I (001)-STEND(02), 2012.240 12:17:53 EXECUTION ENDS
ARC6004I 00D0 ABACKUP ADR013I (001)-CLTSK(01), 2012.240 12:17:58 TASK COMPLETED
ARC6004I 00D0 ABACKUP ADR012I (SCH)-DSSU (01),
 2012.240 12:17:58 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC6382I ACTIVITY LOG HSMACT.H2.ABACKUP.MHLRESA.D12240.T121639 HAS BEEN SUCCESSF
ARC6382I INSTRUCTION DATA SET MHLRESA.ABARS.INSTRUCT HAS BEEN SUCCESSFULLY BACKE
ARC6369I STORAGE REQUIREMENTS FOR AGGREGATE GROUP MHLRESA ARE: L0=2109K, ML1=0,
```

```
ARC6061I VOLUMES USED FOR CONTROL FILE MHLRESA.C.C01V0001 DURING AGGREGATE BACKU
  THS014
ARC6060I VOLUMES USED FOR DATA FILE MHLRESA.D.C01V0001 DURING AGGREGATE BACKUP F
  THS006
ARC6071I VOLUMES USED FOR INSTRUCTION/ACTIVITY LOG FILE MHLRESA.I.C01V0001 DURIN
MHLRESA ARE:
  THS007
ARC6055I AGGREGATE BACKUP HAS COMPLETED FOR AGGREGATE GROUP MHLRESA, AT 12:18:33
```

10.4.5 Perform an aggregate recover (ARECOVER)

We describe how to recover a single data set that was backed up previously. On the Aggregate Group Application Selection panel, enter option 6 to recover an aggregate group, as shown in Figure 10-18.

```
                    AGGREGATE GROUP APPLICATION SELECTION
Command ===>

To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . 'ACTIVE'
                                 (1 to 44 Character Data Set Name or 'Active')
  Aggregate Group Name . . MHLRESA   (for Aggregate Group List, fully or
                                     Partially Specified or * for All)
Select one of the following Options:
  6  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria

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

Figure 10-18 Perform ARECOVER by using ISMF

Pressing Enter will open the next panel, which is shown in Figure 10-19 on page 292.

```
                           AGGREGATE GROUP RECOVER               Page 1 of 8
Command ===>

  Abackup Control Dataset  . .
                                             (1 to 44 Characters)
    Xmit . . . . . . . . . . . . N           (Y or N)
    Stack / Nostack  . . . . .               (S=STACK, N=NOSTACK or blank)
  Aggregate Group Name . . . . MHLRESA
    Date . . . . . . . . . . . . 2012/08/27  (yyyy/mm/dd)
    Version  . . . . . . . . .               (1 to 9999)

  Processing Option  . . . . . 3             (1=Prepare, 2=Verify, 3=Execute)
  Wait for Completion  . . . . N             (Y or N)
  Target GDG Data Set Status                 (A=ACTIVE, D=DEFERRED, R=ROLLEDOFF,
                                             S=SOURCE or blank)
  Volume Count . . . . . . . .               (A=ANY or blank)
  Recover Instruction Data Set  . . N        (Y or N)
  Recover Activity Log  . . . . . . N        (Y or N)
  Recover Individual Data Sets  . . Y        (Y or N)

Use ENTER to Continue; Use DOWN to View Additional Options;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-19 ABARS Group Recover panel

We chose to recover a single data set, 'MHLRESA.REPORT.JCL', on the same system (assuming it was lost and no longer cataloged) by entering option Y in the Recover Individual Data Sets field, as shown in Figure 10-19.

The next panel requires you to enter the data set name that you want to recover, which, in our case, is 'MHLRESA.REPORT.JCL', as shown in Figure 10-20 on page 293.

```
                         AGGREGATE GROUP RECOVER               Page 7 of 8
Command ===>

AGGREGATE GROUP NAME:
MHLRESA

Specify Data Set Name for Individual Recovery:

  Single Data Set Name . . . . 'MHLRESA.REPORT.JCL'
                                                   (1 to 44 Characters)
  List Of Names Data Set . . .
                                                   (1 to 44 Characters)









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

Figure 10-20 ABARS recover single data set by using ISMF

Pressing Enter opens the Aggregate Group Recover panel that is shown in Figure 10-21.

```
                          AGGREGATE  GROUP  RECOVER           Page  8 of  8
Command ===>


DFSMShsm Command and Processing Option:
NOWAIT ARECOVER AGGREGATE(MHLRESA) DATE(2012/08/27)
EXECUTE
MIGRATEDDATA(ML1)
ONLYDATASET(NAME('MHLRESA.REPORT.JCL'))







Enter 1 to Submit DFSMShsm ARECOVER COMMAND
Enter 2 to Save Generated ARECOVER PARAMETERS

  Select Option  . . 1                         (1=SUBMIT, 2=SAVE)

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

Figure 10-21 ABARS recover single data set second panel

We chose to run the job, so we entered option 1. DFSMShsm starts ABARS to recover the data set.

After the ARECOVER process completes, you can view the ARECOVER log data set as shown in Example 10-15.

Example 10-15 ARECOVER log data set

```
PAGE 0001  Z/OS DFSMSHSM 1.13.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER 12.2
ARC6000I ARECOVER AGGREGATE(MHLRESA) DATE(2012/08/27) EXECUTE MIGRATEDDATA(ML1)
ARC6102I AGGREGATE RECOVERY STARTING USING CONTROL FILE DATA SET MHLRESA.C.C01V0
 STARTED TASK = DFHSMABR.ABAR0154
ARC6030I ACTIVITY LOG FOR CONTROL FILE DATA SET MHLRESA.C.C01V0001 WILL BE ROUTE
HSMACT.H2.ARECOVER.MHLRESA.D12240.T123654
ARC6308I CONFLICT RESOLUTION DATA SET HSM.MHLRESA.CONFLICT.D12240.T121639 WILL B
ARC6115I AGGREGATE RECOVERY USING CONTROL FILE DATA SET MHLRESA.C.C01V0001 WILL
  THS006
  THS007
ARC6119I STORAGE CLASS NAMES EXISTED DURING AGGREGATE BACKUP OF AGGREGATE GROUP
MHLRESA.C.C01V0001
  STANDARD
ARC6119I MANAGEMENT CLASS NAMES EXISTED DURING AGGREGATE BACKUP OF AGGREGATE GRO
MHLRESA.C.C01V0001
  MC54NMIG
  MCDB22
ARC6004I 00D0 ARECOVER PAGE 0001     5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVI
ARC6004I 00D0 ARECOVER ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS
ARC6004I 00D0 ARECOVER  RESTORE DATASET(FILTERDD(SYS00007)) -
ARC6004I 00D0 ARECOVER  INDDNAME(SYS00006) -
ARC6004I 00D0 ARECOVER  OUTDYNAM( -
ARC6004I 00D0 ARECOVER   (ML940D) -
ARC6004I 00D0 ARECOVER   ) -
ARC6004I 00D0 ARECOVER  PERCENTUTILIZED( -
ARC6004I 00D0 ARECOVER  090 -
ARC6004I 00D0 ARECOVER   ) -
ARC6004I 00D0 ARECOVER  VOLCOUNT(ANY) -
ARC6004I 00D0 ARECOVER  SPHERE  -
ARC6004I 00D0 ARECOVER  TGTGDS(SOURCE    ) -
ARC6004I 00D0 ARECOVER  CATALOG FORCE FORCECP(0)
ARC6004I 00D0 ARECOVER ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO
ARC6004I 00D0 ARECOVER ADR109I (R/I)-RI01 (01), 2012.240 12:38:05 INITIAL SCAN O
ARC6004I 00D0 ARECOVER ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION
ARC6004I 00D0 ARECOVER ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FO
ARC6004I 00D0 ARECOVER ADR006I (001)-STEND(01), 2012.240 12:38:05 EXECUTION BEGI
ARC6004I 00D0 ARECOVER ADR780I (001)-TDDS (01),
ARC6004I 00D0 ARECOVER THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL DAT
ARC6004I 00D0 ARECOVER DFSMSDSS VERSION
ARC6004I 00D0 ARECOVER                          1 RELEASE 13 MODIFICATION LEVEL
ARC6004I 00D0 ARECOVER ADR711I (001)-NEWDS(01), DATA SET MHLRESA.REPORT.JCL HAS
ARC6004I 00D0 ARECOVER NO DATACLAS, AND MGMTCLAS MC54NMIG
ARC6004I 00D0 ARECOVER ADR474I (001)-TDNVS(01),
ARC6004I 00D0 ARECOVER DATA SET MHLRESA.REPORT.JCL CONSISTS OF 00000010 TARGET T
ARC6004I 00D0 ARECOVER ADR489I (001)-TDLOG(01), DATA SET MHLRESA.REPORT.JCL WAS
ARC6004I 00D0 ARECOVER ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUC
ARC6004I 00D0 ARECOVER                           MHLRESA.REPORT.JCL
ARC6004I 00D0 ARECOVER ADR006I (001)-STEND(02), 2012.240 12:38:07 EXECUTION ENDS
ARC6004I 00D0 ARECOVER ADR013I (001)-CLTSK(01), 2012.240 12:38:12 TASK COMPLETED
PAGE 0002  Z/OS DFSMSHSM 1.13.0  DATA FACILITY HIERARCHICAL STORAGE MANAGER 12.2
ARC6004I 00D0 ARECOVER ADR012I (SCH)-DSSU (01),
```

```
ARC6004I 00D0 ARECOVER 2012.240 12:38:12 DFSMSDSS PROCESSING COMPLETE. HIGHEST R
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY RECOVERED USING AGGREGATE GRO
  MHLRESA.REPORT.JCL
ARC6103I AGGREGATE RECOVERY HAS COMPLETED FOR AGGREGATE GROUP MHLRESA, USING CON
 AT 12:38:12, RETCODE = 000
```

You can use the same process to recover the entire data sets that were backed up by ABARS in the same way that you recover a single data set.

10.4.6 Interrupting and restarting ABARS processing

Three methods are available to interrupt automatic dump processing:

- Hold ABARS processing: You can hold ABARS processing by specifying the **HOLD** command for the **ABACKUP**, **ARECOVER**, or **ALL** parameter. If you hold ABARS processing with the **ENDOFDATASET** parameter, processing ends after DFSMShsm finishes processing the current data set.
- Place DFSMShsm in emergency mode: You can place DFSMShsm in emergency mode by specifying the **SETSYS EMERGENCY** command.
- Stop DFSMShsm processing: You can stop DFSMShsm by entering the MVS or DFSMShsm **STOP** command.

You can restart automatic dump processing by entering the corresponding commands:

- `RELEASE ABACKUP`, `ARECOVER`, or `RELEASE ALL`
- `SETSYS NOEMERGENCY`
- `START DFSMSHSM` (MVS command)

10.5 Fast replication backup and recovery

Fast replication is a function that uses volume-level fast replication to create backup versions for sets of storage groups. It uses IBM FlashCopy®, which is a point in time copy of a volume. A point in time copy gives the appearance of an almost instantaneous volume copy.

The process of fast data replication occurs so fast because it builds a map, with pointers, to the source volume tracks or extents. You no longer need to wait for the physical copy to complete before applications can resume the access to the data. Both the source and target data are available for read/write access almost immediately, while the copy process continues in the background. This process guarantees that the contents of the target volume are an exact duplicate of the source volume at that point in time.

You define a set of storage groups with the SMS “copy pool” construct. Fast replication target volumes contain the fast replication backup copies of volumes that are managed by DFSMShsm. Fast replication target volumes are defined with the SMS “copy pool backup” storage group type. The fast replication backup versions on DASD can then be dumped to tape by using either the **FRBACKUP** command or with Automatic Dump.

Recovery from the fast replication backup versions can be performed at the data set, volume, or copy pool level. The entire copy pool, individual volumes, and data sets within a copy pool can be recovered from the fast replication backup versions on DASD or tape. Individual data sets are recovered to the volume or volumes that they existed on at the time of the backup.

The following functions are described:

- Define a copy pool backup storage group
- Assign a storage group to copy pool backup and copy pool construct
- Perform a simple fast replication backup
- Perform a simple fast replication recover

10.5.1 Define copy pool backup storage group

The SMS copy pool backup storage group type contains the candidate target volumes for DFSMShsm fast replication requests. The volumes that are associated with a copy pool backup storage group are for DFSMShsm use. If you need to restrict the use of a volume in a copy pool backup storage group, vary the volume offline.

For each source volume in a storage group to be copied, enough eligible target volumes must exist in the copy pool backup storage group to satisfy the number of volumes that are required by the number of specified backup versions.

An eligible target volume must meet the following requirements:

- Have the same track format as the source volume.
- Be the same size as the source volume.
- For FlashCopy:
  - Not be a primary or secondary volume in a Global Mirror (XRC) volume pair.
  - For FlashCopy version 1:
    - Reside in the same logical subsystem (LSS) as the source volume
    - Not be in a FlashCopy relationship at the time of the backup

In our example, we chose the name of *DB0BCPB* for the copy pool. From the ISMF main menu, enter option `6` (Storage Group Application Selection). We entered the name `DB0BCPB` in the Storage Group Name field, and we entered `COPY POOL BACKUP` in the Storage Group Type field. We entered option `3` to define a new storage group, as shown in Figure 10-22 on page 297.

```
                     STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Storage Group Name    DBOBCPB           (For Storage Group List, fully or
                                          partially specified or * for all)
  Storage Group Type    COPY POOL BACKUP  (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                          OBJECT, OBJECT BACKUP, or TAPE)
Select one of the following options  :
  3  1. List    - Generate a list of Storage Groups
     2. Display - Display a Storage Group (POOL only)
     3. Define  - Define a Storage Group
     4. Alter   - Alter a Storage Group
     5. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
        Space Info in GB           Respecify Sort Criteria
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-22 ISMF Storage Group Application Selection panel

When you press Enter, ISMF opens the next panel. We entered the description for this copy pool, as shown in Figure 10-23.

```
                     COPY POOL BACKUP STORAGE GROUP DEFINE
 Command ===>

 SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
 Storage Group Name  : DBOBCPB

 To DEFINE Storage Group, Specify:

  Description ==> DBOB COPY POOL BACKUP
              ==>











  DEFINE    SMS Storage Group Status . . . Y (Y or N)

 Use ENTER to Perform Verification;
 Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-23 Copy Pool Backup Storage Group Define panel

The next panel allows us to define SMS storage group status for all systems in the SMSplex. In our case, we enabled this storage group to all four systems in the SMSplex by entering `ENABLE`, as shown in Figure 10-24.

```
                      SMS STORAGE GROUP STATUS DEFINE         Page 1 of 2
Command ===>

SCDS Name  . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name : DB0BCPB
Storage Group Type : COPY POOL BACKUP
To DEFINE Storage Group System/                   ( Possible SMS SG Status
Sys Group Status, Specify:                         for each:
                                                   - Pool SG Type
 System/Sys      SMS SG   System/Sys      SMS SG       NOTCON, ENABLE, DISALL
 Group Name      Status   Group Name      Status       DISNEW, QUIALL, QUINEW
 ----------      ------   ----------      ------   - Tape SG Type
 SC63       ===> ENABLE   SC64       ===> ENABLE       NOTCON, ENABLE,
 SC65       ===> ENABLE   SC70       ===> ENABLE       DISALL, DISNEW
            ===>                     ===>          - Copy Pool Backup SG Type
            ===>                     ===>              NOTCON, ENABLE )
            ===>                     ===>          * SYS GROUP = sysplex
            ===>                     ===>            minus Systems in the
            ===>                     ===>            Sysplex explicitly
            ===>                     ===>            defined in the SCDS
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-24 SMS Storage Group Status Define panel

Press PF3 twice to save the new definition. ISMF returns to the first panel.

The next step is to define DASD volumes for this storage group, as shown in Figure 10-25 on page 299.

```
                       STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Storage Group Name   DBOBCPB          (For Storage Group List, fully or
                                        partially specified or * for all)
  Storage Group Type                    (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                        OBJECT, OBJECT BACKUP, or TAPE)
Select one of the following options  :
  5  1. List    - Generate a list of Storage Groups
     2. Display - Display a Storage Group (POOL only)
     3. Define  - Define a Storage Group
     4. Alter   - Alter a Storage Group
     5. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
        Space Info in GB           Respecify Sort Criteria
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-25 Storage Group Application Selection panel

On the next panel, we entered the volsers or volume patterns to define them to be part of this storage group, as shown in Figure 10-26.

```
                        STORAGE GROUP VOLUME SELECTION
Command ===>

CDS Name  . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : DBOBCPB
Storage Group Type  : COPY POOL BACKUP

Select One of the following Options:
  2  1. Display - Display SMS Volume Statuses (Pool & Copy Pool Backup only)
     2. Define  - Add Volumes to Volume Serial Number List
     3. Alter   - Alter Volume Statuses (Pool & Copy Pool Backup only)
     4. Delete  - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
      Prefix   From     To    Suffix  Type
      ______  ______  ______  _____    _
 ===> SBOXK   A       H                A
 ===> SBOXK   0       1                X
 ===>
 ===>
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 10-26 Storage Group Volume Selection panel

In our example, we defined volumes SBOXKA - SBOXKH and SBOXK0 - SBOXK1 to be part of this copy pool storage group.

On the next panel, we defined the volume status on each system of the SMSplex, as shown in Figure 10-27. We chose to enable all volumes to all systems.

```
                         SMS VOLUME STATUS DEFINE             Page 1 of 2
Command ===>

SCDS Name . . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  . : DB0BCPB
Volume Serial Numbers : SBOXKA - SBOXKH       SBOXK0 - SBOXK1

To DEFINE SMS Volume Status, Specify:

 System/Sys      SMS Vol     System/Sys      SMS Vol    ( Possible SMS Vol
 Group Name      Status      Group Name      Status       Status for each:
 ----------      -------     ----------      -------      NOTCON, ENABLE,
 SC63       ===> ENABLE      SC64       ===> ENABLE       DISALL, DISNEW,
 SC65       ===> ENABLE      SC70       ===> ENABLE       QUIALL, QUINEW )
            ===>                        ===>
            ===>                        ===>            * SYS GROUP = sysplex
            ===>                        ===>              minus systems in the
            ===>                        ===>              sysplex explicitly
            ===>                        ===>              defined in the SCDS
            ===>                        ===>
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-27 SMS Volume Status Define panel

We pressed PF3 twice to save the new configuration and to complete the copy pool backup definition process.

The next step is to define this copy pool backup and copy pool construct to any storage group where it is wanted as a copy pool.

10.5.2 Assign storage groups to copy pool backup and copy pool construct

In our example, we chose storage group, *DB0BARCH*.

From the Storage Group List panel, we entered the `ALTER` command against the DB0BARCH storage group, as shown in Figure 10-28 on page 301.

```
                           STORAGE GROUP LIST
Command ===>                                                    Scroll ===> CSR
                                                          Entries 1-8 of 8
                                                          Data Columns 3-6 of 48
CDS Name : MHLRESA.SMS.SCDS

Enter Line Operators below:

     LINE       STORGRP  SG                 VIO      VIO   AUTO
     OPERATOR   NAME     TYPE               MAXSIZE  UNIT  MIGRATE
    ---(1)----  --(2)--- -------(3)------  --(4)--  (5)-  --(6)---
    ALTER       DBOBARCH POOL               -------  ----  NO
                DBOBCPB  COPY POOL BACKUP  -------  ----  --------
                DBOBDATA POOL               -------  ----  NO
                DBOBIMAG POOL               -------  ----  NO
                DBOBLOG1 POOL               -------  ----  NO
                DBOBLOG2 POOL               -------  ----  NO
                DBOBMISC POOL               -------  ----  NO
                DBOBTARG POOL               -------  ----  NO
    ----------  -------- ------  BOTTOM  OF  DATA  ------  --------  ----------
```

Figure 10-28 Storage Group List panel

We then typed the copy pool DBOBCPB in the Copy Pool Backup SG Name field, as shown in Figure 10-29, and saved it.

```
                          POOL STORAGE GROUP ALTER         Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Group Name  : DBOBARCH
To ALTER Storage Group, Specify:
 Description ==> DBOB ARCHIVE LOGS
             ==>
 Auto Migrate . . N  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . N  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . N  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . . . .
                                      Copy Pool Backup SG Name . . . DBOBCPB

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

 ALTER     SMS Storage Group Status . . . N   (Y or N)
 ALTER     SMA Attributes . . . . . . . . N   (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-29 Pool Storage Group Alter panel

You also need to define storage group candidates to the copy pool construct.

From the ISMF primary panel, we entered option `P` Copy Pool (Specify Pool Storage Groups for Copies), as shown in Figure 10-30.

```
                   ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R13
Selection or Command ===>

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
G  Report Generation         - Create Storage Management Reports
L  List                      - Perform Functions Against Saved ISMF Lists
P  Copy Pool                 - Specify Pool Storage Groups for Copies
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                      - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 10-30 ISMF Primary Option Menu panel

The copy pool construct that relates to DB2 must point to the SMS storage groups that will be processed for fast replication operations.

The BACKUP SYSTEM utility requires that at least a database copy pool, which consists of all database objects (DB2 catalog, directory, and all user data), exists to take a data-only backup. If you plan to take full system backups, a log copy pool that consists of the active logs and bootstrap data sets (BSDSs) are required. The DB2 RESTORE SYSTEM utility uses only the database copy pool. A copy pool can contain up to 256 SMS storage groups to be processed for fast replication operations.

The copy pool naming convention must be in the following form:

`DSN$`*`locn-name`*`$`*`cp-type`*

- DSN is the unique DB2 product identifier.
- $ is a required delimiter.
- *locn-name* is the DB2 location name.
- $ is a required delimiter.
- *cp-type* is the copy pool type:
  - `DB` for database
  - `LG` for logs

In our example, we chose the *locn-name*, *DB0B*. Therefore, we defined the copy pool construct as *DSN$DB0B$DB*.

Figure 10-31 on page 303 shows how to define a new copy pool construct.

```
                      COPY POOL APPLICATION SELECTION
Command ===>

To perform Copy Pool Operations, Specify:
  CDS Name . . . . 'SYS1.SMS.MHLRES3.SCDS'
                                (1 to 44 character data set name or 'Active' )
  Copy Pool Name   DSN$DBOB$LG                    (For Copy Pool List, fully
                                         or partially specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Copy Pools
     2. Display - Display a Copy Pool
     3. Define  - Define a Copy Pool
     4. Alter   - Alter a Copy Pool

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 10-31 Copy Pool Application Selection panel

Next, specify the options for dump, as shown in Figure 10-32. We chose to use this copy pool construct with automatic dump. We entered `DB0BTST1` for the Dump Class.

```
                              COPY POOL DEFINE                  Page 1 of 5
Command ===>

SCDS Name  . . : SYS1.SMS.MHLRES3.SCDS
Copy Pool Name : DSN$DBOB$LG

To DEFINE Copy Pool, Specify:
  Description ==> DBOB ARCH AND LOGS
              ==>
  Auto Dump . . . Y   (Y or N)        Dump Sys/Sys Group Name . . .
  Dump Class  . . DBOBTST1            Dump Class  . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .

  Number of DASD Fast Replication Backup
   Versions with Background Copy . . . . . 1       (0 to 85 or blank)
  FRBACKUP to PPRC Primary Volumes allowed . . .     (NO, PN, PP, PR or blank)
  FRRECOV to PPRC Primary Volumes allowed  . . .     (NO, PN, PP, PR or blank)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-32 Copy Pool Define panel

In the next panel, we defined the catalog name that resides in the storage group DB0BDATA, as shown in Figure 10-33 on page 304.

```
                              COPY POOL DEFINE                  Page 2 of 5
Command ===>

SCDS Name  . . : SYS1.SMS.MHLRES3.SCDS
Copy Pool Name : DSN$DB0B$LG

To DEFINE Copy Pool, Specify:
  Catalog Name ==> UCAT.DB0BLOGS
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
  Capture Catalog Information for Data Set Recovery . . . R   (R, P or N)
  Allow Fast Reverse Restore  . . . . . . . . . . . . . . Y   (Y or N)

Use ENTER to Perform Verification; Use UP/DOWN Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-33 Copy Pool Define next panel

In Figure 10-33, we chose to capture the information of catalog UCAT.DB0BLOGS for the FRBACKUP function. If the FRBACKUP process failed to capture this catalog information, it fails the backup version.

We also allowed the Fast Reverse Restore function. No additional keywords need to be specified to recover from a DASD copy pool fast reverse restore-eligible version. If the state of the FlashCopy relationships meets the fast reverse restore requirements, DFSMShsm uses fast reverse restore to recover the copy pool. Otherwise, DFSMShsm uses regular FlashCopy if the background copy completed.

After a successful fast reverse restore, the contents of the DASD backup volume are invalid. DFSMShsm invalidates and initializes the individual DASD backup volume so it is ready to be reused. When the entire copy pool is successfully recovered, DFSMShsm invalidates the DASD copy pool backup version.

If several of the volumes fail to recover, the FASTREPLICATIONSTATE of the DASD version becomes FCFRRINCOMPLETE. You can issue the **FRRECOV COPYPOOL(*cpname*)** command again after you address the reason for the failure. DFSMShsm tries to recover the remaining volumes again by using fast reverse restore.

Fast reverse restore is not supported with multiple FlashCopy targets. For a successful fast reverse restore, verify that no multiple FlashCopy targets exist before you issue the **FRRECOV COPYPOOL(*cpname*)** command. To remove unneeded FlashCopy targets that are managed by DFSMShsm, issue **FRBACKUP COPYPOOL(*cpname*) WITHDRAW** to withdraw the most recent DASD backup copy, or issue **FRDELETE COPYPOOL(*cpname*) VERSION(*n*)** to delete a specific backup version. For FlashCopy targets that are not managed by DFSMShsm, issue **ICKDSF FLASHCPY WITHDRAW** or **SDM FCWITHDR** to withdraw the unneeded FlashCopy relationships.

To determine whether a copy pool is defined to allow fast reverse restore and to select a fast reverse restore eligible backup version, issue **LIST COPYPOOL (*cpname*)**. If the copy pool is defined to allow fast reverse restore at the time of the backup, the output displays the backup version with FCFRR=Y. Otherwise, the output contains FCFRR=N.

Example 10-16 shows the copy pool construct DSN$DB0B$LG, which is defined to allow fast reverse restore.

Example 10-16 DSN$DB0B$LG information from the LIST CP command

```
1-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 12:32:34 ON
12/08/31 FOR SYSTEM=SC64

0COPYPOOL=DSN$DB0B$LG
 ALLOWPPRCP FRB=NO FRR=NO

 VERSION  VTOCENQ     DATE          TIME     FASTREPLICATIONSTATE  DUMPSTATE
   007      Y      2012/08/29      18:03:38   RECOVERABLE           NONE
 TOKEN(C)=C'MHLRESA'
 TOKEN(H)=X'D4C8D3D9C5E2C1'
 TOTAL NUM OF VOLUMES=00004,INCREMENTAL=N,CATINFO=Y,FCFRR=Y,RECOVERYINCOMPLETE=N

 SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
 DB0BARCH SBOXJ0 - SBOXK0  SBOXJ1 - SBOXK1
 DB0BLOG1 SBOXJ8 - SBOXKA
 DB0BLOG2 SBOXJ9 - SBOXKB

0----- END OF -- COPY POOL -- LISTING -----
```

To determine whether fast reverse restore can be used for a copy pool version, issue the **QUERY COPYPOOL** command. When fast reverse restore can be used, the **QUERY COPYPOOL** command output displays "background copy percent-complete" (PCT-COMP) information other than "***". Percent-complete information (a percentage) is available for full-volume FlashCopy pairs with an incomplete background copy only. A full-volume FlashCopy relationship is established when the FlashCopy technique (such as fast reverse restore or incremental) designates it, or when SETSYS FASTREPLICATION(FCRELATION(FULL)) was specified.

Because fast reverse restore invalidates the entire DASD backup version after a successful recovery, you can use the PCT-COMP percentage to determine the progress of the background copy and decide whether to use fast reverse restore for recovery.

Example 10-17 shows the output from the **QUERY COPYPOOL(DSN$DB0B$LG)** command.

Example 10-17 Output from the QUERY COPYPOOL(DSN$DB0B$LG) command

```
ARC1820I THE FOLLOWING VOLUMES IN COPY POOL DSN$DB0B$LG, VERSION 008, HAVE AN
ARC1820I (CONT.) ACTIVE FLASHCOPY BACKGROUND COPY
ARC1820I (CONT.) SGNAME   FR-PRIMARY FR-BACKUP  PCT-COMP
ARC1820I (CONT.) DB0BARCH SBOXJ0     SBOXK0     008
ARC1820I (CONT.) DB0BARCH SBOXJ1     SBOXK1     009
ARC1820I (CONT.) DB0BLOG1 SBOXJ8     SBOXKA     068
ARC1820I (CONT.) DB0BLOG2 SBOXJ9     SBOXKB     001
***
```

Notes:

1. A DASD backup version can be used for fast reverse restore only one time because the process invalidates the data on the backup volume. Keep dump tape copies in addition to DASD backup copies when you plan to use the fast reverse restore function.
2. VERIFY(N) cannot be specified when the copy pool is defined to allow fast reverse restore unless a previous FRRECOV COPYPOOL operation determined that the copy pool backup version is no longer eligible to use fast reverse restore. Regular fast replication recovery can be used instead.
3. When a recovery operation, such as fast reverse restore from a DASD backup, will invalidate an existing DASD copy with an incomplete dump copy, the recovery operation fails. To force the use of a DASD copy with an incomplete dump copy, specify the **FORCE** parameter on the **FRRECOV** command.
4. The preserve mirror operation cannot be used in combination with fast reverse restore. Do not specify PMPREF or PMREQ in the “FRBACKUP to PPRC Primary Volumes allowed” or “FRRECOV to PPRC Primary Volumes allowed” fields for a copy pool that is defined to allow fast reverse restore.

In the next panel, we entered the storage group name DBOBARCH to be copied to this copy pool construct, as shown in Figure 10-34.

```
                              COPY POOL DEFINE        ENTER STORAGE GROUP NAME
Command ===>

SCDS Name  . . : SYS1.SMS.MHLRES3.SCDS
Copy Pool Name : DSN$DBOB$LG

To DEFINE Copy Pool, Specify:
  Storage Group Names:  (specify 1 to 256 names)
   ==> DBOBARCH
   ==>
   ==>
   ==>
   ==>
   ==>


Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 10-34 Copy Pool Define next panel

We pressed PF3 to save the new definition, which completed the define storage group to copy pool backup storage group and copy pool construct.

10.5.3 Perform fast replicate backup (FRBACKUP)

The *fast replication backup* function is not available as part of automatic backup or dump. You can invoke the **FRBACKUP** command from a system console, with a TSO command (**HSENDCMD**), with a batch job, or by using the ARCHSEND macro interface. DFSMShsm invokes the DFSMSdss COPY FULL function to create a fast replication backup for each volume in the specified copy pool.

The simple syntax command is shown:

```
FRBACKUP COPYPOOL(cpname) EXECUTE
```

In our example, we chose to use FRBACKUP with a batch job and to use the DUMP option to also copy those volumes to tapes after the fast replication successfully completed, as shown in Example 10-18.

Example 10-18 Sample JCL for FRBACKUP with DUMP options

```
//MHLRESAY JOB (999,POK),'MHLRESA',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM SYSAFF=*
//HSMFR EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//PRINTER  DD SYSOUT=*
//INDEX    DD SYSOUT=*
//SYSIN    DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
 HSEND FRBACKUP COPYPOOL(DSN$DBOB$LG) TOKEN(MHLRESA) DUMP -
 DUMPCLASS(DBOBTST1) EXECUTE
```

The **FRBACKUP COPYPOOL** command is successful only after DFSMSdss successfully establishes a fast replication relationship for each source volume. If one of the volumes fails, the entire backup version is failed with the following results:

- DFSMShsm continues to process volumes to determine whether any remaining volumes will also encounter an error.
- For each volume in error, the target volume is disassociated from the source volume. If a problem exists with the target volume, the target volume needs to be removed from the copy pool backup storage group. Otherwise, it can be reselected during the next **FRBACKUP** command.
- FlashCopy relationships that are successfully established are withdrawn after all volumes are processed.
- The backup version is marked as invalid after all volumes are processed.

After the batch job is complete, you can check in the DFSMShsm log to verify the FRBACKUP process, as shown in Example 10-19.

Example 10-19 Sample of FRBACKUP activities log

```
ARC1801I FAST REPLICATION BACKUP DUMP IS STARTING FOR COPY POOL DSN$DBOB$LG, AT
13:38:55 ON 2012/08/31, TOKEN='MHLRESA'
 ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 13:38
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC
CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  PARALLEL
```

```
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO
COMMAND 'PARALLEL '
 ARC0640I ARCFRTM -  COPY IDY(SBOXJ0) ODY(SBOXK0) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  FCFVR -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(SBOXJ1) ODY(SBOXK1) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  FCFVR -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(SBOXJ8) ODY(SBOXKA) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  FCFVR -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 004 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(SBOXJ9) ODY(SBOXKB) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  FCFVR -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 005 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2012.244 13:38:55 INITIAL SCAN OF
USER CONTROL STATEMENTS COMPLETED
 ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
  2012.244 13:38:55 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN
EFFECT
 ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR050I (004)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (004)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR050I (005)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (005)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR006I (005)-STEND(01), 2012.244 13:38:55 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (005)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (005)-TOMI (02), VOLUME SBOXJ9 WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (005)-STEND(02), 2012.244 13:38:55 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (005)-CLTSK(01), 2012.244 13:38:55 TASK COMPLETED WITH
RETURN CODE 0000
```

```
 ARC0640I ARCFRTM - ADR006I (004)-STEND(01), 2012.244 13:38:55 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (004)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (004)-T0MI (02), VOLUME SBOXJ8 WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (004)-STEND(02), 2012.244 13:38:55 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (004)-CLTSK(01), 2012.244 13:38:56 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2012.244 13:38:55 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (003)-T0MI (02), VOLUME SBOXJ1 WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2012.244 13:38:55 EXECUTION ENDS
 ARC0640I ARCFRTM - PAGE 0003    5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 13:38
 ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2012.244 13:38:56 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - PAGE 0002    5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 13:38
 ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2012.244 13:38:55 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (002)-T0MI (02), VOLUME SBOXJ0 WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2012.244 13:38:55 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2012.244 13:38:57 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2012.244 13:38:57 DFSMSDSS PROCESSING
COMPLETE. HIGHEST RETURN CODE IS 0000
 ARC1805I THE FOLLOWING 00004 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST
REPLICATION BACKUP OF COPY POOL DSN$DB0B$LG
 ARC1805I (CONT.) SBOXJ0
 ARC1805I (CONT.) SBOXJ1
 ARC1805I (CONT.) SBOXJ8
 ARC1805I (CONT.) SBOXJ9
 ARC1802I FAST REPLICATION BACKUP DUMP HAS COMPLETED FOR COPY POOL DSN$DB0B$LG, AT
13:42:25 ON 2012/08/31,
  FUNCTION RC=0000, MAXIMUM VOLUME RC=0000, CAPTURE CATALOG RC=0000
```

The log also shows the DUMP activities, as shown in Example 10-20.

Example 10-20 Sample of full volume dump activities for FRBACKUP with the DUMP option

```
ARC0622I FULL VOLUME DUMP STARTING ON VOLUME SBOXJ8(SMS) AT 13:42:07 ON
2012/08/31, SYSTEM SC64, TASK ID=ARCDVOL1 ,
  TO DUMP CLASS(ES)=   DB0BTST1
 ARC0728I VTOC FOR VOLUME SBOXJ8 COPIED TO DATA SET
HSM.DUMPVTOC.T553813.VSBOXJ8.D12244 ON VOLUME SBXHS6
 ARC0640I ARCDVOL1 - PAGE 0001    5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 13:42
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC
CLASS CHK DEFAULT TO YES
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT
DEFAULT TO
 ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS01492)         -
```

```
 ARC0640I ARCDVOL1 -  OUTDDNAME(SYS01488) -
 ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
 ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO
COMMAND 'DUMP '
 ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2012.244 13:42:07 INITIAL SCAN OF
USER CONTROL STATEMENTS COMPLETED
 ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION
INTERFACE
 ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2012.244 13:42:07 EXECUTION BEGINS
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2012.244 13:42:16 EXECUTION ENDS
 ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2012.244 13:42:16 TASK COMPLETED
WITH RETURN CODE 0000
 ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2012.244 13:42:16 DFSMSDSS
PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
 ARC0637I DUMP COPY OF VOLUME SBOXJ8 COMPLETE, DCLASS=DB0BTST1, EXPDT=2012/09/01
 ARC0623I FULL VOLUME DUMP OF VOLUME SBOXJ8 ENDING AT 13:42:17, PROCESSING
SUCCESSFUL
 ARC0622I FULL VOLUME DUMP STARTING ON VOLUME SBOXJ9(SMS) AT 13:42:17 ON
2012/08/31, SYSTEM SC64, TASK ID=ARCDVOL1 ,
  TO DUMP CLASS(ES)=   DB0BTST1
 ARC0728I VTOC FOR VOLUME SBOXJ9 COPIED TO DATA SET
HSM.DUMPVTOC.T553813.VSBOXJ9.D12244 ON VOLUME SBXHS6
 ARC0640I ARCDVOL1 - PAGE 0001    5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 13:42
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC
CLASS CHK DEFAULT TO YES
 ARC0640I ARCDVOL1 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT
DEFAULT TO
 ARC0640I ARCDVOL1 -  DUMP FULL INDDNAME(SYS01494)          -
 ARC0640I ARCDVOL1 -  OUTDDNAME(SYS01488) -
 ARC0640I ARCDVOL1 -  ALLEXCP ALLDATA(*) OPTIMIZE(3) TOLERATE(IOERROR)
 ARC0640I ARCDVOL1 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO
COMMAND 'DUMP '
 ARC0640I ARCDVOL1 - ADR109I (R/I)-RI01 (01), 2012.244 13:42:17 INITIAL SCAN OF
USER CONTROL STATEMENTS COMPLETED
 ARC0640I ARCDVOL1 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION
INTERFACE
 ARC0640I ARCDVOL1 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(01), 2012.244 13:42:17 EXECUTION BEGINS
 ARC0640I ARCDVOL1 - ADR006I (001)-STEND(02), 2012.244 13:42:25 EXECUTION ENDS
 ARC0640I ARCDVOL1 - ADR013I (001)-CLTSK(01), 2012.244 13:42:25 TASK COMPLETED
WITH RETURN CODE 0000
 ARC0640I ARCDVOL1 - ADR012I (SCH)-DSSU (01), 2012.244 13:42:25 DFSMSDSS
PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
 ARC1802I FAST REPLICATION BACKUP DUMP HAS COMPLETED FOR COPY POOL DSN$DB0B$LG, AT
13:42:25 ON 2012/08/31,
  FUNCTION RC=0000, MAXIMUM VOLUME RC=0000, CAPTURE CATALOG RC=0000
 ARC0637I DUMP COPY OF VOLUME SBOXJ9 COMPLETE, DCLASS=DB0BTST1, EXPDT=2012/09/01
 ARC0623I FULL VOLUME DUMP OF VOLUME SBOXJ9 ENDING AT 13:42:34, PROCESSING
SUCCESSFUL
```

Note: When a DFSMShsm Fast Replication backup is initiated by issuing an **FRBACKUP** command to pair a primary volume with a secondary volume in the backup storage group, target volumes that were already processed potentially will be re-examined. Therefore, multiple ARC1809I return code 2 messages result for the same target volume. The suppression of ARC1809I messages is possible with a patch, which means that *all* ARC1809I messages are omitted.

To manage the behavior with the **SETSYS** command instead, use a new **SETSYS** parameter:

`FASTREPLICATION(VOLUMEPAIRMESSAGES(YES|NO))`

Setting this parameter to `YES` will limit the informational message to be issued one time for each target volume for each source storage group. Changing the setting to `NO` causes only one ARC1809I message for any secondary volume.

The current patch to disable the occurrence of ARC1809I messages will continue to be supported.

You can also check the data sets that were backed up by using **FRBACKUP** by using this command:

`LIST COPYPOOL(`*`cpname`*`) DATASETS`

The sample output of the **LIST** command is shown in Example 10-21.

Example 10-21 Sample output of the LIST COPYPOOL(DSN$DB0B$LG) DATASETS command

```
-- DFSMShsm CONTROL DATA SET -- COPY POOL -- LISTING -- AT 14:05:36 ON 12/08/31
FOR SYSTEM=SC64

 COPYPOOL=DSN$DB0B$LG                   ,VER=008,GEN=000,CATINFO=Y
 CATALOG INFORMATION DATA SET NAME=HSM.HSMCIDS.D12244.T133855.C001
 TKN(C)=C'MHLRESA'
 TKN(H)=X'D4C8D3D9C5E2C1'

 CATALOG NAME = UCAT.DB0BLOGS

 DATA SET NAME
 DB0BA.ARCHLOG1.A0000001
 DB0BA.ARCHLOG1.A0000002
 DB0BA.ARCHLOG1.A0000003
 DB0BA.ARCHLOG1.A0000004
 DB0BA.ARCHLOG1.A0000005
 DB0BA.ARCHLOG1.A0000006
 DB0BA.ARCHLOG1.A0000007
 DB0BA.ARCHLOG1.A0000008
 DB0BA.ARCHLOG1.A0000009
 DB0BA.ARCHLOG1.A0000010
 DB0BA.ARCHLOG1.A0000011
 DB0BA.ARCHLOG1.A0000012
 DB0BA.ARCHLOG1.A0000013
 DB0BA.ARCHLOG1.A0000014
 DB0BA.ARCHLOG1.A0000015
 DB0BA.ARCHLOG1.A0000016
 DB0BA.ARCHLOG1.A0000017
 DB0BA.ARCHLOG1.A0000018
 DB0BA.ARCHLOG1.A0000019
```

```
 DB0BA.ARCHLOG1.B0000001
 DB0BA.ARCHLOG1.B0000002
 DB0BA.ARCHLOG1.B0000003
 DB0BA.ARCHLOG1.B0000004
 DB0BA.ARCHLOG1.B0000005
 DB0BA.ARCHLOG1.B0000006
 DB0BA.ARCHLOG1.B0000007
 DB0BA.ARCHLOG1.B0000008
 DB0BA.ARCHLOG1.B0000009
 DB0BA.ARCHLOG1.B0000010
 DB0BA.ARCHLOG1.B0000011
 DB0BA.ARCHLOG1.B0000012
 DB0BA.ARCHLOG1.B0000013
 DB0BA.ARCHLOG1.B0000014
 DB0BA.ARCHLOG1.B0000015
 DB0BA.ARCHLOG1.B0000016
 DB0BA.ARCHLOG1.B0000017
 DB0BA.ARCHLOG1.B0000018
 DB0BA.ARCHLOG1.B0000019
 DB0BA.ARCHLOG2.A0000001
1
 -- DFSMShsm CONTROL DATA SET -- COPY POOL -- LISTING -- AT 14:05:36 ON 12/08/31
FOR SYSTEM=SC64

 CATALOG NAME = UCAT.DB0BLOGS                                (CONTINUED)

 DATA SET NAME
 DB0BA.ARCHLOG2.A0000002
 DB0BA.ARCHLOG2.A0000003
 DB0BA.ARCHLOG2.A0000004
 DB0BA.ARCHLOG2.A0000005
 DB0BA.ARCHLOG2.A0000006
 DB0BA.ARCHLOG2.A0000007
 DB0BA.ARCHLOG2.A0000008
 DB0BA.ARCHLOG2.A0000009
 DB0BA.ARCHLOG2.A0000010
 DB0BA.ARCHLOG2.A0000011
 DB0BA.ARCHLOG2.A0000012
 DB0BA.ARCHLOG2.A0000013
 DB0BA.ARCHLOG2.A0000014
 DB0BA.ARCHLOG2.A0000015
 DB0BA.ARCHLOG2.A0000016
 DB0BA.ARCHLOG2.A0000017
 DB0BA.ARCHLOG2.A0000018
 DB0BA.ARCHLOG2.A0000019
 DB0BA.ARCHLOG2.B0000001
 DB0BA.ARCHLOG2.B0000002
 DB0BA.ARCHLOG2.B0000003
 DB0BA.ARCHLOG2.B0000004
 DB0BA.ARCHLOG2.B0000005
 DB0BA.ARCHLOG2.B0000006
 DB0BA.ARCHLOG2.B0000007
 DB0BA.ARCHLOG2.B0000008
 DB0BA.ARCHLOG2.B0000009
 DB0BA.ARCHLOG2.B0000010
```

```
 DB0BA.ARCHLOG2.B0000011
 DB0BA.ARCHLOG2.B0000012
 DB0BA.ARCHLOG2.B0000013
 DB0BA.ARCHLOG2.B0000014
 DB0BA.ARCHLOG2.B0000015
 DB0BA.ARCHLOG2.B0000016
 DB0BA.ARCHLOG2.B0000017
 DB0BA.ARCHLOG2.B0000018
 DB0BA.ARCHLOG2.B0000019
 DB0BB.BSDS01
 DB0BB.BSDS02
 DB0BL.LOGCOPY1.DS01
 DB0BL.LOGCOPY1.DS02
 DB0BL.LOGCOPY1.DS03
 DB0BL.LOGCOPY2.DS01
 DB0BL.LOGCOPY2.DS02
1
 -- DFSMShsm CONTROL DATA SET -- COPY POOL -- LISTING -- AT 14:05:36 ON 12/08/31
FOR SYSTEM=SC64

 CATALOG NAME = UCAT.DB0BLOGS                                (CONTINUED)

 DATA SET NAME
 DB0BL.LOGCOPY2.DS03

 TOTAL NUMBER OF DATA SETS =         84
```

10.5.4 Perform fast replicate recover (FRRECOV)

You can use the **FRRECOV** command to recover a copy pool or individual volumes and data sets from the managed copy pool copies. The backup copy to be recovered can reside on either DASD or tape. If the backup copy resides on both DASD and tape, the default is to use the DASD backup copy.

To restrict the recovery to backup copy versions that reside on DASD or tape, use the FROMDASD (for DASD) or FROMDUMP (for tape) options. If the backup copy version is not on either DASD or tape, the recovery request fails.

When DATE, GENERATION, TOKEN, or VERSION is specified, the corresponding backup copy is recovered. If no specific backup copy is specified, an attempt to recover generation zero occurs. If no valid backup copy (either the indicated or implicit) is found, on DASD or tape, the recovery request fails.

A specific dump class to recover the version from can be specified when you are recovering from a dump copy on tape. When recovery is performed at the copy pool level, and the dump copy to recover is a partial dump, the recovery request will fail unless the PARTIALOK option is specified.

You can either recover an individual data set or the whole copy pool. We show how to recover for both cases.

Using FRRECOV to recover a data set

You can recover an individual data set by using the **FRRECOV** command. The syntax is shown:

FRRECOV DSNAME(*dsn*) FROMDASD | FROMDUMP

Again, we use **FRRECOV** with a batch job.

In our example, we try to recover data set DB0BA.ARCHLOG1.A0000001 from a full volume dump. The JCL is shown in Example 10-22.

Example 10-22 Sample JCL for a data set recover by using FRRECOV from a dump

```
//MHLRESAY JOB (999,POK),'MHLRES1',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM SYSAFF=*
//HSMFR EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//PRINTER  DD SYSOUT=*
//INDEX    DD SYSOUT=*
//SYSIN    DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
 HSEND FRRECOV DSNAME(DB0BA.ARCHLOG1.A0000001) REPLACE FROMDUMP
```

After the job completes, you can verify the recovery process in the DFSMShsm log, as shown in Example 10-23.

Example 10-23 DFSMShsm log

```
ARC1801I FAST REPLICATION DATA SET RECOVERY IS STARTING FOR DATA SET
DB0BA.ARCHLOG1.A0000001, AT 14:13:45 ON 2012/08/31
 ARC1861I THE FOLLOWING 0001 DATA SET(S) WERE SUCCESSFULLY PROCESSED DURING FAST
REPLICATION DATA SET RECOVERY:
 ARC1861I (CONT.) DB0BA.ARCHLOG1.A0000001, COPYPOOL=DSN$DB0B$LG, DEVTYPE=TAPE
 ARC1802I FAST REPLICATION DATA SET RECOVERY HAS COMPLETED FOR DATA SET
DB0BA.ARCHLOG1.A0000001,
  AT 14:14:06 ON 2012/08/31, FUNCTION RC=0000, MAXIMUM DATA SET RC=0000
```

The recovery activities from the DFSMShsm dump log are shown in Example 10-24.

Example 10-24 DFSMShsm dump log

```
DFSMSHSM DUMP LOG, TIME 13:42:34,  DATE 12/08/31
 ARC0640I GDSN01 - PAGE 0001     5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 14:13
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC
CLASS CHK DEFAULT TO YES
 ARC0640I GDSN01 - ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT
DEFAULT TO
 ARC0640I GDSN01 -  RESTORE DS(INCLUDE(DB0BA.ARCHLOG1.A0000001
)) -
 ARC0640I GDSN01 -  INDDNAME(SYS01524) OUTDYNAM(SBOXJ0,3390    ) REPLACE
CANCELERROR -
 ARC0640I GDSN01 -  BYPASSACS(DB0BA.ARCHLOG1.A0000001                     ) -
 ARC0640I GDSN01 -  STORCLAS(DB0BARCH                          ) CATALOG FORCECP(0) -
 ARC0640I GDSN01 -  MGMTCLAS(MCDB22                            )
 ARC0640I GDSN01 - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO
COMMAND 'RESTORE '
 ARC0640I GDSN01 - ADR109I (R/I)-RI01 (01), 2012.244 14:13:45 INITIAL SCAN OF USER
CONTROL STATEMENTS COMPLETED
 ARC0640I GDSN01 - ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION
INTERFACE
```

```
 ARC0640I GDSN01 - ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS
TASK
 ARC0640I GDSN01 - ADR006I (001)-STEND(01), 2012.244 14:13:45 EXECUTION BEGINS
 ARC0640I GDSN01 - ADR780I (001)-TDDS (01),
  THE INPUT DUMP DATA SET BEING PROCESSED IS IN FULL VOLUME FORMAT AND WAS CREATED
BY DFSMSDSS VERSION 1
 ARC0640I GDSN01 -                          RELEASE 13 MODIFICATION LEVEL 0 ON
2012.244 13:39:32
 ARC0640I GDSN01 - ADR378I (001)-TDDS (01), THE FOLLOWING DATA SETS WERE
SUCCESSFULLY PROCESSED FROM VOLUME SBOXJ0
 ARC0640I GDSN01 -                      DB0BA.ARCHLOG1.A0000001
RESTORED ON SBOXJ0
 ARC0640I GDSN01 - ADR006I (001)-STEND(02), 2012.244 14:14:06 EXECUTION ENDS
 ARC0640I GDSN01 - ADR013I (001)-CLTSK(01), 2012.244 14:14:06 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I GDSN01 - ADR012I (SCH)-DSSU (01), 2012.244 14:14:06 DFSMSDSS PROCESSING
COMPLETE. HIGHEST RETURN CODE IS 0000
  DFSMSHSM MIGRATION LOG, TIME 14:05:00,  DATE 12/08/31
```

Using FRRECOV to recover a copy pool

This option indicates to DFSMShsm to recover all source volumes that are associated with the named copy pool.

The command syntax to use the **FRRECOV** command to recover a copy pool is shown:

FRRECOV COPYPOOL(*cpname*) FROMDASD | FROMDUMP

The sample JCL to recover the copy pool DSN$DB0B$LG is shown in Example 10-25.

Example 10-25 Sample JCL to recover a copy pool by using FRRECOV

```
//MHLRESAR JOB (999,POK),'MHLRES1',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM SYSAFF=*
//HSMFR EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//PRINTER  DD SYSOUT=*
//INDEX    DD SYSOUT=*
//SYSIN    DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
 HSEND FRRECOV CP(DSN$DB0B$LG) FROMDASD
```

Before the recovery of a copy pool, DFSMShsm deallocates the specified catalogs. If DFSMShsm cannot deallocate any of the catalogs, the **FRRECOV** operation might fail. **CATALOG** reallocates the catalogs when it is used. Example 10-26 shows that the system automatically deallocates the catalog that is involved.

Example 10-26 Syslog activities for FRRECOV

```
STC28942 00000290  F CATALOG,UNALLOCATE(UCAT.DB0BLOGS
INTERNAL 00000090  IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
         00000290  IEF196I IGD104I UCAT.DB0BLOGS
         00000290  IEF196I DDNAME=SYS00177
INTERNAL 00000090  IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
STC28942 00000090  IGD103I SMS ALLOCATED TO DDNAME SYS01528
```

```
STC28942 00000090  IGD103I SMS ALLOCATED TO DDNAME SYS01529
STC28942 00000090  IGD103I SMS ALLOCATED TO DDNAME SYS01530
STC28942 00000090  IGD103I SMS ALLOCATED TO DDNAME SYS01531
STC28942 00000090  ARC1805I THE FOLLOWING 00004 VOLUME(S) WERE 945
     945 00000090  ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST REPLICATION R
     945 00000090  ARC1805I (CONT.) OF COPY POOL DSN$DB0B$LG
STC28942 00000090  ARC1805I (CONT.) SBOXJ0
STC28942 00000090  ARC1805I (CONT.) SBOXJ1
STC28942 00000090  ARC1805I (CONT.) SBOXJ8
STC28942 00000090  ARC1805I (CONT.) SBOXJ9
STC28942 00000090  ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR 950
     950 00000090  ARC1802I (CONT.) COPY POOL DSN$DB0B$LG, AT 14:26:12 ON 2012/0
     950 00000090  ARC1802I (CONT.) FUNCTION RC=0000, MAXIMUM VOLUME RC=0000
```

When the recovery process completes, you can also verify it in the DFSMShsm log, as shown in Example 10-27.

Example 10-27 DFSMShsm backlog activities for the copy pool recover with FRRECOV

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR COPY POOL DSN$DB0B$LG, AT
14:26:10 ON 2012/08/31
 ARC0640I ARCFRTM - PAGE 0001    5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 14:26
 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC
CLASS CHK DEFAULT TO YES
 ARC0640I ARCFRTM -  PARALLEL
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO
COMMAND 'PARALLEL '
 ARC0640I ARCFRTM -  COPY IDY(SBOXK0) ODY(SBOXJ0) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(SBOXK1) ODY(SBOXJ1) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(SBOXKA) ODY(SBOXJ8) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 004 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM -  COPY IDY(SBOXKB) ODY(SBOXJ9) DUMPCOND FR(REQ) PUR ALLX
ALLD(*)      -
 ARC0640I ARCFRTM -  DEBUG(FRMSG(DTL))
 ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 005 HAS BEEN ASSIGNED TO
COMMAND 'COPY '
 ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2012.244 14:26:10 INITIAL SCAN OF
USER CONTROL STATEMENTS COMPLETED
 ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
  2012.244 14:26:10 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN
EFFECT
 ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
```

```
 ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR050I (004)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (004)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR050I (005)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY
APPLICATION INTERFACE
 ARC0640I ARCFRTM - ADR016I (005)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR
THIS TASK
 ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2012.244 14:26:10 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (002)-T0MI (02), VOLUME SBOXK0 WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2012.244 14:26:10 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2012.244 14:26:10 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - ADR006I (004)-STEND(01), 2012.244 14:26:10 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (004)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (004)-T0MI (02), VOLUME SBOXKA WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (004)-STEND(02), 2012.244 14:26:10 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (004)-CLTSK(01), 2012.244 14:26:11 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2012.244 14:26:10 EXECUTION BEGINS
 ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (003)-T0MI (02), VOLUME SBOXK1 WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2012.244 14:26:10 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2012.244 14:26:11 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - ADR006I (005)-STEND(01), 2012.244 14:26:10 EXECUTION BEGINS
 ARC0640I ARCFRTM - PAGE 0002    5695-DF175  DFSMSDSS V1R13.0 DATA SET SERVICES
2012.244 14:26
 ARC0640I ARCFRTM - ADR241I (005)-DDTFP(01), TARGET VTOC BEGINNING AT 000000005:00
AND ENDING AT 000000016:14 IS OVERLAID
 ARC0640I ARCFRTM - ADR806I (005)-T0MI (02), VOLUME SBOXKB WAS COPIED USING A FAST
REPLICATION FUNCTION
 ARC0640I ARCFRTM - ADR006I (005)-STEND(02), 2012.244 14:26:10 EXECUTION ENDS
 ARC0640I ARCFRTM - ADR013I (005)-CLTSK(01), 2012.244 14:26:12 TASK COMPLETED WITH
RETURN CODE 0000
 ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2012.244 14:26:12 DFSMSDSS PROCESSING
COMPLETE. HIGHEST RETURN CODE IS 0000
 ARC0400I VOLUME SBOXJ0 IS 96% FREE, 0000000014 FREE TRACK(S), 000029087 FREE
CYLINDER(S), FRAG .000
 ARC0401I LARGEST EXTENTS FOR SBOXJ0 ARE CYLINDERS     29087, TRACKS     436315
 ARC0402I VTOC FOR SBOXJ0 IS 00180 TRACKS(0009000 DSCBS), 0008959 FREE DSCBS(99%
OF TOTAL)
```

```
 ARC0400I VOLUME SBOXJ1 IS 96% FREE, 0000000014 FREE TRACK(S), 000028991 FREE
CYLINDER(S), FRAG .000
 ARC0401I LARGEST EXTENTS FOR SBOXJ1 ARE CYLINDERS     28991, TRACKS     434875
 ARC0402I VTOC FOR SBOXJ1 IS 00180 TRACKS(0009000 DSCBS), 0008957 FREE DSCBS(99%
OF TOTAL)
 ARC0400I VOLUME SBOXJ8 IS 97% FREE, 0000000005 FREE TRACK(S), 000009727 FREE
CYLINDER(S), FRAG .000
 ARC0401I LARGEST EXTENTS FOR SBOXJ8 ARE CYLINDERS      9727, TRACKS     145905
 ARC0402I VTOC FOR SBOXJ8 IS 00180 TRACKS(0009000 DSCBS), 0008989 FREE DSCBS(99%
OF TOTAL)
 ARC0400I VOLUME SBOXJ9 IS 98% FREE, 0000000005 FREE TRACK(S), 000009847 FREE
CYLINDER(S), FRAG .000
 ARC0401I LARGEST EXTENTS FOR SBOXJ9 ARE CYLINDERS      9847, TRACKS     147705
 ARC0402I VTOC FOR SBOXJ9 IS 00180 TRACKS(0009000 DSCBS), 0008991 FREE DSCBS(99%
OF TOTAL)
 ARC1805I THE FOLLOWING 00004 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST
REPLICATION RECOVERY OF COPY POOL DSN$DB0B$LG
 ARC1805I (CONT.) SBOXJ0
 ARC1805I (CONT.) SBOXJ1
 ARC1805I (CONT.) SBOXJ8
 ARC1805I (CONT.) SBOXJ9
 ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR COPY POOL DSN$DB0B$LG, AT
14:26:12 ON 2012/08/31, FUNCTION RC=0000,
  MAXIMUM VOLUME RC=0000
```

11

Maintaining your DFSMShsm environment

In this chapter, we describe the main procedures that are performed by a storage manager regularly. These procedures include holding and releasing DFSMShsm functions, canceling tasks, and changing DFSMShsm control parameters, among other activities.

11.1 Maintaining DFSMShsm

To keep DFSMShsm up and running, the storage administrator needs to perform these administrative activities regularly:

- Holding DFSMShsm functions
- Releasing DFSMShsm functions
- Canceling DFSMShsm requests
- Changing DFSMShsm control parameters
- Restarting DFSMShsm after an abnormal end
- Expiring old data set backup versions
- Recycling tapes and converting to new tape volumes
- Moving data sets to new DASD volumes
- Reorganizing the control data sets (CDSs)
- Auditing DFSMShsm

All of these storage manager activities are described. The main DFSMShsm management concepts and examples are explained.

For detailed information about setting up your product, see Chapter 3, “Getting started” on page 29.

11.1.1 Holding DFSMShsm functions

The **HOLD** command is used to selectively prevent or interrupt DFSMShsm functions from running without stopping DFSMShsm. You can use the command to prevent tape-related processing if, for example, you are drive-constrained at a particular moment or hardware errors exist.

For functions that process on a volume basis (backup, dump, migration, and recover), you can choose whether you want to interrupt the function at the end of the data set that is being processed or at the end of the volume that is being processed.

The **HOLD** command attempts to terminate the process at a reasonable point, so that the function can be restarted or reissued later. A function is held until it is released or DFSMShsm is shut down and restarted. Depending on the **HOLD** command that you issue, you can prevent processing of all DFSMShsm functions or selectively choose the following functions to hold:

- Command
- Automatic primary space management
- Automatic secondary space management
- Common queue
- Fast replication backup
- Fast replication recovery
- Recall or deletion of a migrated data set
- Command backup
- Automatic backup
- Aggregate backup
- Aggregate recovery
- Command dump
- Automatic dump
- Audit command processing
- List
- Report
- Recovery, and restore

- Recycle
- Logging
- Tape copy
- Tape replace
- Expiring backup versions

The **HOLD** command offers flexibility. The **HOLD** commands that you will use most are described. For more details about the **HOLD** command, see the *DFSMShsm Storage Administration Guide,* SC26-0421.

Before you submit any DFSMShsm **HOLD** commands, you must ensure that you defined the necessary RACF or AUTH settings to your user ID. For more details about this access, see Chapter 5, “Implementing DFSMShsm security” on page 79.

You can issue all of the **HOLD** commands through TSO by issuing the **HSEND** command, or through the console with the following command:

```
F hsmtask,command
```

As you plan your DFSMShsm daily processing window, DFSMShsm activities might compete for the same resources, such as DASD, data sets, tape drives, or processor time. By holding the conflicting task instead of shutting down DFSMShsm, you can give DFSMShsm a chance to finish the current data set or volume processing before processing stops, preventing CDS mismatches and data loss.

All functions that are held by command are automatically released after DFSMShsm is shut down and restarted. If you want DFSMShsm to start with any specific function held, you can add the correct **HOLD** command in the ARCCMDxx member in the DFSMShsm PARMLIB.

If you are in the middle of space management processing, and only a few tape drives are available for use by batch processing, it might be necessary to stop your current space management task to free extra drives so batch processing is not affected. Hold MIGRATION, so DFSMShsm will stop the space management task after it finishes the current data set that is being processed by issuing one of the commands that are shown in Example 11-1.

Example 11-1 Preventing automatic migration

```
HSEND HOLD MIGRATION(AUTO)
HSEND HOLD AUTOMIGRATION
```

The commands that are shown in Example 11-1 do not prevent users from manually executing migration tasks.

To prevent or hold both user and automatic migration, you can issue the following command:

```
HSEND HOLD MIGRATION
```

Important: Every time a main function, such as migration, is held, all subfunctions are held also. In this case, both automatic and command migrations are held.

If processing peaks are caused by DFSMShsm tasks, check the specific task that is causing the peaks, and then hold it. If you are uncertain which task is in error, another approach is to hold all tasks, and then release one task at a time, until you can determine the task in error. The following command holds all DFSMShsm tasks, except for the common recall queue (CRQ):

```
HSEND HOLD ALL
```

To hold the common recall queue, issue the following command:

```
HSEND HOLD COMMONQUEUE
```

For certain functions, you can control whether you want the process to stop after the current data set processing, or after current volume processing. With the following functions, you can make this selection:

- Space management (automatic or command)
- Backup (automatic or command)
- Dump (automatic or command)
- Recover
- Restore

If you want to stop HSM task processing after the current volume processing ends, use the **HOLD** command and the **ENDOFVOLUME** parameter. This command holds the dump after the current dumps that are processing finish dumping the volume:

```
HSEND HOLD DUMP ENDOFVOLUME
```

You can hold a task after the current data set is processed. Issue the following command to hold the dump after the current dumps that are processing finish dumping the data set:

```
HSEND HOLD DUMP ENDOFDATASET
```

> **Note:** If neither the **ENDOFDATASET** nor **ENDOFVOLUME** parameter is specified, the default value is **ENDOFDATASET** to space management, backup, recover, and restore functions, and **ENDOFVOLUME** to dump, FRBACKUP dump, and FRBACKUP dumponly functions.

You might need to hold all processing from accessing a specific type of media. In these scenarios, you can hold all DFSMShsm tasks that use the unavailable device. If you want to prevent tape access due to scheduled maintenance, link issues, or hardware failures, you can issue several of the following commands to stop all tape processing and still allow users to access data in other media, as shown in Example 11-2.

Example 11-2 Holding DFSMShsm functions separately

```
HSEND HOLD BACKUP(AUTO DSCOMMAND(TAPE))
HSEND HOLD MIGRATION
HSEND HOLD RECALL(TAPE)
HSEND HOLD RECYCLE
HSEND HOLD TAPECOPY
HSEND HOLD TAPEREPL
HSEND HOLD RECOVER(TAPEDATASET)
HSEND HOLD FRRECOV(TAPE)
HSEND HOLD COMMONQUEUE
HSEND HOLD DUMP
HSEND HOLD ABACKUP
HSEND HOLD ARECOVER
```

> **Note:** You might not need to issue all of these commands in your environment to prevent tape processing. For information about the tasks to hold, see the *DFSMShsm Storage Administration Guide,* SC26-0421.

11.1.2 Releasing DFSMShsm functions

Any function that can be held can be released with the **RELEASE** command. You are able to release all DFSMShsm tasks that you can put on hold. All of the tasks have the same name and parameters for both hold and release.

The only time that functions will not be released is if journaling was disabled. After journaling is disabled, it holds all DFSMShsm functions. The **RELEASE** command will not be effective until the CDSs are successfully backed up.

To check whether any functions are on hold, issue the following command from TSO or the console. This command queries active requests:

```
HSEND QUERY ACTIVE
```

The command that is shown in Example 11-3 displays all of the DFSMShsm task statuses, including those tasks that are on hold.

Example 11-3 Output from the QUERY ACTIVE command

```
ARC0101I QUERY ACTIVE COMMAND STARTING ON HOST=2
ARC0144I AUDIT=NOT HELD AND INACTIVE, LIST=NOT HELD AND INACTIVE, RECYCLE=HELD
ARC0144I (CONT.) AND INACTIVE, REPORT=NOT HELD AND INACTIVE
ARC0160I MIGRATION=NOT HELD, AUTOMIGRATION=HELD, RECALL=NOT HELD,
ARC0160I (CONT.) TAPERECALL=NOT HELD, DATA SET MIGRATION=INACTIVE, VOLUME
ARC0160I (CONT.) MIGRATION=INACTIVE, DATA SET RECALL=INACTIVE
ARC0163I BACKUP=NOT HELD, AUTOBACKUP=HELD, RECOVERY=NOT HELD,
ARC0163I (CONT.) TAPEDATASETRECOVERY=NOT HELD, DATA SET BACKUP=NOT HELD, VOLUME
ARC0163I (CONT.) BACKUP=INACTIVE, DATA SET RECOVERY=INACTIVE, VOLUME
ARC0163I (CONT.) RECOVERY=INACTIVE
ARC0276I DATA SET BACKUP=INACTIVE, DATA SET BACKUP ACTUAL IDLETASKS=(ALLOC=00,
ARC0276I (CONT.) MAX=00)
```

You can see several of the DFSMShsm functions, including RECYCLE, which is held.

If the reason that the task was held is no longer valid, such as the hardware maintenance finishes, the link was reestablished, or the processor usage decreased, you can release the recycle task with the following command:

```
HSEND RELEASE RECYCLE
```

If you set a main function on hold, such as migration, you cannot release only a subfunction of this task, such as automatic migration. Example 11-4 shows you the results of trying to release a subfunction of a held task.

Example 11-4 Holding and releasing functions and subfunctions

```
F HSM1,HOLD BACKUP
ARC0100I HOLD COMMAND COMPLETED
F HSM1,RELEASE AUTOMIGRATION
ARC0111I SUBFUNCTION MIGRATION(AUTO) CANNOT BE  664
ARC0111I (CONT.) RELEASED WHILE MAIN FUNCTION MIGRATION IS HELD
ARC0100I RELEASE COMMAND COMPLETED
```

To release only automatic space management, you must release backup, and then hold only the user command **BACKUP**, as shown in Example 11-5 on page 324.

Example 11-5 Holding and releasing functions and subfunctions

```
HSEND RELEASE BACKUP
HSEND HOLD BACKUP(DSCOMMAND)
```

Many variations of the **RELEASE** command exist. For more information and a comprehensive list of the commands, see *DFSMShsm Storage Administration Reference,* SC26-0422.

11.1.3 Canceling queued DFSMShsm requests

Use DFSMShsm to cancel queued and active requests. A *queued request* is a request that is not yet selected for processing by DFSMShsm. After a request is selected for processing, it becomes an active request. Even though it can be canceled, we do not recommend canceling it, except to prevent outages or shutting down DFSMShsm.

Several ways exist to cancel DFSMShsm requests that are waiting in the queue. You can decide the best way to cancel all of the requests that you need to cancel without affecting other users or requests.

Before you cancel a request, you might want to display the requests that are in the queue, and then decide which requests to cancel. You can issue the following command to get a list of waiting requests:

```
HSEND QUERY REQUEST
```

Important: If many requests are waiting in the queue, your window might be flooded with requests. You can issue the **HSEND QUERY WAIT** command to check how many requests are in the queue before you display the queue.

Example 11-6 is a sample display from the command.

Example 11-6 Output from the QUERY REQUEST command

```
ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=2
ARC0167I MIGRATE MWE FOR DATA SET MHLRES7.ACSTEST.LISTDS FOR USER MHLRES7,
ARC0167I (CONT.) REQUEST 00000013, WAITING TO BE PROCESSED, 00000 MWE(S) AHEAD
ARC0167I (CONT.) OF THIS ONE
ARC0167I MIGRATE MWE FOR DATA SET MHLRES7.BV FOR USER MHLRES7, REQUEST
ARC0167I (CONT.) 00000014, WAITING TO BE PROCESSED, 00001 MWE(S) AHEAD OF THIS
ARC0167I (CONT.) ONE
ARC0167I MIGRATE MWE FOR DATA SET MHLRES7.BV.THURSDAY FOR USER MHLRES7,
ARC0167I (CONT.) REQUEST 00000015, WAITING TO BE PROCESSED, 00002 MWE(S) AHEAD
ARC0167I (CONT.) OF THIS ONE
```

DFSMShsm returns three lines of response to every request on the queue. This response includes the task to be executed, the data set to be processed, the user who issued the request, the request ID, and how many requests are ahead in the queue.

Now that you have all of the necessary information about the waiting requests, you can define the best way to cancel the requests without affecting other requests.

One way to cancel requests is by specifying the request ID when you issue the **CANCEL** command. This method cancels only the exact specified request. The other requests are unchanged. The following example cancels a specific request:

```
HSEND CANCEL REQ(14)
```

DFSMShsm returns the following output, as shown in Example 11-7.

Example 11-7 Output from the CANCEL REQ command

```
ARC1008I MHLRES7.BV MIGRATE REQUEST 00000014 WAS CANCELLED
ARC0931I (H)CANCEL COMMAND COMPLETED, NUMBER OF REQUESTS CANCELLED=1
ARC1007I COMMAND REQUEST 00000031 SENT TO DFSMSHSM
```

Another way to cancel a request is by specifying the exact data set name for the request:

```
HSEND CANCEL DATASETNAME(mhlres7.bv.thursday)
```

This **HSEND CANCEL** command produces the output that is shown in Example 11-8.

Example 11-8 HSEND CANCEL command output

```
ARC1008I MHLRES7.BV.THURSDAY MIGRATE REQUEST 00000015 WAS CANCELLED
ARC0931I (H)CANCEL COMMAND COMPLETED, NUMBER OF REQUESTS CANCELLED=1
ARC1007I COMMAND REQUEST 00000033 SENT TO DFSMSHSM
```

Another way to cancel requests is by identifying the user ID that issued the request. Canceling requests by identifying a user ID will cancel all requests from that user, as shown in the following example:

```
HSEND CANCEL USERID(mhlres7)
```

The command in Example 11-9 will cancel all requests that are associated with user ID MHLRES7.

Example 11-9 HSEND CANCEL output

```
ARC1008I MHLRES7.BVDEL MIGRATE REQUEST 00000016 WAS CANCELLED
ARC1008I MHLRES7.BVDEX MIGRATE REQUEST 00000017 WAS CANCELLED
ARC1008I MHLRES7.CMD.CLIST MIGRATE REQUEST 00000018 WAS CANCELLED
ARC1008I MHLRES7.EXEC.RMM.CLIST MIGRATE REQUEST 00000019 WAS CANCELLED
ARC1008I MHLRES7.EXEC.RMM.HOLDX.CLIST MIGRATE REQUEST 00000020 WAS CANCELLED
ARC1008I MHLRES7.HSM.BCDS MIGRATE REQUEST 00000021 WAS CANCELLED
ARC1008I MHLRES7.HSM.DUMPVOL MIGRATE REQUEST 00000022 WAS CANCELLED
ARC1008I MHLRES7.HSM.DVOL MIGRATE REQUEST 00000023 WAS CANCELLED
ARC1008I MHLRES7.HSM.KSDS1 MIGRATE REQUEST 00000024 WAS CANCELLED
ARC0931I (H)CANCEL COMMAND COMPLETED, NUMBER OF REQUESTS CANCELLED=9
ARC1007I COMMAND REQUEST 00000034 SENT TO DFSMSHSM
```

You can also cancel active requests. However, this method is not recommended, unless to avoid outages and DFSMShsm shutdown. For more information about canceling active tasks, see Chapter 14, “Problem determination” on page 401.

11.1.4 Changing DFSMShsm control parameters

Each time that you start DFSMShsm, a subset of parameters is established, by default. You might want to change the defaults during normal operation, change the times that the automatic functions are scheduled to run, or increase the number of tasks that relate to a specific function.

Use the **SETSYS** command to dynamically change the automatic functions without requiring you to restart DFSMShsm. The **SETSYS** parameters control the main DFSMShsm functions and schedule. Ensure that only storage administrators can access these commands. For more information about how to control user access to DFSMShsm resources, see Chapter 5, "Implementing DFSMShsm security" on page 79.

In certain situations, you might need to shrink your space management, or dump window, or move it to a later time to prevent it from affecting the production environment. In these cases, you can issue the following command to alter your primary space management window. This command sets DFSMShsm parameters:

```
HSEND SETSYS PRIMARYSPMGMSTART(0400 0500)
```

This command sets the primary space management window to start at 04:00 and finish at 05:00.

If all of the processing is not complete by 05:00, DFSMShsm shuts down primary space management and issues the following message:

```
ARC0521I PRIMARY SPACE MANAGEMENT ENDING TIME REACHED
```

For more information about how to set up the space management window, see Chapter 9, "Space management" on page 211.

For automatic backup, and automatic dump processing, you must specify one more value, which is the latest time that DFSMShsm can start the backup or dump process. If you do not specify this value, it is set to midnight (24:00). The following command sets DFSMShsm parameters:

```
HSEND SETSYS AUTOBACKUPSTART(0600 0700 0800)
```

This command sets DFSMShsm to start automatic backup at 06:00. If automatic backup is not started until 07:00, which is our late start time, DFSMShsm will not perform automatic backup in this cycle. The value of 08:00 is the deadline for automatic backups, so no new backup will start after this time. Any backups that are currently running at 08:00 are not affected.

In addition to changing primary DFSMShsm function windows, you can also define several configurations to DFSMShsm, including how many instances of a task can run concurrently, how many CDS backup versions to keep, monitoring DFSMShsm CDSs, monitoring control tape units, and so on.

During your maintenance window, you might decide that it is necessary to reduce the number of tape migration tasks so more tape drives will be available for use by other applications. Instead of canceling or holding your space management task, you can reduce the maximum number for migration tasks that are running concurrently in your system. Reducing the maximum does not completely stop your migration and reduces the chance of not completing the task within the window.

To change the number of migration tasks that are running concurrently to three, you can issue the following command from TSO or the console:

```
HSEND SETSYS MAXMIGRATIONTASKS(3)
```

If more than three tasks are running when you issue the command, DFSMShsm first finishes the current instances, and does not start new instances until it reaches the specified number of tasks that are running.

You can also control the maximum number of concurrent tasks that are running for the following functions:

- Aggregate backup and recovery support (ABARS)
- Backup
- Dump
- Fast replication
- Recall
- Recover
- Interval migration
- Secondary space management
- Recycle

Example 11-10 shows examples of how to set the maximum number of instances that are running for each task.

Example 11-10 Setting DFSMShsm parameters

```
HSEND SETSYS MAXABARSADDRESSSPACE(3)
HSEND SETSYS MAXBACKUPTASKS(3)
HSEND SETSYS MAXDUMPTASKS(3)
HSEND SETSYS MAXCOPYPOOLTASKS(FRBACKUP(3))
HSEND SETSYS MAXRECALLTASKS(3)
HSEND SETSYS MAXDSRECOVERTASKS(3)
HSEND SETSYS MAXINTERVALTASKS(3)
HSEND SETSYS MAXSSMTASKS(3)
HSEND SETSYS MAXRECYCLETASKS(3)
```

Important: The values in Example 11-10 are examples only and must not be considered as recommended values.

The **SETSYS** command has a wide range of parameters and possible configurations that are not covered. To learn more about the **SETSYS** control parameter, see *DFSMShsm Storage Administration Reference,* SC26-0422.

11.1.5 Restarting DFSMShsm after an abnormal end

DFSMShsm automatically recovers after most abnormal terminations. However, if DFSMShsm cannot recover from the abnormal end and the **RESTART** keyword was used in the PROC statement of the startup procedure, DFSMShsm will restart itself. With the **RESTART** keyword, you can specify a startup procedure, which can be the same or different from the original startup procedure, and pass additional parameters to DFSMShsm.

If an abnormal end occurs that interrupts MVS processing and on the condition that the extended common service area (ECSA) is not destroyed, DFSMShsm can continue to process waiting requests. Additionally, if DFSMShsm is restarted, it will process any requests that are waiting in the ECSA.

If a task that was processing within the DFSMShsm address or storage space ended abnormally, the following message is issued:

```
ARC0003I taskname TASK ABENDED, CODE abendcode IN MODULE modname AT OFFSET offset,
STORAGE LOCATION location
```

Try to understand the area of processing that was affected. DFSMShsm attempts recovery in most cases, but the ultimate responsibility lies with the user to determine whether any further recovery actions are required.

You need these valuable sources of information to analyze any problems:

- DFSMShsm activity logs
- DFSMShsm Problem Determination Aid (PDA) trace
- DFSMShsm-generated dumps
- System log

If a process, such as primary space management, secondary space management, automatic backup, or automatic dump processing, ends abnormally, you might need to extend the processing window to allow all of the work to complete. You can issue the commands that are shown in Example 11-11 from TSO or the console to change the end time of your tasks, and restart it.

Example 11-11 Setting DFSMShsm parameters

```
SETSYS PRIMARYSPMGMTSTART (starttime,endingtime)
SETSYS SECONDARYSPMGMTSTART (starttime,endingtime)
SETSYS AUTOBACKUPSTART (starttime,latestarttime,quiescetime)
SETSYS AUTODUMPSTART (starttime,latestarttime,quiescetime)
```

Note: Use the same start time that is used in your PROCLIB. Update *endingtime*, *latestarttime*, and *quiescetime* to values that will accommodate your workload.

After the processing finishes, you can issue the **SETSYS** commands again by using the same parameters that are used in ARCCMDxx to ensure that DFSMShsm will use the standard values on the next cycle.

For more information about abnormal ends and problem determination, see Chapter 14, “Problem determination” on page 401.

11.1.6 Expiring backup versions

Availability management of DFSMShsm does not automatically delete backup versions when the related user data set is deleted. Over time, a number of unneeded backup versions can accumulate on backup volumes. You can use the **EXPIREBV** command to identify and delete these unnecessary backup versions based on the most recent status of the data set.

The **EXPIREBV** command causes DFSMShsm to search the backup control data set (BCDS) for old, unwanted backup versions and to delete them (or display information about them) based on the attributes in the management class for each data set.

You can use the **EXPIREBV** command with the **DISPLAY** parameter to see which backup copies are eligible for deletion. The command in Example 11-12 places a list of the eligible backup versions into the specified output data set.

Example 11-12 Displaying old backup versions that are available for deletion

```
HSEND EXPIREBV DISPLAY ODS('output')
```

If you browse the data set pointed to by the **OUTDATASET** parameter, you see the output from this command, as shown in Example 11-13 on page 329.

Example 11-13 HSEND EXPIREBV output

```
DISPLAY OF BACKUP VERSIONS ELIGIBLE FOR EXPIRATION AT 23:46:06 ON 2012/08/21 FOR

COMMAND INPUT: (DEFAULTS)

DSNAME = MHLRES4.SC64.SPFLOG5.LIST                  DELETED*,          WAS SMS
  (* DETERMINED ON 2012/08/21)
MANAGEMENT CLASS USED = MCDB22

BACKUP VERSION DSNAME                         SYS  GEN       RET  BACK
                                              CAT  NMBR AGE  VERS PROF
HSM.BACK.T031418.MHLRES4.SC64.A0313           YES  000  651  NO   NO

DSNAME = MHLRES4.SC70.SPFLOG1.LIST                  DELETED*,          WAS SMS
  (* DETERMINED ON 2012/08/21)
MANAGEMENT CLASS USED = MCDB22

BACKUP VERSION DSNAME                         SYS  GEN       RET  BACK
                                              CAT  NMBR AGE  VERS PROF
HSM.BACK.T021418.MHLRES4.SC70.A0313           YES  000  651  NO   NO
```

To expire the backup versions, issue the following command to delete the expired backup versions:

```
HSENDCMD EXPIREBV EXECUTE
```

The output from the **EXPIREBV EXECUTE** command is in the backup activity log.

Use **EXPIREBV** to define how long you want to keep backups of non-SMS-managed data sets after they are deleted. You can use the **NONSMSVERSIONS(CATALOGEDDATA(*xx*))** parameter to specify that the backups will be kept for *xx* days after the non-SMS-managed cataloged data set is deleted. You can also control backup versions for uncataloged non-SMS data sets. In this case, you use the **UNCATALOGEDDATA(*xx*)** parameter.

The following command shows you how to expire any backup version non-SMS-managed data sets. This example will expire all backups for cataloged data sets 30 days after its deletion, and 20 days after uncataloged data sets are deleted. Use this command to delete expired backup versions:

```
HSEND EXPIREBV NONSMSVERSIONS(CATALOGEDDATA(30) UNCATALOGEDDATA(20)) EXECUTE
```

If you do not specify the retention periods for cataloged non-SMS-managed data sets, DFSMShsm will use 60 days as the default value. If you do not specify **CATALOGEDDATA**, DFSMShsm will not process cataloged non-SMS expiration. The same rule applies to **UNCATALOGEDDATA**.

When you issue the **EXPIREBV** command, DFSMShsm determines that the user data set was scratched and stores the date of the EXPIREBV execution as the scratch date in the DFSMShsm BCDS record (MCB). You can look at the date in the DFSMShsm BCDS record by using the following command:

```
HSEND FIXCDS B 'dsname' OUTDATASET('output')
```

In the data set pointed to by the **OUTDATASET** parameter, you will see the MCB record for the data set (see Example 11-14). The date EXPIREBV determined as the date on which the user data set was deleted is at offset X'48'. A subsequent **EXPIREBV EXECUTE** command deletes the unwanted backup versions when the requested time passes. The requested time is specified in the management class for storage management subsystem (SMS) data sets, or in the **EXPIREBV** command for non-SMS-managed data sets.

Example 11-14 HSEND FIXCDS output

```
FIXCDS B 'MHLRES4.SC70.SPFLOG1.LIST' ODS('MHLRES7.OUTPUT.FIXCDS')
MCH=   00D02000 CA0EE734 72CDA86B CA0EE65C CE2883EB
+0000  E3C8E2F0 F0F0FFFF 00000000 00000000 13312752 0112015F 40006D5E 00500000
+0020  00000001 00005E03 0000000C 0000A000 00000001 0001FFFF 0112235C 00000000
+0040  00000000 00000000 0112235F 00000000 C8E2D44B C2C1C3D2 4BE3F2F7 F3F1F1F3
+0060  4BD4C8D3 D9C5E2F4 4BE2C3F7 F04BC1F2 F0F1F540 40404040 40404040 D4D3C4F1
+0080  F0C35200 0112015F 00000000 00000000
ARC0197I TYPE B, KEY MHLRES4.SC70.SPFLOG1.LIST, FIXCDS DISPLAY SUCCESSFUL
```

We recommend that you run the **EXPIREBV EXECUTE** command regularly.

If you use ABARS in your environment, you also need to issue a specific **EXPIREBV** command for ABARS versions to ensure that old ABARS versions are deleted following “retention only version” and “retention extra version” parameters in the ABARS management class.

The following command deletes any ABARS data sets that are older than the retention that is specified in the management class. Use the following command to delete expired ABARS backup versions:

```
HSEND EXPIREBV ABARSVERSIONS EXECUTE
```

Note: You must run a specific EXPIREBV for ABARS. It is mutually exclusive with SMS, and NONSMSVERSIONS.

In certain situations, you need to delete a valid backup version, such as a missing or broken tape. You can delete valid backup versions of specific data sets by issuing the **BDELETE** or **HBDELETE** command.

The following command can delete the specified backup version 001 of *dsname*.

```
HSEND BDELETE 'dsname' VERSION(001)
```

If you intend to delete all backup versions of a specific data set, you can also issue the following command:

```
HSEND BDELETE 'dsname' ALL
```

11.1.7 Recycling tape volumes

Over time, migrated data sets expire, are recalled, or are marked for deletion by **DELETE** and **HDELETE** commands. Similarly, tape backup data sets are rolled off by automatic backup, or they are marked for deletion by **BDELETE**, **HBDELETE**, and **EXPIREBV** commands. Those data sets are still physically occupying space on the tape volumes. The percent data valid in a DFSMShsm tape tends to decrease as backups and data sets are deleted and the data in the tape is invalidated, resulting in inefficiently used tape media.

Also, migration tapes can be requested for a recall during the migration process. In this case, DFSMShsm allocates another tape for migration, and releases the current tape to proceed with recall. Because the first tape was not fully filled, it is called a *partial tape*.

Partial tapes can be selected by DFSMShsm to continue the migration process, but many partial tapes mean much unused space. To reduce the number of partial tapes, remove invalid data sets, and consolidate the valid data on fewer tape volumes. You can use the **RECYCLE** command.

We describe the most common commands that you can use to consolidate the DFSMShsm-owned tape data. The **RECYCLE** command can be issued from an operator console or by using the **HSEND** command. With the **HSEND** command, you can issue the **RECYCLE** command from any TSO user ID that was authorized through RACF, or the **AUTH** command.

Before you submit the **RECYCLE** command for processing, we recommend that you display a list of tapes that match the requirements to be recycled. The following command displays DFSMShsm tapes that are available for recycle:

```
HSEND RECYCLE DISPLAY ALL ODS('dsname')
```

The display parameter causes DFSMShsm to list the tapes that are selected for recycle, but not to execute it. Use **ODS** to direct the output of the command to a data set that you can access later.

You can also select the tapes that will be recycled, including the following types:

- ALL
- BACKUP
- DAILY
- SPILL
- Migration-level 2 (ML2)

Each option gives you the ability to select only a specific group of tapes to recycle. This approach can be used if you have a large tape environment, and recycling all eligible tapes at one time might cause the command to run for an extended period, or affect the number of tape drives that are available to other applications.

You can also limit the number of concurrent RECYCLE tasks that are running with the **SETSYS MAXRECYCLETASKS** parameter. With the **SETSYS MAXRECYCLETASKS** parameter, you can dynamically change the number of recycle tape processing tasks, even during recycle processing, to any number 1 - 15. The default for recycle-processing tasks is 2. This parameter can be added to the ARCCMDxx member and issued from TSO or the console for special processing requirements. The following command sets the maximum number of recycle tasks to 3:

```
HSEND SETSYS MAXRECYCLETASKS(3)
```

Note: The previous command limits the maximum number of recycle tasks to 3. Remember that each instance of recycle uses two tape drives: one tape drive for input and other tape drive for output.

You can specify to DFSMShsm when a tape becomes eligible for recycle processing by setting a threshold value on the **SETSYS RECYCLEPERCENT** and **ML2RECYCLEPERCENT** parameters in the ARCCMDxx member. You can also specify a value for the **PERCENTVALID** subparameter of the **RECYCLE** command if you need to override the **SETSYS** values. If you do not set any of these parameters, the default value of 20% is assumed.

The following command recycles DFSMShsm ML2 tapes with 10% or less valid data:

```
HSEND RECYCLE ML2 PERCENTVALID(10) EXECUTE
```

We recommend that you recycle your tapes regularly to avoid running out of scratch or DFSMShsm-owned tapes. You can choose a window of less tape activity at your installation to run the **RECYCLE** command.

You can also limit the number of tapes that are recycled by a RECYCLE process. With the **LIMIT** parameter, you can specify RECYCLE to quiesce after the net specified number of tapes are returned to scratch status. The following command causes RECYCLE processing to quiesce after a net of 40 tapes is returned to scratch status. The net is determined by tapes that are read in for input minus tapes that are written to for output.

This example releases up to 40 DFSMShsm ML2 tapes with less than 25% usage:

```
HSEND RECYCLE ML2 EXECUTE PERCENTVALID(25) LIMIT(40)
```

Note: By using our example, during the RECYCLE process, you can empty 44 ML2 tapes, but mount four scratch tapes for output, which results in the net 40 tapes. So, RECYCLE processing will quiesce after it processes a total of 44 tapes. The **LIMIT** parameter is subject to the number of tapes that are allowed by the **PERCENTVALID** parameter.

In certain cases, a single tape does not have enough space to accommodate all data sets that are being migrated, or backed up. In these cases, DFSMShsm selects a second tape to use for migration or backup, and connects it to the first tape, creating a connected set.

Although you can create connected sets as large as 255 tapes, you can recycle only connected sets up to 40 tapes long. In these cases, you need to break the connected set into pieces that are smaller than 40 tapes, and then proceed with RECYCLE. For more information about how to proceed, see Chapter 17, “Hints, tips, and suggested practices” on page 439.

Also, when you recycle connected sets, DFSMShsm checks the first tape in the connected set, and compares it with the percent value that is specified either in the **RECYCLE** command, or in the ARCCMDxx member. If the first tape does not match the requirements, the connected set is not processed, regardless of the overall percent valid of the connected set.

To bypass this control, you can use the **CHECKFIRST(N)** parameter to make DFSMShsm check all tapes in the connected set and calculate the overall percent valid value before it compares it with the recycle requirements. This approach causes more tapes to be recycled because DFSMShsm checks all connected sets to check their eligibility for recycle, but it also causes recycle to run longer for the same reason. The following example recycles DFSMShsm tapes with 10% or less valid data:

```
HSEND RECYCLE ALL PERCENTVALID(10) CHECKFIRST(N) EXECUTE
```

This command causes DFSMShsm to recycle all tapes with 10% or less valid data, and to check all connected sets to calculate whether they are eligible for recycle.

If you use manual tape drives for the DFSMShsm workload, we recommend that you create a list of tapes that are being recycled. Send it to the operators before you start RECYCLE processing so that they can coordinate that the tapes are mounted promptly.

With the **VERIFY** parameter of the **RECYCLE** command, you get two lists of the eligible volumes:

- A *pull list* that operators can use to manually pull the required tapes for the RECYCLE processing. The pull list is made up of small groups of alphabetized volume serial number (volser) lists. The pull groups are listed in the sequence in which recycle processing will most likely request them.
- A *mount list* that aids operators with the mount order of tapes that are being recycled. The mount list is in the anticipated order of the tape mounts that RECYCLE processing will request. The list of tapes is grouped according to the requested category (for example, ML2 and SPILL).

These lists can be helpful in an environment with many tape cartridges and without automated tape libraries.

Both lists are dynamically allocated to either the data set with the prefix that you specified with the **TAPELIST PREFIX(*tapelist_prefix*)** parameter or to a fully qualified data set name with the **RECYCLE** command **TAPELIST FULLDSNAME(*tapelist_dsn*)** parameter. If none of the previous parameters are specified in the **RECYCLE** command, DFSMShsm writes the output to the SYSOUT class that is specified by the **SETSYS** parameter. The data set is deallocated when the command completes.

The following command causes DFSMShsm to look for all eligible ML2 tapes and create the pull and mount lists in a data set with `prefix` as its high-level qualifier (HLQ):

```
HSEND RECYCLE ML2 VERIFY TAPELIST(PREFIX(prefix))
```

If all of your tape volumes are in automated tape libraries, the output from the **TAPELIST** or **VERIFY** parameters are beneficial to you as a reporting tool.

In addition to the options that are presented so far, you can also define a specific tape volume, or tape range, for recycle. This capability gives you an extended granularity to decide what tapes and how many tapes will be recycled when you are migrating to new tape media or technology. You can include as many tape ranges as you need in the **RECYCLE** command.

To recycle all tapes from TP0000 to TP0099 (a specific range), you can use the following command:

```
HSEND RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TA0000:TA0099))) -
PERCENTVALID(100)
```

By selecting PERCENTVALID(100), you ensure that all tapes in the specified range will be selected for recycle.

You can also exclude ranges from processing, which makes it easier to migrate only a subset of tapes in a determined range, for example:

```
HSENDCMD RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TAPE00:TAPE99)) -
EXCLUDE(RANGE(TAPE08:TAPE09))) PERCENTVALID(100)
```

To select multiple ranges to run a recycle, place a space between the ranges that are selected, as shown:

```
HSEND RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TA0000:TA0099 TB0000:TB0099))) -
PERCENTVALID(100)
```

Finally, consider the tape copies of recycled volumes. After a tape is recycled, the tape copy is no longer useful. When you process the **RECYCLE** command, DFSMShsm invalidates alternate or duplex tapes at the same time that it invalidates the tapes from which they were copied.

11.1.8 Moving data sets to new DASD volumes

As your system grows, you face situations where your current available DASD space is insufficient to handle the amount of stored data, and you are required to move the data from the existing DASD technology to larger DASD volumes, or even a new DASD controller.

DFSMShsm can help you with the task of migrating data over to other volumes, and replacing existing migration-level 1 (ML1) volumes with new volumes.

Converting level 0 DASD volumes

DFSMSdss can be used to perform a device conversion of level 0 DASD volumes. Migration tools use mirroring to convert the old DASD volumes to new DASD volumes. You can also use DFSMShsm to move a single data set or an entire level 0 volume to other level 0 volumes.

When you process SMS-managed data sets, you cannot direct the movement to a specific volume. Although you can specify a volume, SMS processing determines the actual volume to which the data set is moved. The process for moving data sets first migrates the data sets to ML1 volumes and then recalls them to level 0 volumes. At the beginning of processing for **MIGRATE VOLUME** commands, DFSMShsm obtains a list of the active management classes in the system. As DFSMShsm processes each data set, it checks the value that is specified for the data set for the COMMAND OR AUTO MIGRATE attribute in the management class. If the value of the attribute is BOTH, DFSMShsm processes the data set.

In recalling each data set, DFSMShsm invokes the automatic class selection (ACS) services of DFSMSdfp. If SMS is active, the ACS routine might return a storage class to DFSMShsm and, optionally, a management class. If the ACS routine returns a storage class, DFSMShsm passes the data set name, with its storage class name and management class name (if present), to DFSMSdss, which interfaces with DFSMSdfp to select a volume to receive the data set. The following command moves all DASD data sets that are on the level 0 volume to other level 0 volumes:

```
HSENDCMD MIGRATE VOLUME(volser) DAYS(0) CONVERT
```

> **Important:** Your SMS polices for the data sets that are being converted can be altered during the migrate and recall process, depending on how you control your ACS routines.

To prevent SMS from selecting the source volume as the target volume for the recall, change the status attribute for the volume in the storage group. The suitable status attributes are DISABLENEW and QUIESCENEW.

Any data set that needs to be expired is expired without being migrated, during the conversion process.

Converting level 1 DASD volumes

With the **FREEVOL** command, you can empty an ML1 volume in preparation for new equipment or to replace old or partially damaged DASD volumes. The **FREEVOL** command moves migration copies of SMS-managed data sets from an ML1 volume based on each data set's management class attribute value.

You can use the following command to empty a specific volume from your ML1 pool:

```
HSEND FREEVOL MIGRATIONVOLUME(volser) AGE(0)
```

The **AGE(0)** parameter, when it is entered for an ML1 volume, causes DFSMShsm to move the migration copies of all SMS-managed data sets, except those copies that need backup, from the volume and place them on other ML1 or ML2 volumes, depending on the data set's management class. If the management class value is specified for the LEVEL-1-DAYS-NONUSAGE and the age is met, the data set migrates to an ML2 volume. If the management class value is specified for LEVEL-1-DAYS-NONUSAGE and the data set age does not meet the criterion or if the data set has an attribute of NOLIMIT, the data set migrates to another ML1 volume. You can restrict which ML1 DASD volumes receive data sets by using the **DRAIN** parameter of the **ADDVOL** command. **DRAIN** removes a volume from selection candidacy.

The data sets, which need backup (that were migrated but are waiting for automatic backup), and the backup versions, which were created by the **BACKDS** command, are not moved off the volume. When you plan to remove an ML1 volume, you must first run AUTOBACKUP on the primary host, and then execute the **FREEVOL AGE(0)** command.

After you empty the ML1 volume, delete the volume by using the following command:

```
HSEND DELVOL volser MIGRATION
```

11.1.9 Auditing DFSMShsm

The **AUDIT** command detects, reports, diagnoses, and often provides repairs for discrepancies between CDSs, catalogs, and DFSMShsm-owned volumes.

To ensure data integrity, DFSMShsm uses numerous data set records to track individual data sets. These records are contained in the following places:

- Master catalog, which is a list of data sets for the entire system
- User catalog, which is a list of data sets that are accessible from that catalog
- Journal, which keeps a running record of backup and migration transactions
- Small-data-set packing (SDSP) data sets on migration volumes
- Migration control data set (MCDS), which is an inventory of migrated data sets and migration volumes
- Backup control data set (BCDS), which is an inventory of backed-up data sets and volumes, dumped volumes, and backed-up aggregates
- Offline control data set (OCDS), which contains a tape table of contents (TTOC) inventory of migration and backup tape volumes

In normal operation, these records stay synchronized. However, because of data errors, hardware failures, or human errors, these records can become unsynchronized. The **AUDIT** command allows the system to cross-check the various records about data sets and DFSMShsm resources. AUDIT can list errors and propose diagnostic actions or, at your option, complete most repairs itself.

Consider the use of the **AUDIT** command for the following reasons:

- After any CDS restore (highly recommended)
- After an ARC184I message (error when reading or writing DFSMShsm CDS records)
- Errors on the RECALL or DELETE of migrated data sets
- Errors on BDELETE or RECOVER of backup data sets
- DFSMShsm tape selection problems
- RACF authorization failures
- Power or hardware failure
- Periodic checks

You can use the **AUDIT** command to cross-check the following sources of control information:

- MCDS or individual migration data set records
- BCDS or individual backup data set records or ABARS records
- OCDS or individual DFSMShsm-owned tapes
- DFSMShsm-owned DASD volumes
- Migration-volume records
- Backup-volume records
- Recoverable-volume records (from dump or incremental backup)
- Contents of SDSP data sets

Use the **AUDIT** command at times of low system activity because certain audit processes can run for a long time. However, the **AUDIT** process can be used almost any time.

The following command shows how to instruct DFSMShsm to audit the BCDS and fix any errors:

```
HSEND AUDIT BCDS FIX
```

See *z/OS DFSMS Storage Administration Reference (for DFSMShsm, DFSMSdss, DFSMSdfp)*, SC26-7402, which provides examples of coding the command and guidance about interpreting its output.

11.2 Journal as a large format data set

The journal data set provides DFSMShsm with a record of each critical change to a CDS from any host since the last time that CDS was successfully backed up. DFSMShsm recovers control data sets (CDSs) by merging the journal with a backed-up version of the CDS. CDSs cannot be recovered to the point of failure without the journal. Use of the journal is highly recommended.

As your system and DFSMShsm-managed data grows, the number of backups, dumps, ABARS, migrations, recalls, and recovers are expected to increase proportionally. This increase can affect your DFSMShsm processing if your journal data set is defined with a small allocation and becomes full before CDS backups take place.

The journal is a sequential data set that can reside only in a single volume, and it must not have secondary allocation. In old z/OS releases, the journal data set was limited to 65,535 tracks allocation, which was the maximum number of tracks that are allocated by a sequential data set in a single volume. Starting with V1R7, you can increase this limit by setting the journal as a large format data set.

The large format sequential data set allows the user to set up a journal with more than 65,535 tracks, relieving the user from backing up the journal and CDSs many times a day in systems with high DFSMShsm processing.

Before you migrate a journal to large format data set, consider the following information:

- In a multiple DFSMShsm host environment, if hosts share a single set of CDSs, they must also share a single journal. All DFSMShsm recovery procedures are based on a single journal to merge with a backed-up version of a CDS.
- Before you migrate the journal to large format, all DFSMShsm hosts that share the journal must be at z/OS V1R7 or later.
- You must allocate the journal as a single volume, single extent data set that is contiguous and non-striped.

Migrating the journal to a large format data set

Using a larger journal data set can allow more DFSMShsm activity to take place between journal backups and helps to avoid journal full conditions. For more information about large format data sets, see *DFSMS Using Data Sets,* SC26-7410.

If you decide to migrate your current journal to a large format data set, follow these steps to avoid corrupting your CDSs during the conversion process.

As a first step, you can run an IEFBR14 program to preallocate the new journal data set, as shown in Example 11-15. The target volume must have the contiguous space that you specified and no secondary allocation is allowed.

Example 11-15 Define the journal data set

```
//MHLRES71 JOB (999,POK),'HSM JOURNAL',MSGCLASS=Y,REGION=0M,TIME=10,
// NOTIFY=&SYSUID,CLASS=A
//***********************************************************//
//***********************************************************//
//** BEFORE SUBMITTING THE JOB, PLEASE UPDATE THE FOLLOWING **//
//**                   PARAMETERS                           **//
//**                                                        **//
//** yourVOLS - The VOLSER you are allocation new journal   **//
//** yourJRNL - The new journal name                        **//
//**                                                        **//
//** You may also want to change space values to fit your   **//
//**                  environment needs                     **//
//**                                                        **//
//***********************************************************//
//***********************************************************//
//ALLOCATE EXEC PGM=IEFBR14
//JOURNAL DD DSN=yourJRNL,DISP=(,CATLG),UNIT=3390,
//   VOL=SER=yourVOLS,SPACE=(CYL,(400),,CONTIG),DSNTYPE=LARGE
//SYSTSIN    DD DUMMY
//SYSTSPRT DD SYSOUT=*
```

After you preallocate the new journal, bring down all but one DFSMShsm task. Ensure that you give DFSMShsm enough time to finish any currently running process. It might take a few minutes to bring down DFSMShsm, depending on the process that is running.

Next, hold all DFSMShsm tasks to ensure that no new tasks come in during CDS backups. After you put all functions on hold, and no DFSMShsm functions are running, you can issue the command that is shown in Example 11-16 to back up CDSs and clean up the journal. Do not forget to also hold the common queue if it is implemented in your environment.

Example 11-16 Backing up CDSs

```
HSEND HOLD ALL
HSEND HOLD COMMONQUEUE
HSEND BACKVOL CDS
```

> **Note:** It might take time for all current tasks to finish after you issue the **`HOLD`** command. We recommend that you issue an **`HSEND Q ACT`** command to ensure that no tasks are running before you issue the **`HSEND BACKVOL CDS`** command.

Example 11-17 on page 338 shows the message that you receive when your **`BACKVOL CDS`** command is complete.

Example 11-17 Output from Example 11-16 on page 337

```
ARC0740I CDS BACKUP STARTING AT 12:23:50 ON 2012/08/23, SYSTEM SC64, TO DASD
ARC0740I (CONT.) IN PARALLEL MODE, DATAMOVER=DSS
ARC0742I BACKUP FOR MCDS STARTING AT 12:23:50 ON 2012/08/23, BACKUP COPY
ARC0742I (CONT.) TECHNIQUE IS STANDARD
ARC0742I BACKUP FOR BCDS STARTING AT 12:23:50 ON 2012/08/23, BACKUP COPY
ARC0742I (CONT.) TECHNIQUE IS STANDARD
ARC0742I BACKUP FOR OCDS STARTING AT 12:23:50 ON 2012/08/23, BACKUP COPY
ARC0742I (CONT.) TECHNIQUE IS STANDARD
ARC0750I BACKUP FOR JRNL STARTING AT 12:23:50, ON 2012/08/23, BACKUP TECHNIQUE
ARC0750I (CONT.) IS QUIESCED(B)
ARC0743I JRNL SUCCESSFULLY BACKED UP TO HSM.JRNL.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:50, DATE=2012/08/23
ARC1007I COMMAND REQUEST 00000612 SENT TO DFSMSHSM
ARC0743I MCDS SUCCESSFULLY BACKED UP TO HSM.MCDS.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:51, DATE=2012/08/23
ARC0743I OCDS SUCCESSFULLY BACKED UP TO HSM.OCDS.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:51, DATE=2012/08/23
ARC0743I BCDS SUCCESSFULLY BACKED UP TO HSM.BCDS.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:51, DATE=2012/08/23
ARC0748I LAST SUCCESSFUL CDS BACKUP-SET QUALIFIER IS D0000029
ARC0741I CDS BACKUP ENDING AT 12:23:51 ON 2012/08/23, STATUS=SUCCESSFUL
```

After the backups finish, you can shut down the last DFSMShsm host.

Rename your current journal data set to something else, and rename the new journal you created to the DFSMShsm journal name. Then, restart all of the DFSMShsm hosts.

You can delete your old journal data set after you checked that all DFSMShsm hosts are working as expected.

11.3 Maintaining SDSP data sets: Reorganization

In most environments, a major part of the data sets that are created on DASD every day are small data sets, occupying a few tracks or even one track in a level 0 volume. Many of these data sets store only a few records and the rest of the tracks that are allocated are free and wasted space.

DFSMShsm helps you to reduce the wasted space with these small data sets by giving you a simple way to migrate them to ML1 volumes so that they allocate only the required space, and also take advantage of data compression that is performed during this migration.

Small user data sets reside on separate tracks of level-0 user volumes. DFSMShsm can migrate the small user data sets (as records) and store them as records in a VSAM key-sequenced SDSP data set on a level-1 migration volume. DASD space is reduced because multiple data sets then share the same tracks on level-1 migration volumes.

It is important to plan the number of SDSP data sets in relation to the number of concurrent migration tasks and the amount of processing by functions with a higher usage priority for the SDSP data sets, such as recalls or ABARS backups.

The SDSP data set is a VSAM key-sequenced data set and as the data sets are migrated to, and recalled from SDSP, VSAM will need to be reorganized to erase old, and unnecessary records, and to reorganize its index and data components to improve performance.

You can significantly reduce the need to reorganize SDSP data sets by enabling the control area (CA) reclaim function for them. For more information, see the topic about reclaiming CA space in *DFSMS Using Data Sets,* SC26-7410.

We recommend that you plan your SDSP reorganization in a window apart from the space management window, so no migration tasks will request access to the SDSP data set.

Before you start your reorganization process, verify that no recalls are actively accessing your SDSP. Next, hold recalls from all DFSMShsm products with access to the SDSP. Also, hold the common request queue (CRQ) if it is implemented in your environment. Example 11-18 shows you both of the commands to hold recall and the common queue.

Example 11-18 Holding recall and the common queue

```
HSEND HOLD RECALL
HSEND HOLD COMMONQUEUE
```

You can also use the **DRAIN** parameter to drain the volume where the SDSP data set is allocated, so no migration tasks will select this volume for migration, preserving the available space on the volume. To drain the volume, you can issue the following command. This command adds the migration volume to DFSMShsm:

```
HSEND ADDVOL volser MIGRATION(MIGRATIONLEVEL1 SDSP DRAIN) unit(3390)
```

In sequence, we will export (EXPORT) the current SDSP as part of our reorganization process. Example 11-19 shows you a sample JCL to export the SDSP data set.

Example 11-19 Exporting SDSP data set

```
//MHLRES71 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*****************************************************************//
//*****************************************************************//
//**    BEFORE SUBMITTING THE JOB, PLEASE CHANGE THE FOLLOWING     **//
//**                         PARAMETERS                            **//
//**                                                               **//
//**  yourSDSP - Your SDSP data set name                           **//
//**  yourEXPT - Your temporary EXPORT copy                        **//
//**  yourUNIT - Your Unit name                                    **//
//**                                                               **//
//**   You may also need to change space information on ALLOCATE   **//
//**          STEP to accommodate your SDSP entries                **//
//**                                                               **//
//*****************************************************************//
//*****************************************************************//
//ALLOCATE EXEC PGM=IEFBR14
//EXPORTS DD DSN=yourEXPT,DISP=(,CATLG),
// UNIT=yourUNIT,SPACE=(CYL,(20,2))
//IDCAMS EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EXAMINE NAME(yourSDSP) INDEXTEST
IF LASTCC = 0 THEN -
        EXPORT yourSDSP ODS(yourEXPT) TEMPORARY
/*
```

> **Note:** Your space requirements might vary depending on the size of your SDSP and the amount of valid data.

Next, we run another job to reallocate the SDSP data set on the ML1 volume, and then import the data from the export copy. See Example 11-20.

Example 11-20 Defining the SDSP VSAM data set and import

```
//MHLRES71 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*******************************************************************//
//*******************************************************************//
//**    BEFORE SUBMITTING THE JOB, PLEASE CHANGE THE FOLLOWING     **//
//**                         PARAMETERS                            **//
//**                                                               **//
//**  yourVOLS - The ML1 volser the SDSP will be allocated         **//
//**  yourSDSP - Your SDSP data set name                           **//
//**  yourEXPT - Your EXPORT data set name                         **//
//**                                                               **//
//*******************************************************************//
//*******************************************************************//
//STEP1 EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=*
//SDSP1 DD UNIT=SYSDA,VOL=SER=yourVOLS,DISP=SHR
//SYSIN DD *
 DEFINE CLUSTER (NAME(yourSDSP) VOLUMES(yourVOLS) -
        CYLINDERS(5 0) FILE(SDSP1) -
        RECORDSIZE(2093 2093) FREESPACE(0 0) -
        INDEXED KEYS(45 0) -
        SPEED BUFFERSPACE(530432) -
        UNIQUE NOWRITECHECK) -
        DATA -
        (CONTROLINTERVALSIZE(26624)) -
        INDEX -
        (CONTROLINTERVALSIZE(1024))
 IF MAXCC=0 THEN -
  IMPORT IDS(yourEXPT) ODS(yourSDSP) IEMPTY
```

Note: Your space requirements might vary depending on the size of your SDSP and the amount of valid data.

The SDSP reorganization is complete. Now, we need to release all of the functions that we held at the beginning of the reorganization, and issue the **ADDVOL** command with **NODRAIN**, so DFSMShsm will set this volume as available for migration. See Example 11-21.

Example 11-21 Releasing recall and adding ML1 volume to DFSMShsm

```
HSEND RELEASE RECALL
HSEND RELEASE COMMON QUEUE
HSEND ADDVOL volser MIGRATION(MIGRATIONLEVEL1 SDSP NODRAIN) UNIT(3390)
```

For more information about the parameters to use in **DEFINE CLUSTER**, see the detailed information about VSAM data sets in *VSAM Demystified*, SG24-6105.

12

Control data set considerations

In this chapter, we describe the care and maintenance of your DFSMShsm control data sets (CDSs).

12.1 DFSMShsm journal and control data sets

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

We briefly explain how CDSs work, CDS reorganization, backup and recovery techniques and considerations, multicluster CDSs, record-level sharing (RLS) for CDSs, and CDS enhancements.

For information about how to allocate CDSs, see Chapter 3, “Getting started” on page 29.

12.1.1 How control data sets work

Most DFSMShsm tasks that involve user data sets, such as backing up, migrating, and dump, must be tracked so DFSMShsm can recover the data set if the user accidentally deletes it, recall a migrated data set, or restore a dump in a disaster recovery.

DFSMShsm tracks its activities through the backup control data set (BCDS), migration control data set (MCDS), offline control data set (OCDS), and journal data set.

Each CDS has its own function in the DFSMShsm structure. Although BCDS, MCDS, or OCDS are not required by DFSMShsm, at least one of them must be allocated. Not having one of these CDSs affects functions that are provided by DFSMShsm. Although the journal data set is not required by DFSMShsm to run, we strongly recommend that you allocate the journal for recovery.

The OCDS stores all information about offline media that is used by DFSMShsm. It works with MCDS and BCDS to supply DFSMShsm with all of the information necessary to perform all tape-related tasks. If you do not allocate the OCDS, or if the OCDS becomes unavailable, you will still be able to perform all DFSMShsm function, but all requests must use DASD.

The BCDS is responsible for storing backup information for every data set that is managed and backed up by DFSMShsm on either on DASD or tape. It also records information about aggregate backup and recovery support (ABARS) data sets, dumps, and fast recovery backups. Every time a recover is issued for any of these functions, DFSMShsm uses BCDS information to process the request. If BCDS is not allocated, DFSMShsm is not able to perform ABARS, backups, dumps, or fast recovery backup-related tasks. You will not be able to recover the data from any existing backups that are controlled by DFSMShsm if the BCDS is lost.

The MCDS, or migration control data set, records all information regarding DFSMShsm migration to DASD and tape. DFSMShsm uses MCDS information to migrate and recall data sets from DASD volumes and uses MCDS and OCDS information to migrate and recall data sets from tape. If you decide not to use MCDS, or if it becomes unavailable, DFSMShsm will not be able to migrate or recall data sets from DASD or tape.

The journal records information about all DFSMShsm activities. Every time DFSMShsm performs an activity, such as the activities described here, it updates both the CDSs and the journal data set, so, if any CDSs are lost, you can recover DFSMShsm to the current point by recovering the last CDS backup, and applying the journal updates. If you lose your journal data set, you will not be able to perform CDS recovery.

DFSMShsm uses all of the described data sets to store and retrieve the information that it needs to perform its activities.

All of the CDSs are VSAM key-sequenced data sets (KSDSs), and occasionally require reorganization to remove invalid data, and to extend space allocation.

The journal data set is a sequential data set, and it is nulled every time a successful CDS backup runs. You might want automation set up to automatically back up CDSs every time that a journal reaches a determined percent utilization. For more information about backing up CDSs, see the 12.3, "CDS backup procedures" on page 367.

12.1.2 CDS reorganization

The DFSMShsm CDSs, as any other VSAM files, require occasional reorganization. Detection of a need to reorganize can be based on the information that is received automatically through the ARC0909I message. This message is issued when the occupancy threshold that is requested in the **`SETSYS MONITOR`** command is exceeded. You can obtain the current occupancy status by using the **`QUERY CONTROLDATASETS`** command. Status is reported in the ARC0148I message. See Example 12-1.

Example 12-1 Output from HSEND Q CDS command

```
ARC0101I QUERY CONTROLDATASETS COMMAND STARTING ON HOST=2
ARC0947I CDS SERIALIZATION TECHNIQUE IS RLS
ARC0148I MCDS TOTAL SPACE=72000 K-BYTES, CURRENTLY ABOUT 27% FULL, WARNING
ARC0148I (CONT.) THRESHOLD=80%, TOTAL FREESPACE=94%, EA=YES, CANDIDATE
ARC0148I (CONT.) VOLUMES=4
ARC0948I MCDS INDEX TOTAL SPACE=210 K-BYTES, CURRENTLY ABOUT 26% FULL, WARNING
ARC0948I (CONT.) THRESHOLD=80%, CANDIDATE VOLUMES=4
ARC0148I BCDS TOTAL SPACE=72000 K-BYTES, CURRENTLY ABOUT 48% FULL, WARNING
ARC0148I (CONT.) THRESHOLD=80%, TOTAL FREESPACE=96%, EA=YES, CANDIDATE
ARC0148I (CONT.) VOLUMES=4
ARC0948I BCDS INDEX TOTAL SPACE=210 K-BYTES, CURRENTLY ABOUT 46% FULL, WARNING
ARC0948I (CONT.) THRESHOLD=80%, CANDIDATE VOLUMES=4
ARC0148I OCDS TOTAL SPACE=72000 K-BYTES, CURRENTLY ABOUT 34% FULL, WARNING
ARC0148I (CONT.) THRESHOLD=80%, TOTAL FREESPACE=96%, EA=YES, CANDIDATE
ARC0148I (CONT.) VOLUMES=4
ARC0948I OCDS INDEX TOTAL SPACE=210 K-BYTES, CURRENTLY ABOUT 33% FULL, WARNING
ARC0948I (CONT.) THRESHOLD=80%, CANDIDATE VOLUMES=4
ARC0148I JOURNAL TOTAL SPACE=3875053 K-BYTES, CURRENTLY ABOUT 000% FULL,
ARC0148I (CONT.) WARNING THRESHOLD=080%, TOTAL FREESPACE=100%, EA=NO, CANDIDATE
ARC0148I (CONT.) VOLUMES=0
ARC0101I QUERY CONTROLDATASETS COMMAND COMPLETED ON HOST=2
```

You do not have to reorganize the CDSs for performance reasons. The occurrence of control interval (CI) or control area (CA) splits in the CDSs does not affect DFSMShsm performance.

Figure 12-1 on page 344 shows the anatomy of a CDS. For the MCDS, mid-section records are generated for each migrated data set or VSAM component. For the BCDS, mid-section records are generated for each backup copy.

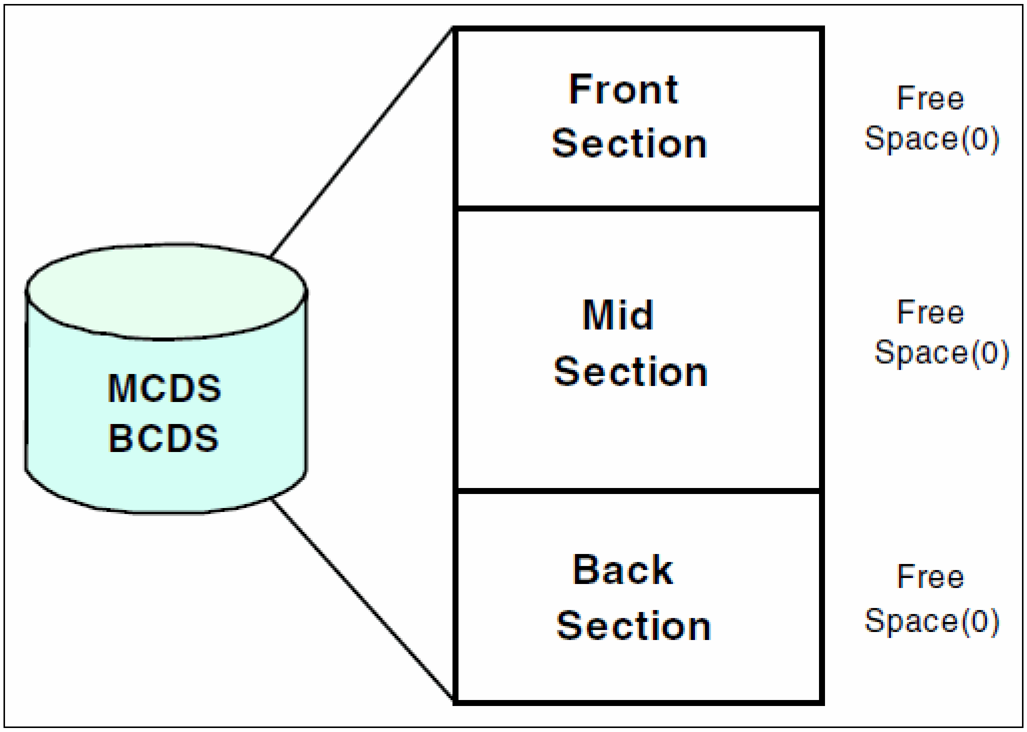

Figure 12-1 Control data set anatomy

Note: Because DFSMShsm has its own logic when it defines keys, it is not possible to guarantee that free space that is left in CIs and CAs will ever be used.

Consider the following information before you reorganize your CDSs. The following list is a summary of considerations for reorganizing the CDSs:

- Only reorganize them when you have to increase their size, or when you are moving the CDSs to another DASD device.

  Because DFSMShsm does not perform many sequential reads for data, its performance is not directly affected by CI and CA splits, and you can increase DFSMShsm availability time with fewer reorganizations.

- DFSMShsm must be shut down in all hosts that have access to the CDSs, and no other job in any host must be using them during this operation.

  Shutting down all DFSMShsm hosts will ensure that no updates are performed on CDSs or the journal during reorganization. An update during the reorganization process can lead to a mismatch between CDSs and the loss of user data.

- Allocate the CDS that you are going to reorganize with DISP=OLD.

  Using DISP=OLD in your JCL will request exclusive access to CDSs, so no DFSMShsm or jobs will be able to allocate these data sets.

- Reorganize the CDSs with FREESPACE(0) and let DFSMShsm split the mid-section intervals. Performance will be degraded for about two or three weeks during the CI and CA split process.

  This approach is recommended because allocating FREESPACE might result in wasted non-used space in CDSs.

When you decide to reorganize DFSMShsm CDSs, follow a few procedures to ensure that your reorganization is successful, or ensure that you have the necessary information to proceed with a backout if the reorganization fails.

First, bring down all DFSMShsm hosts that access the CDSs, and leave only one host up. It is necessary to leave one host up so you can perform CDS backups before the reorganization.

In sequence, you can create a backup of all CDSs and the journal, so if you must back out your reorganization, you have a current backup to recover from, reducing recovery time.

To create this backup, you can hold all DFSMShsm tasks to ensure that no new tasks come in during CDS reorganization. After you put all functions on hold, and you confirm that no DFSMShsm functions are running, you can issue the command to back up CDSs and clean up the journal. If any tasks are running on any DFSMShsm hosts, wait for them to complete before you back up the CDSs. Do not forget to also hold the common queue if it is implemented in your environment. See Example 12-2.

Example 12-2 Holding all DFSMShsm functions and performing CDS backup

```
HSEND HOLD ALL
HSEND HOLD COMMONQUEUE
HSEND BACKVOL CDS
```

Example 12-3 shows the message that you will receive when your **BACKVOL CDS** command is complete.

Example 12-3 Output from HSEND BACKVOL CDS command

```
ARC0740I CDS BACKUP STARTING AT 12:23:50 ON 2012/08/23, SYSTEM SC64, TO DASD
ARC0740I (CONT.) IN PARALLEL MODE, DATAMOVER=DSS
ARC0742I BACKUP FOR MCDS STARTING AT 12:23:50 ON 2012/08/23, BACKUP COPY
ARC0742I (CONT.) TECHNIQUE IS STANDARD
ARC0742I BACKUP FOR BCDS STARTING AT 12:23:50 ON 2012/08/23, BACKUP COPY
ARC0742I (CONT.) TECHNIQUE IS STANDARD
ARC0742I BACKUP FOR OCDS STARTING AT 12:23:50 ON 2012/08/23, BACKUP COPY
ARC0742I (CONT.) TECHNIQUE IS STANDARD
ARC0750I BACKUP FOR JRNL STARTING AT 12:23:50, ON 2012/08/23, BACKUP TECHNIQUE
ARC0750I (CONT.) IS QUIESCED(B)
ARC0743I JRNL SUCCESSFULLY BACKED UP TO HSM.JRNL.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:50, DATE=2012/08/23
ARC1007I COMMAND REQUEST 00000612 SENT TO DFSMSHSM
ARC0743I MCDS SUCCESSFULLY BACKED UP TO HSM.MCDS.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:51, DATE=2012/08/23
ARC0743I OCDS SUCCESSFULLY BACKED UP TO HSM.OCDS.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:51, DATE=2012/08/23
ARC0743I BCDS SUCCESSFULLY BACKED UP TO HSM.BCDS.BACKUP.D0000025, ON VOLUME(S)
ARC0743I (CONT.) SBXHS5, TIME=12:23:51, DATE=2012/08/23
ARC0748I LAST SUCCESSFUL CDS BACKUP-SET QUALIFIER IS D0000029
ARC0741I CDS BACKUP ENDING AT 12:23:51 ON 2012/08/23, STATUS=SUCCESSFUL
```

After backups finish, you can shut down the last DFSMShsm host.

With all DFSMShsm hosts down, you can run your job to examine CDSs, perform export, allocate new CDSs, and import data. Each step is described in a different example, so you can skip or replace the content that is provided, as required.

The first job uses IEFBR14 to allocate the data sets to use during the export job. See Example 12-4.

Example 12-4 Creating export data sets

```
//MHLRES71 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*****************************************************************//
//*****************************************************************//
//**                                                             **//
//**   Before submitting this JCL, update the following fields   **//
//**                                                             **//
//** ?UID - The HLQ used during export processing                **//
//** ?UNIT - The Unit name used for export data sets             **//
//**                                                             **//
//** You may also want to review space allocation, to make sure  **//
//**             you have enough space to export                 **//
//**                                                             **//
//*****************************************************************//
//*****************************************************************//
//ALLOCATE EXEC PGM=IEFBR14
//EXPMCDS DD DSN=?UID.EXPORT.MCDS,DISP=(,CATLG),
// UNIT=?UNIT,SPACE=(CYL,(20,2))
//EXPBCDS DD DSN=?UID.EXPORT.BCDS,DISP=(,CATLG),
// UNIT=?UNIT,SPACE=(CYL,(20,2))
//EXPOCDS DD DSN=?UID.EXPORT.OCDS,DISP=(,CATLG),
// UNIT=?UNIT,SPACE=(CYL,(20,2))
```

Next, the job will examine all CDSs to ensure that no discrepancies exist between INDEX and DATA components. See Example 12-5.

Example 12-5 Examining CDSs

```
//MHLRES71 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//*****************************************************************//
//*****************************************************************//
//**                                                             **//
//**   Before submitting this JCL, update the following fields   **//
//**                                                             **//
//** ?UID.MCDS - Change to your MCDS name                        **//
//** ?UID.BCDS - Change to your BCDS name                        **//
//** ?UID.OCDS - Change to your OCDS name                        **//
//**                                                             **//
//*****************************************************************//
//*****************************************************************//
//IDCAMS EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 LISTCAT ENT(?UID.MCDS) ALL
 LISTCAT ENT(?UID.BCDS) ALL
 LISTCAT ENT(?UID.OCDS) ALL
 EXAMINE NAME(?UID.MCDS) INDEXTEST
 EXAMINE NAME(?UID.BCDS) INDEXTEST
 EXAMINE NAME(?UID.OCDS) INDEXTEST
```

Note: If you receive a return code other than zero, you can check the job log for error messages. For more information about IDCAMS errors messages, see *z/OS MVS System Messages Vol 6 (GOS-IEA),* SA22-7636.

If all examinations ran acceptably, you can proceed with the export job. The export process will read your CDS and copy all of the valid data to a sequential data set. See Example 12-6.

Example 12-6 Exporting CDSs

```
//EDGUXAOL JOB ,RMM,MSGLEVEL=1,MSGCLASS=H,REGION=0M,NOTIFY=&SYSUID
//*****************************************************************//
//*****************************************************************//
//**                                                             **//
//**   Before submitting this JCL, update the following fields   **//
//**                                                             **//
//** ?UID.MCDS - Change to your MCDS name                        **//
//** ?UID.BCDS - Change to your BCDS name                        **//
//** ?UID.OCDS - Change to your OCDS name                        **//
//** ?UID.EXPORT.MCDS - Change to your EXPORT MCDS data set      **//
//** ?UID.EXPORT.bCDS - Change to your EXPORT bCDS data set      **//
//** ?UID.EXPORT.oCDS - Change to your EXPORT OCDS data set      **//
//**                                                             **//
//*****************************************************************//
//*****************************************************************//
//IDCAMS EXEC PGM=IDCAMS,REGION=512K
//MCDS DD DSN=?UID.MCDS,DISP=OLD
//BCDS DD DSN=?UID.BCDS,DISP=OLD
//OCDS DD DSN=?UID.OCDS,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  EXPORT ?UID.MCDS ODS(?UID.EXPORT.MCDS) TEMPORARY
  EXPORT ?UID.MCDS ODS(?UID.EXPORT.MCDS) TEMPORARY
  EXPORT ?UID.MCDS ODS(?UID.EXPORT.MCDS) TEMPORARY
```

After the export job runs, you can submit a new job to delete old CDSs and allocate new CDSs. Figure 12-7 on page 371 is a sample JCL that you can use to delete and allocate your CDSs. This example is a sample job, and you might need to update parameters to meet your requirements. You can execute a **LISTC** against your current CDSs to check their allocation specifications. Before you submit this job, confirm that all export jobs ran with RC=0 and all export data sets are cataloged.

Example 12-7 Deleting old CDSs and allocating new ones

```
//MHLRES71 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//****************************************************************//
//****************************************************************//
//**                                                            **//
//** In order to submit this job, you have to update several    **//
//** parameters, including CDSs names, volsers, units and space **//
//**                    among others.                           **//
//**                                                            **//
//****************************************************************//
//****************************************************************//
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
```

```
//HSMMCDS  DD UNIT=3390,VOL=SER=HG6622,DISP=SHR
//HSMBCDS  DD UNIT=3390,VOL=SER=HG6622,DISP=SHR
//HSMOCDS  DD UNIT=3390,VOL=SER=HG6622,DISP=SHR
//SYSIN    DD *
   DEFINE CLUSTER (NAME(?UID.MCDS) VOLUMES(HG6622) -
        CYLINDERS(10) FILE(HSMMCDS) -
        RECORDSIZE(435 2040) FREESPACE(0 0) -
        INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
        SPEED BUFFERSPACE(530432) -
        UNIQUE NOWRITECHECK LOG(NONE)) -
        DATA(NAME(?UID.MCDS.DATA) -
        CONTROLINTERVALSIZE(12288)) -
        INDEX(NAME(?UID.MCDS.INDEX) -
        CONTROLINTERVALSIZE(4096))
   DEFINE CLUSTER (NAME(?UID.BCDS) VOLUMES(HG6622) -
        CYLINDERS(10) FILE(HSMBCDS) -
        RECORDSIZE(6544 6544) FREESPACE(0 0) -
        INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
        SPEED BUFFERSPACE(530432) -
        UNIQUE NOWRITECHECK) -
        DATA(NAME(?UID.BCDS.DATA) -
        CONTROLINTERVALSIZE(12288)) -
        INDEX(NAME(?UID.BCDS.INDEX) -
        CONTROLINTERVALSIZE(4096))
   DEFINE CLUSTER (NAME(?UID.OCDS.OCDS) VOLUMES(HG6622) -
        CYLINDERS(10) FILE(HSMOCDS) -
        RECORDSIZE(1800 2040) FREESPACE(0 0) -
        INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
        SPEED BUFFERSPACE(530432) -
        UNIQUE NOWRITECHECK) -
        DATA(NAME(?UID.OCDS.DATA) -
        CONTROLINTERVALSIZE(12288)) -
        INDEX(NAME(?UID.OCDS.INDEX) -
        CONTROLINTERVALSIZE(4096))
```

The last step of your reorganization is importing the data back to the new CDSs. Execute the IDCAMS program with the **IMPORT** command. See Example 12-8.

Example 12-8 Importing data back to new CDSs

```
//MHLRES77 JOB ,RMM,MSGLEVEL=1,MSGCLASS=H,REGION=0M,NOTIFY=&SYSUID
//IDCAMS EXEC PGM=IDCAMS,REGION=512K
//*****************************************************************//
//*****************************************************************//
//**                                                             **//
//** Before submitting this job, please update the following     **//
//**                        parameters                           **//
//**                                                             **//
//** ?UID.MCDS - Your MCDS data set                              **//
//** ?UID.BCDS - Your BCDS data set                              **//
//** ?UID.OCDS - Your OCDS data set                              **//
//** ?UID.EXPORT.MCDS - Your EXPORT MCDS data set                **//
//** ?UID.EXPORT.BCDS - Your EXPORT BCDS data set                **//
//** ?UID.EXPORT.OCDS - Your EXPORT OCDS data set                **//
//**                                                             **//
//*****************************************************************//
```

```
//**************************************************************//
//MCDS DD DSN=?UID.MCDS,DISP=OLD
//**                                                          **//
//**************************************************************//
//**************************************************************//
//MCDS DD DSN=?UID.MCDS,DISP=OLD
//BCDS DD DSN=?UID.BCDS,DISP=OLD
//OCDS DD DSN=?UID.OCDS,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        IMPORT IDS(?UID.EXPORT.MCDS) ODS(?UID.MCDS) IEMPTY
        IMPORT IDS(?UID.EXPORT.BCDS) ODS(?UID.BCDS) IEMPTY
        IMPORT IDS(?UID.EXPORT.OCDS) ODS(?UID.OCDS) IEMPTY
```

12.1.3 CDS and journal performance suggestions

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

You can take several simple actions to help you achieve the CDS performance goals. Although most of them are not required, we recommend that you read all of the following actions, and consider implementing them when possible.

First, when allocating the CDSs, think about the type of media and hardware to use. Selecting a cached DASD for CDSs will help you to increase performance when reading and writing data on a CDS. Using DASD fast write for the journal gives you an increased speed when writing data to the journal, which is the likely operation you will perform to this data set.

You can select one cluster for each CDS, or up to four clusters for MCDS and BCDS, and only one cluster for OCDS. Either way, you cannot define a single cluster to extend to more than one volume. Using more than one cluster for a CDS is referred to as *multicluster CDS*.

Do not specify a secondary allocation for a cluster, or you will not be able to use multicluster for the specified CDS. You might receive more than one monitor message when the CDS threshold is reached; deadlocks might occur. DFSMShsm will also display ARC0130I during the start to inform you that secondary allocation was specified for this CDS.

Allocate each CDS and journal on a different volume where possible to minimize the risk of contention between a CDS and other CDSs or data sets on the volume. If not possible, you must not allocate CDSs on the same volume as JES or system data sets.

The data component of one CDS must not be on the same volume as the index component of another CDS unless the index of the first CDS is also on the volume.

For performance reasons, index components of different CDSs must not be on the same volume. You must not allocate the index of a different CDS with the data component of another CDS if its index is not on the same volume.

If CDSs and the journal are storage management subsystem (SMS)-managed, they can be assigned to a storage class with the GUARANTEED SPACE attribute. This attribute will make it easier for you to manage where the data sets are placed, and avoid allocating two indexes on the same volume.

Allocate CDSs on DASD devices that offer the Concurrent Copy feature, and use the **SETSYS CDSVERSIONBACKUP** command to back up the CDSs and journal to minimize the outage during CDS backup.

Ensure that the CDSs, CDS backup copies, journal, and journal backup copies are not allowed to migrate. Migrating CDSs or the journal can make your DFSMShsm stop working, and migrating backup data sets will cause CDS backups to fail.

Use different head disk assemblies (HDAs) for CDSs and the journal; consider the use of volumes that connect to different control units. This way, recovery times will be shorter in a hardware failure because not all of the CDSs will need to be recovered.

Implement and test CDS backup and recovery procedures before you begin to manage user or application data. After you test the CDS backup and recovery procedures, document them.

12.1.4 CDS and journal backup

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

We recommend that you use multiple backup versions that are written to pre-allocated DASD data sets. DFSMShsm will change the suffix of the data set name dynamically and it is similar to that of a generation data set (GDS). For tape backups, it is not necessary to preallocate any data sets.

We recommend at least eight, and preferably 14 backups. Certain clients keep two to four CDS backups on DASD, then they use the ARCCBEXT to trigger a process to copy the backups to tape. They let their tape management system, such as DFSMSrmm, keep the tapes for a designated period, such as 14 days, and then return them to scratch. For more information about how to set up ARCCBEXT exits, see *DFSMS Installation Exits,* SC26-7396.

Our recommendation is that you use DFSMSdss to back up the CDSs from within DFSMShsm. The invocation of DFSMSdss checks the structure of the CDSs as it dumps them and advises you of any errors that might make recovery of the CDSs impossible.

Because the area of CDS and journal backup can be complicated, we describe in the next sections considerations you must consider, and how to set up your environment for successfully backing up your CDSs and journal.

Creating CDS and journal backups

When DFSMShsm starts, it gets environmental and function information from PARMLIB. You are not required to use SYS1.PARMLIB. The PARMLIB pointed to by the HSMPARM DD statement in the startup procedure will use the ARCCMD00 member or an alternate member that you indicate by the **CMD** keyword. The **SETSYS**, **DEFINE**, and **ADDVOL** commands that define how DFSMShsm manages your data are in the PARMLIB member (see PARMLIB member ARCCMD00 in the starter set).

The **SETSYS CDSVERSIONBACKUP** command determines how DFSMShsm backs up your CDSs. You can specify the following factors with the subparameters of the **CDSVERSIONBACKUP** command:

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

By using SMS storage groups and management classes, you can specify that Concurrent Copy is used to back up the CDSs, assuming that they are allocated on volumes that connect to a controller that provides Concurrent Copy.

If your CDSs, journal, or backup data sets are SMS-managed, ensure that they will not back up or migrate. If they are non-SMS-managed, ensure that they are not under DFSMShsm control, so they will not migrate or take backups.

If you use DASD backup data sets, you can use the following example job to preallocate the data sets. See Example 12-9.

Example 12-9 Allocating CDSs and journal backup data sets

```
//MHLRES71 JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
//ALLOCBK EXEC PGM=IEFBR14
//*******************************************************************//
//*******************************************************************//
//*    THIS SAMPLE JOB ALLOCATES AND CATALOGS THE CONTROL DATA SET   *//
//*    BACKUP VERSION DATA SETS ON DASD VOLUMES.                     *//
//*                                                                  *//
//*    ENSURE THAT BACKUP VERSION DATA SETS ARE PLACED ON VOLUMES    *//
//*    THAT ARE DIFFERENT FROM THE VOLUMES THAT THE CONTROL DATA     *//
//*    SETS ARE ON.                                                  *//
//*                                                                  *//
//*  PARAMETER PARAMETER DEFINITION                                  *//
//*                                                                  *//
//*  ?BKUNIT    -  UNIT TYPE OF VOLUME TO CONTAIN THE CDS            *//
//*                BACKUP VERSION.                                   *//
//*  ?BKVOL1    -  VOLUME SERIAL OF VOLUME TO CONTAIN THE FIRST CDS  *//
//*                BACKUP VERSION.                                   *//
//*  ?BKVOL2    -  VOLUME SERIAL OF VOLUME TO CONTAIN THE SECOND CDS *//
//*                BACKUP VERSION.                                   *//
//*  ?BKVOL3    -  VOLUME SERIAL OF VOLUME TO CONTAIN THE THIRD CDS  *//
//*                BACKUP VERSION.                                   *//
//*  ?BKVOL4    -  VOLUME SERIAL OF VOLUME TO CONTAIN THE FOURTH CDS *//
//*                BACKUP VERSION.                                   *//
//*  ?SCBVOL1   -  STORAGE CLASS NAME FOR BACKUP VERSIONS            *//
//*  ?MCDFHSM   -  MANAGEMENT CLASS NAME OF THE HSM CONSTRUCT        *//
//*  ?CDSSIZE   -  NUMBER OF CYLINDERS ALLOCATED TO CDS BACKUP       *//
//*                VERSIONS.                                         *//
//*  ?JNLSIZE   -  NUMBER OF CYLINDERS ALLOCATED TO JOURNAL DATA     *//
//*                SET.                                              *//
//*  ?UID       -  AUTHORIZED USER ID (1 TO 7 CHARS) FOR THE HSM-    *//
//*                STARTED PROCEDURE. THIS WILL BE USED AS THE       *//
//*                HIGH-LEVEL QUALIFIER OF HSM DATA SETS.            *//
//*                                                                  *//
//*******************************************************************//
//*******************************************************************//
```

```
//MCDSV1 DD DSN=?UID.MCDS.BACKUP.V0000001,DISP=(,CATLG),
//   VOL=SER=?BKVOL1,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//MCDSV2 DD DSN=?UID.MCDS.BACKUP.V0000002,DISP=(,CATLG),
//   VOL=SER=?BKVOL2,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//MCDSV3 DD DSN=?UID.MCDS.BACKUP.V0000003,DISP=(,CATLG),
//   VOL=SER=?BKVOL3,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//MCDSV4 DD DSN=?UID.MCDS.BACKUP.V0000004,DISP=(,CATLG),
//   VOL=SER=?BKVOL4,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//BCDSV1 DD DSN=?UID.BCDS.BACKUP.V0000001,DISP=(,CATLG)
//   VOL=SER=?BKVOL1,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//BCDSV2 DD DSN=?UID.BCDS.BACKUP.V0000002,DISP=(,CATLG)
//   VOL=SER=?BKVOL2,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//BCDSV3 DD DSN=?UID.BCDS.BACKUP.V0000003,DISP=(,CATLG)
//   VOL=SER=?BKVOL3,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//BCDSV4 DD DSN=?UID.BCDS.BACKUP.V0000004,DISP=(,CATLG)
//   VOL=SER=?BKVOL4,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//OCDSV1 DD DSN=?UID.OCDS.BACKUP.V0000001,DISP=(,CATLG)
//   VOL=SER=?BKVOL1,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//OCDSV2 DD DSN=?UID.OCDS.BACKUP.V0000002,DISP=(,CATLG)
//   VOL=SER=?BKVOL2,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//OCDSV3 DD DSN=?UID.OCDS.BACKUP.V0000003,DISP=(,CATLG)
//   VOL=SER=?BKVOL3,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//OCDSV4 DD DSN=?UID.OCDS.BACKUP.V0000004,DISP=(,CATLG)
//   VOL=SER=?BKVOL4,SPACE=(CYL,(?CDSSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//JRNLV1 DD DSN=?UID.JRNL.BACKUP.V0000001,DISP=(,CATLG)
//   VOL=SER=?BKVOL1,SPACE=(CYL,(?JNLSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//JRNLV2 DD DSN=?UID.JRNL.BACKUP.V0000002,DISP=(,CATLG)
//   VOL=SER=?BKVOL2,SPACE=(CYL,(?JNLSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//JRNLV3 DD DSN=?UID.JRNL.BACKUP.V0000003,DISP=(,CATLG)
//   VOL=SER=?BKVOL3,SPACE=(CYL,(?JNLSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
//JRNLV4 DD DSN=?UID.JRNL.BACKUP.V0000004,DISP=(,CATLG)
//   VOL=SER=?BKVOL4,SPACE=(CYL,(?JNLSIZE,5)),UNIT=?BKUNIT,
//   MGMTCLAS=?MCDFHSM,STORCLAS=?SCBVOL1
```

Note: Use the job that is shown in Example 12-9 on page 351 only if you plan to use DASD for backup. You might also want to change parameters, such as data set names. This example shows single cluster CDSs. For a multicluster CDS, you need to create the backup data set for every cluster.

Defining the CDS and journal backup environment

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

1. Prevent the CDSs and the journal from being backed up as part of the user data set backup. The CDSs and the journal are backed up separately as specified by the **SETSYS CDSVERSIONBACKUP** command.
2. Modify the block size for CDS backup version data sets.

   Preallocate DASD backup data set copies with a block size that is equal to one-half the track size of the DASD device. For example, the half-track capacity of a 3390 device is 27998.

   If the **BLKSIZE** keyword is specified on the pre-allocated DASD backup data set copies, it must be in the range of 7892 - 32760, inclusively.
3. If the CDSs and journal are SMS-managed, perform the following actions:
   - Place them on volumes that are defined in a storage group with this attribute value:

     AUTO BACKUP ===> NO

     or
   - Associate them with a management class with the following attribute:

     AUTO BACKUP ===> N

     If the CDSs and the journal are non-SMS-managed, issue the **ALTERDS** command to prevent them from being backed up outside of the **CDSVERSIONBACKUP**.
4. Prevent the CDSs and journal from migrating. Allowing the CDSs and the journal to migrate is inadvisable because you might not be able to recover if any of the CDSs are damaged.

   If the CDSs and journal are SMS-managed, perform the following actions:
   - Place them on volumes that are defined in a storage group with this attribute value:

     AUTO MIGRATE ===> NO

     or
   - Associate them with a management class with the following attribute:

     COMMAND OR AUTO MIGRATE ===> NONE
   - If the CDSs and journal are non-SMS-managed, issue the **SETMIG** command to prevent them from migrating.
5. Determine whether your CDSs are backed up by Concurrent Copy. If you want your CDSs to be backed up by Concurrent Copy, you must review the following areas:
   - Ensure that they are associated with a management class Backup Copy Technique attribute of `CONCURRENT REQUIRED` or `CONCURRENT PREFERRED`.
   - Ensure that they are on a DASD volume with a concurrent-copy-capable controller.
   - Ensure that you specify DATAMOVER(DSS) in step 6.

6. Determine whether the data mover for the CDSs is DFSMShsm or DFSMSdss. We recommend DFSMSdss as the data mover because DFSMSdss validates the CDSs during backup and supports Concurrent Copy.
   - If you specify **SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS))**, DFSMShsm invokes DFSMSdss to perform a logical dump of the CDSs and uses sequential I/O to back up the journal. DFSMSdss validates the CDSs while backing them up and uses Concurrent Copy if it was specified in the management class.
   - If you specify **SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM))**, DFSMShsm exports the CDSs and backs up the journal with sequential I/O. The CDSs are not validated during backup.
7. Choose the number of backup versions that you want to keep for the CDSs. The number of backup versions that DFSMShsm keeps is determined by the number that you specify on the **BACKUPCOPIES** subparameter of the **SETSYS CDSVERSIONBACKUP** command.

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

8. Choose the device category (DASD or tape) to which you want DFSMShsm to back up your CDSs and journal. Parallel is faster than serial and required to use Concurrent Copy.
   - If you specify **SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(DASD))**, DFSMShsm always backs up the CDSs in parallel to DASD devices. If you are backing up the CDSs and the journal to DASD, you must preallocate the backup version data sets. You can preallocate the DFSMShsm CDS and journal data set by running the starter set job ALLOCBK1 (see 3.2.2, “ALLOCBK1” on page 50) before you start DFSMShsm.
   - If you specify **SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE))**, DFSMShsm backs up the CDSs to tape.

     Whether tape CDS backups are in parallel is determined by the data mover that you specify and the optional PARALLEL|NOPARALLEL option for DFSMShsm CDS backup.
   - If you specify **SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS))**, CDSs are backed up to tape in parallel. Concurrent Copy can be used.
   - If you specify **SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM))**, DFSMShsm backs up the CDSs serially. Concurrent Copy is not available, and the CDSs are not validated during backup.
   - If you specify **SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM) PARALLEL)**, DFSMShsm backs up the CDSs to tape in parallel. Concurrent Copy is not available, and the CDSs are not validated during backup.

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

9. Determine the names for the backup data sets.

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

   Example 12-10 is an example of the **SETSYS CDSVERSIONBACKUP** command and its subparameters, as it appears in PARMLIB member ARCCMDxx.

   Example 12-10 Sample ARCCMD configuration for CDS backups

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

After you complete all of these steps successfully, on your next start, DFSMShsm will use these pre-allocated data sets to hold the backup copies.

Backing up the CDS and journal manually

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

Remember the following points before you issue the command to back up the CDSs and journal manually:

- Do not issue the command during intense DFSMShsm activity because the CDSs cannot change while they are being backed up.
- After you issue the command, the only way to prevent the backup is to stop DFSMShsm.
- The structural integrity of the CDSs is validated only if you specified DATAMOVER(DSS).

Use the DFSMShsm **BACKVOL** command to back up the CDSs and journal. With this command, you can perform the following tasks:

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

To back up the CDSs and journal, you can use the following command:

```
HSEND BACKVOL CDS
```

DFSMShsm will use the attributes that are specified in its PARMLIB to define the backup method and media.

If you set DATAMOVER(DSS), a DFSMSdss logical dump is used, so that you can use Concurrent Copy. This approach might reduce any serialization delays that are introduced by the exclusive enqueue that is placed on the CDSs while the backup occurs. Additionally, the CDSs will be validated.

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

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

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

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

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

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

12.1.5 VSAM record-level sharing

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

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

Requirements for CDS RLS serialization

CDSs that are accessed in RLS mode enqueue certain resources differently than CDSs that are accessed in non-RLS mode. Before you consider implementing RLS for your CDSs, you must ensure that all of the following criteria are met:

- Global resource serialization (GRS) or an equivalent function is implemented.

  > **Note:** GRS is required only if you are sharing CDSs between multiple systems. If you have one or more hosts on a single system, it is not required.

- All operating systems running DFSMShsm must be coupling facility-capable, and the processors must share access to the coupling facility.
- Your CDSs must be SMS-managed.
- All processors in the installation must access the CDSs in RLS mode.
- All DFSMShsm hosts must specify CDSSHR=RLS in the DFSMShsm startup procedure.
- You must not convert the ARCGPA/ARCRJRN reserve to an enqueue for performance reasons.
- The CDSs storage class must indicate the coupling facility to use.
- The CDSs must not be keyrange KSDSs.
- You must know how to implement recovery for RLS data sets.
- You must specify DFSMSdss as the data mover for **CDSVERSIONBACKUP**.
- If CDS backup is directed to tape, the **PARALLEL** parameter must be used.

Making your CDSs RLS-eligible

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

The following example shows how to use the **IDCAMS ALTER** command to make the CDSs that we previously defined RLS-eligible:

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

Example 12-11 is an example of how we can use the **DEFINE** command when we initially set up our DFSMShsm environment. We show only the definition for the MCDS, but the same definition must be done for the other CDSs.

Example 12-11 Sample IDCAMS DEFINE command for the DFSMShsm data set

```
DEFINE CLUSTER (NAME(HSM.MCDS) VOLUMES(HG6622) -
        CYLINDERS(100) FILE(HSMMCDS) -
        RECORDSIZE(435 2040) FREESPACE(0 0) -
        INDEXED KEYS(44 0) SHAREOPTIONS(3 3) -
        SPEED BUFFERSPACE(530432) -
        UNIQUE NOWRITECHECK LOG(NONE)) -
        DATA(NAME(HSM.MCDS.DATA) -
        CONTROLINTERVALSIZE(12288)) -
        INDEX(NAME(HSM.MCDS.INDEX) -
        CONTROLINTERVALSIZE(4096))
```

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

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

Removing keyrange CDSs

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

If your CDSs are defined with keyranges, you can perform a CDS reorganization, and remove the **KEYRANGE** parameter from the **DEFINE** command.

Determining the CDS serialization technique

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

```
HSEND Q CDS
```

If you are using RLS, the following messages are the first messages that are returned as a result of your **QUERY CDS** command:

```
ARC0101I QUERY CONTROLDATASETS COMMAND STARTING ON HOST=2
ARC0947I CDS SERIALIZATION TECHNIQUE IS RLS
```

RLS implementation

We show the steps that we took to implement VSAM RLS for our CDSs. In addition to the requirements that we describe in “Requirements for CDS RLS serialization” on page 356, other considerations must be met. We make assumptions about your system knowledge. You must be familiar with SMS for VSAM RLS, SMS constructs, SMS classes, SMS configuration, and the coupling facility cache and lock structures. We do not recommend that you undertake these steps until you consider how RLS implementation can affect your system.

We recommend that any pre-existing VSAM RLS structures are used for accessing the DFSMShsm CDSs in RLS mode. No benefit exists to assigning the DFSMShsm CDSs to unique structures.

Define the SHCDSs

Shared Control Data Sets (SHCDSs) are linear data sets that contain information to allow processing if a system failure might affect RLS. They also act as logs for sharing support.

You must consider size for the SHCDS and you must adhere to a naming convention. For comprehensive information about defining these data sets, see the *OS/390 V2R10.0 DFSMSdfp Storage Administration Reference,* SC26-7331.

We used the JCL in Example 12-12 to allocate the SHCDS.

Example 12-12 Sample SHCDS allocation job

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

Coupling facility cache and lock structures

If you need to allocate cache and lock structures specifically for DFSMShsm, use the following recommendations:

- Cache structure: The size of the cache structure must be a minimum of 1 MB per DFSMShsm host in the HSMplex. For example, if 10 hosts are in the HSMplex, the cache structure must be a minimum of 10 MB.
- Lock structure: Use Table 12-1 to determine the minimum size for the lock structure.

Table 12-1 Minimum size for the lock structure

Number of DFSMShsm hosts	Minimum lock structure size
Fewer than 8	1 MB
At least 8, but not more than 23	2 MB
At least 24, but not more than 32	3 MB
More than 32	4 MB

For example, if 10 DFSMShsm hosts are in the HSMplex, the lock structure must be a minimum of 2 MB.

Certain changes to your CFRM policy definitions will be necessary. The DFSMShsm policy needs to be defined. We added the structures for cache and locking to our current CFRM policy definitions by using the administrative data utility IXCMIAPU, as shown in Example 12-13.

Example 12-13 Adding structures for cache and locking to the CFRM policy definitions

```
//DEFCFRM1  JOB (999,POK),'CFRM',CLASS=A,REGION=4096K,
//             MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=MHLRES5
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
     DATA TYPE(CFRM) REPORT(YES)
     DEFINE POLICY NAME(CFRM19) REPLACE(YES)
        CF NAME(CF01)
               TYPE(009672)
               MFG(IBM)
               PLANT(02)
               SEQUENCE(000000040104)
               PARTITION(1)
               CPCID(00)
               DUMPSPACE(2048)
        CF NAME(CF02)
               TYPE(009672)
               MFG(IBM)
               PLANT(02)
               SEQUENCE(000000040104)
               PARTITION(1)
               CPCID(01)
               DUMPSPACE(2048)
        STRUCTURE NAME(IGWLOCK00)
               SIZE(28600)
               INITSIZE(14300)
               PREFLIST(CF02,CF01)
               REBUILDPERCENT(75)
        STRUCTURE NAME(HSMCACHE1)
               SIZE(64000)
               INITSIZE(32000)
               PREFLIST(CF01,CF02)
               REBUILDPERCENT(75)
        STRUCTURE NAME(HSMCACHE2)
               SIZE(64000)
               INITSIZE(32000)
               PREFLIST(CF02,CF01)
               REBUILDPERCENT(75)
```

Note: The code in Example 12-13 on page 359 does not represent the entire policy data for the CFRM data set. It represents the CFRM policy that specifies the requirements for the DFSMShsm RLS structures.

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

Alter the SMS configuration

You must update the SMS configuration with the coupling facility cache structure names that you defined. Use the ISMF panels. Enter 8 from the ISMF Primary Option menu for storage administrators to display the CDS Application Selection panel (Figure 12-2).

```
                          CDS APPLICATION SELECTION
Command ===>

To Perform Control Data Set Operations, Specify:
  CDS Name . . 'SYS1.SMS.MHLRES3.SCDS'
                            (1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:
  7  1. Display       - Display the Base Configuration
     2. Define        - Define the Base Configuration
     3. Alter         - Alter the Base Configuration
     4. Validate      - Validate the SCDS
     5. Activate      - Activate the CDS
     6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
     7. Cache Update  - Define/Alter/Delete CF Cache Sets
     8. Lock Display  - Display CF Lock Structure Names for all CF Lock Sets
     9. Lock Update   - Define/Alter/Delete CF Lock Sets
If CACHE Display is chosen, Enter CF Cache Set Name . . *
If LOCK Display is chosen, Enter CF Lock Set Name . . . *
                             (1 to 8 character CF cache set name or * for all)
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 12-2 CDS Application Selection panel for cache

Enter option 7 to define the cache sets that relate to your coupling facility cache structure names (Figure 12-3).

```
                           CF CACHE SET UPDATE                  PAGE 1 OF 1
Command ===>

SCDS Name  : SYS1.SMS.MHLRES3.SCDS
Define/Alter/Delete CF Cache Sets:       ( 002  Cache Sets Currently Defined )

 Cache Set                      CF Cache Structure Names
 HSM1      HSMCACHE1

 HSM2      HSMCACHE2











More CF Cache Sets to Add?  . . N  (Y/N)
Use ENTER to Perform Validation; Use UP/DOWN Command to View other Pages;
Use HELP Command for Help; Use END Command to Save and Exit.
```

Figure 12-3 CF Cache Set Update panel

The coupling facility cache structure names must be the names that you previously defined (see "Coupling facility cache and lock structures" on page 358).

Storage class changes

We took the option of altering the storage class where our CDSs were defined. The storage class is SC54GRT, and we altered it by entering option 4 from the Storage Class Application Selection panel (Figure 12-4 on page 362).

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

To perform Storage Class Operations, Specify:
  CDS Name  . . . . . . . 'SYS1.SMS.MHLRES3.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Storage Class Name  . . CDSRLS    (For Storage Class List, fully or
                                    partially specified or * for all)
Select one of the following options  :
  4  1. List          - Generate a list of Storage Classes
     2. Display       - Display a Storage Class
     3. Define        - Define a Storage Class
     4. Alter         - Alter a Storage Class
     5. Cache Display - Display Storage Classes/Cache Sets
     6. Lock Display  - Display Storage Classes/Lock Sets
If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
If Cache Display is Chosen, Specify Cache Structure Name . .
If Lock Display is Chosen, Specify Lock Structure Name . . .
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 12-4 Altering a storage class

We added the coupling facility cache set information (Figure 12-5).

```
                            STORAGE CLASS ALTER
Command ===>

SCDS Name . . . . . : SYS1.SMS.MHLRES3.SCDS
Storage Class Name  : CDSRLS

To ALTER Storage Class, Specify:

  Guaranteed Space  . . . . . . . . . Y           (Y or N)
  Guaranteed Synchronous Write  . . . N           (Y or N)
  Multi-Tiered SG . . . . . . . . . .             (Y, N, or blank)
  Parallel Access Volume Capability   N           (R, P, S, or N)
  CF Cache Set Name . . . . . . . . . HSM1        (up to 8 chars or blank)
  CF Direct Weight  . . . . . . . . . 5           (1 to 11 or blank)
  CF Sequential Weight  . . . . . . . 3           (1 to 11 or blank)
  CF Lock Set Name  . . . . . . . . .             (up to 8 chars or blank)
  Disconnect Sphere at CLOSE  . . . . N           (Y or N)
```

Figure 12-5 Storage Class Alter panel page 2

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

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

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

Altering the IGDSMSxx PARMLIB member

To specify to SMS that the RLS address space, SMSVSAM, starts at IPL, and to include other information, we added the information in Example 12-14 to our PARMLIB data set, member IGDSMS54.

Example 12-14 Sample IGDSMSxx PARMLIB to implement RLS

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

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

12.1.6 RACF FACILITY classes

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

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

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

Activating the SMSVSAM address space

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

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

The information that is returned informs you whether your system is active. The returned information also shows the status of the SMS complex.

Activating the SHCDS

If your SHCDS is already active, you do not need to activate it. However, if you are implementing RLS for the first time, issue the commands in Example 12-15, after you modify the commands to represent the names you chose for your SHCDSs.

Example 12-15 Activating SHCDSs

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

These commands perform the following functions:

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

DFSMShsm procedure changes

Add the following information to the PROC statement for your DFSMShsm started procedure member in SYS1.PROCLIB:

```
CDSSHR=RLS
```

Add the following information to your EXEC statement:

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

RLS implementation checklist

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

1. Ensure that all sharing systems are at the prerequisite levels of software.
2. If the SHCDSs are not defined, use IDCAMS to define SHCDSs and ensure that SHAREOPTION(3,3) is specified.
3. Define your coupling facility cache and lock structures.
4. Alter your SMS configuration to include the cache set information.
5. Create or alter a storage class that contains the cache set information.
6. Assign the CDSs to the new or altered storage class.
7. Alter SYS1.PARMLIB member IGDSMSxx with the RLS parameters.
8. Define the new RLS profiles in the RACF FACILITY class and authorize users.
9. Schedule an IPL of the MVS systems and ensure that the SMSVSAM address space becomes active.
10. Activate the SHCDSs.

12.2 Multicluster control data sets

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

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

We recommend the use of non-keyrange multicluster CDSs. The following section applies only to non-keyrange data sets.

For more information about defining CDSs without keyranges, see “Removing keyrange CDSs” on page 357.

12.2.1 Multicluster performance considerations

Multicluster CDSs are a related group of VSAM clusters with the following requirements:

- CDSs cannot be empty when you start DFSMShsm. The MCDS must be defined so that the high key for the first cluster is greater than X'10' \ C'MHCR'.
- All clusters in a multicluster CDS must be cataloged in the same user catalog.
- All clusters must be on DASD of the same category.
- Only the primary allocation can be specified for the index and data; no secondary allocation can be specified for either the index or data.
- The index and data are limited to one volume; they cannot be on different volumes.
- For a multicluster BCDS, the RECORDSIZE and data CONTROLINTERVALSIZE for each cluster depend on the maximum number of backup versions that you want to be able to keep for any data set.
- CDSs that are accessed by RLS cannot be defined as keyrange data sets. If previous CDSs are defined as multicluster and use keyranges, they must be redefined without keyranges. DFSMShsm will dynamically calculate the key boundaries for the redefined CDSs.

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

12.2.2 Conversion steps

Many steps are required to implement a multicluster CDS. The following steps assume a familiarity with DFSMShsm and a requirement at your site to adopt multicluster CDSs.

Follow these steps to implement a multicluster environment:

1. Stop DFSMShsm on all z/OS images.
2. Back up the CDS with the **`AMS EXPORT`** command.
3. Determine whether to split the CDS into two, three, or four clusters.
4. Modify the JCL that is shown in Example 12-16 to suit your installation's requirements.

Example 12-16 Sample JCL for non-keyrange multicluster MCDS

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

```
      (NAME(DFHSM.RLS.MCDS1.DATA) -
       CONTROLINTERVALSIZE(4096)) -
   INDEX -
      (NAME(DFHSM.RLS.MCDS1.INDEX) -
       CONTROLINTERVALSIZE(4096))

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

5. Copy the old CDS to the new multicluster CDS with the **AMS REPRO** command, as shown in Example 12-17.

Example 12-17 Sample JCL for non-keyrange multicluster REPRO command

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

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

> **Note:** Do not delete the current CDS. Instead, maintain it for a period until you determine that the new CDS is valid.

8. Make backup copies of the CDSs.
9. Monitor the growth of the multicluster CDS.

After you complete all of these steps successfully, DFSMShsm will use the multicluster CDSs.

12.3 CDS backup procedures

You can back up DFSMShsm CDSs in various ways, depending on your hardware and system level. System performance directly relates to the time that it takes to back up the CDSs and the journal because all DFSMShsm activity and all JES2/JES3 setup and initialization activity are suspended during this process. By improving the backup performance of CDSs and the journal, you can decrease the time that DFSMShsm is unavailable to process requests. To decrease the required time to back up the CDSs and the journal data sets, certain backup techniques are available. We describe the use of Concurrent Copy, SnapShot, and virtual Concurrent Copy as your backup procedures. We also describe the CDS backup improvements that were introduced in the last z/OS versions.

12.3.1 Concurrent Copy

Concurrent Copy is a combined hardware, Licensed Internal Code (LIC), and software systems management solution that creates data dumps or copies while user processing continues.

When DFSMShsm uses Concurrent Copy to back up the CDSs, the period during which system functions are suspended is dramatically reduced. DFSMShsm reserves the data sets only for the time that it takes to initialize a Concurrent Copy session for each of them and to back up and nullify the journal. (DFSMShsm always backs up the journal itself; it does not use DFSMSdss and consequently Concurrent Copy.) After these operations complete, DFSMShsm releases the CDSs for other processing but continues with the backup because the data sets are now protected by Concurrent Copy. This action can reduce the period during which DFSMShsm functions (and in JES3 systems, job initiation) are suspended to only a few minutes. This capability makes it possible to reduce the time that is required to restart DFSMShsm after a failure if a backup copy of one of the CDSs is required.

To use Concurrent Copy for CDS backup, you must perform the following tasks:

- Allocate the CDSs on system-managed volumes that are attached to DASD subsystems that have Concurrent Copy available
- Ensure that the CDSs receive a management class as described next
- Specify that DFSMShsm will use DFSMSdss as the data mover

DFSMShsm uses Concurrent Copy for automatic backup when processing system-managed data sets only. If your CDSs are not located on system-managed volumes, you will need to reallocate them.

When you reallocate the DFSMShsm CDSs, specify (or ensure that your ACS routines select) a storage class with GUARANTEED SPACE=YES and a management class that contains the following specifications:

- Expiration limits (EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS) and RETENTION PERIOD are set to `NOLIMIT`.
- COMMAND or AUTO MIGRATE is set to `NONE`.

- ADMIN or USER COMMAND BACKUP is set to `NONE`.
- BACKUP COPY TECHNIQUE is set to `CONCURRENT PREFERRED`. We recommend that you specify *PREFERRED* rather than REQUIRED because PREFERRED ensures that DFSMShsm backs up the CDSs even if Concurrent Copy is not available. If you specify *CONCURRENT REQUIRED* and Concurrent Copy is not available (for example, because of a cache failure), DFSMShsm does not back up the CDSs.

  If you use tape as your backup device for the CDSs, also specify the `PARALLEL` option to request DFSMShsm to back up the CDSs in parallel. If you do not specify *PARALLEL*, DFSMShsm backs up the CDSs to tape one-by-one, reducing the benefit of Concurrent Copy. DFSMShsm always backs up the CDSs to DASD in parallel. If contention for tape drives is a problem in your installation, consider changing your CDS backup device to DASD when you implement Concurrent Copy.

  Concurrent Copy makes it possible to take backups more frequently, reducing the required time to bring a backup copy “up-to-date” from the DFSMShsm journal. You can take a backup of the CDSs at any time by using the DFSMShsm **`BACKVOL CDS`** command. In addition to using this command to take more frequent CDS backups, you might also need to use this command if a system problem causes CDS backup to fail after the Concurrent Copy sessions are initialized.

12.3.2 Virtual Concurrent Copy

Virtual Concurrent Copy provides a copy operation that is similar to Concurrent Copy by using a combination of RAMAC Virtual Array (RVA), SnapShot, and DFSMSdss. If you are using Concurrent Copy to back up the CDSs, and the device they reside on is an RVA, you do not need to take any additional action. Virtual Concurrent Copy will be automatically invoked.

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

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

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

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

So for virtual Concurrent Copy to succeed, even though the Concurrent Copy criteria are met, you must predefine at least one WSDS, but it is more likely that you will need to allocate multiple WSDSs.

The allocation requirements, catalog search, system data mover (SDM), recommendations, performance, and restrictions for WSDSs are comprehensively covered in *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268. We recommend that you obtain a copy of this publication before you implement virtual Concurrent Copy in your installation.

12.4 CDS extended address volume

As your systems and DFSMShsm-managed data grow, more CDS records will be created to store information about migration, backup, dump, and other DFSMShsm functions that relate to data sets. Depending on the amount of data that is managed by DFSMShsm, the 4 GB limit for VSAM data sets might not be enough to store all CDS records.

To avoid this issue, with DFSMShsm, you can use extended addressability VSAM data sets and extended address volumes (EAVs) to store your CDSs.

With the extended addressability VSAM data set, you can allocate VSAM data sets beyond the 4 GB limit. To create CDSs larger than 4 GB, you must define them as SMS-managed data sets, and assign a data class with “extended addressability” set to `Y`.

The EAVs are volumes that are defined to z/OS with more than 65,520 cylinders, and all of the data sets that are created beyond this line are cylinder-managed data sets. No special requirements exist for using EAVs, but we recommend that you use SMS-managed CDSs, so you can take the advantage of the large space that is available on the volume.

z/OS V1R13 supports volumes as large as 1 terabyte, which means that MCDS and BCDS are limited to 4 TB each because each data set can have up to four clusters. The OCDS is limited to 1 TB because it must be a single-cluster CDS.

12.5 CDS new backup technique

Starting in z/OS V1R13, DFSMShsm introduces a new backup technique, called *nonintrusive journal backup*. With this new feature, you can reduce the time that DFSMShsm activities are unavailable during CDS backup.

Before V1R13, the CDSs and journal were quiesced during CDS backups. Quiescing the CDSs and journal was necessary because the journal must be backed up in real time (that is, no Concurrent Copy option) and therefore holds up DFSMShsm activity.

As the journal data set is written sequentially, the only changes to the journal are appends to the end of the data set. Therefore, after the data is written to the journal, it will not be changed or updated. Nonintrusive journal backup allows DFSMShsm to back up the data that is already recorded in journal data set, without quiescing, and consequently prevents new journal records from being recorded.

By using this new backup technique, DFSMShsm will no longer quiesce journal data sets during the total journal backup. Instead, DFSMShsm will mark the last record that is written to the journal data set before the backup starts. Then, DFSMShsm quiesces the journal while it backs up the control record, and then releases it from quiesce, while asynchronous backup is performed against all records before the logical end mark.

After DFSMShsm finishes backing up all records before the logical end mark, it will quiesce the journal and CDSs, and back up the remaining records and CDS data. Journal and CDS updates will be prevented during this time.

To make it possible, the journal data set is now backed up before the CDSs backup. This change is necessary so all of the journal and CDS updates that are performed during the asynchronous copy of journal records are also backed up during CDS backups and journal synchronous copy.

By allowing DFSMShsm activity while backing up the existing data in the journal data set and by quiescing the journal only when backing up the latest recorded data, you can significantly reduce the time that DFSMShsm activities are unavailable.

Figure 12-6 shows the differences between the quiesced journal backup and the non-intrusive journal backup.

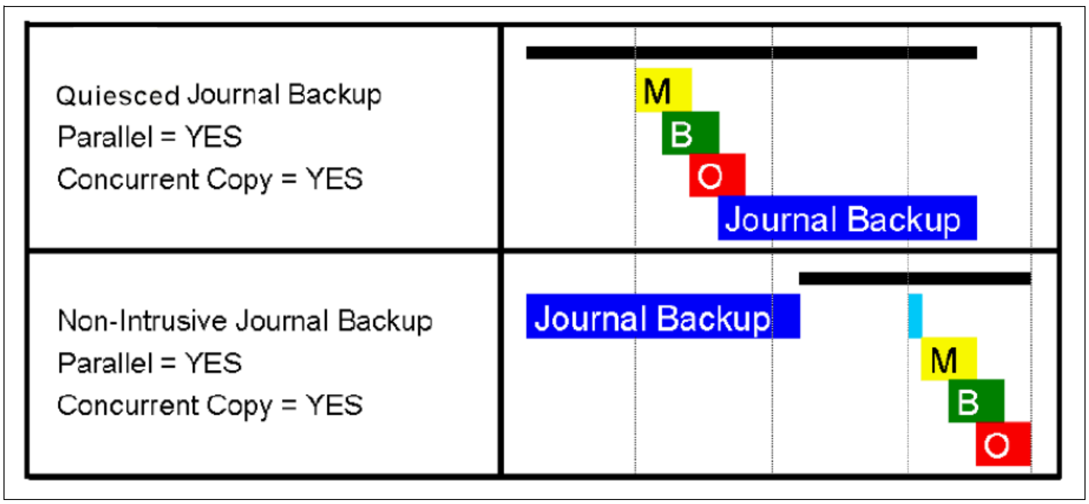

Figure 12-6 Quiesced journal backup versus non-intrusive journal backup

In Figure 12-6, the black line is the time that the journal and CDSs are quiesced for backup. The light blue line shows the journal updates that are performed during asynchronous journal backup.

The elapsed time for nonintrusive journal backup is approximately the same as for the quiesced backup. The advantage of using nonintrusive journal backup is the reduced unavailability time for DFSMShsm processing.

To use nonintrusive journal backup, your system must be on z/OS V1R13. You must meet the following requirements:

- SETSYS JOURNAL(RECOVERY) is set on your DFSMShsm PARMLIB
- SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS)) is set on your DFSMShsm PARMLIB.
- Your CDSs must be SMS-managed.
- The management class of the CDSs must specify Concurrent Copy technique.
- When running on multiple logical partitions (LPARs) with z/OS images before V1R13, you must install toleration APAR OA32293.

If any of these conditions are not met, DFSMShsm will automatically quiesce the journal and the CDSs for backup.

ARCCAT release

ARCGPA/ARCCAT is obtained (shared) whenever a function wants to update the DFSMShsm CDS records. In certain cases, the resource is held for a long time. Typically, when the CDS backups appear to be stopped, the process is simply waiting to exclusively obtain ARCGPA/ARCCAT while another function owns the resource as shared.

CDS backups can be delayed for an extended time if the **BACKVOL CDS** command was issued during a migration or backup of a large data set, tape audits, or other DFSMShsm activities that can run for several minutes or hours.

Starting on z/OS V1R13, DFSMShsm will handle ARCGPA/ARCCAT resources so that when a CDS backup process starts in a host, all DFSMShsms in the HSMplex finish their currently running CDS updates, and immediately release ARCGPA/ARCCAT resources.

This approach enables CDS backup to start immediately. After the CDS backup is complete, all of the processes that were running before the backup start to reacquire ARCGPA/ARCCAT resources and continue processing.

Figure 12-7 shows ARCCAT resource allocation during DFSMShsm CDS backup processing.

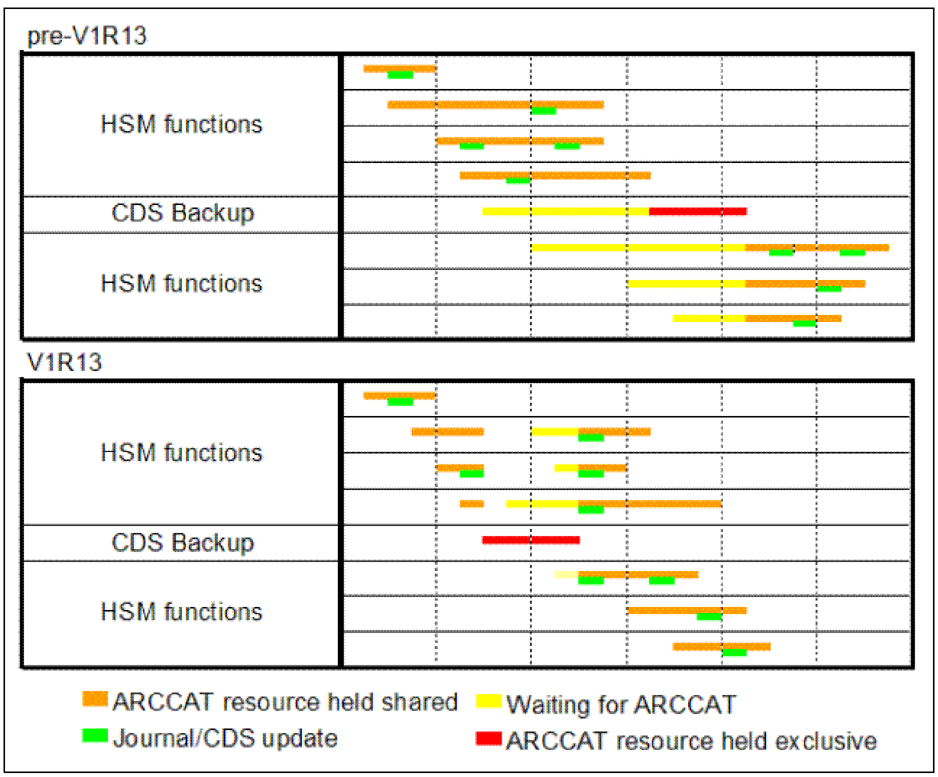

Figure 12-7 ARCCAT resource allocation during DFSMShsm processing

When you are using the ARCCAT release in z/OS V1R13, the patches that were used on older DFSMShsm releases to periodically release ARCCAT will not be effective.

To use the ARCCAT release, you must be running z/OS V1R13 or higher. If any of your hosts are under this version, apply the coexistence APAR OA32293. The ARCCAT release is not fully effective until all hosts are running z/OS V1R13 or higher because only those hosts can perform the release.

The ARCCAT release depends on cross-system coupling facility (XCF) services to communicate the start of CDS backup to other HSM hosts. In the absence of XCF services, the ARCCAT release will be performed only on the host that performs the CDS backup.

13

Monitoring and reporting

You can monitor DFSMShsm in real time or historically in a number of ways. DFSMShsm commands provide a comprehensive view of what is happening, what happened, and what needs to happen.

In this chapter, we explain how to use the DFSMShsm commands to collect information about DFSMShsm processing and monitor DFSMShsm activity.

Each of these commands gathers information from different sources within DFSMShsm, such as the migration control data set (MCDS), backup control data set (BCDS), offline control data set (OCDS), functional statistics records (FSRs), or the DFSMShsm address space.

13.1 List command

The DFSMShsm **LIST** command obtains its information from the control data sets (CDSs): MCDS, BCDS, and OCDS. You can list the following categories of information:

- Aggregate backup and recovery support (ABARS) activity
- Backup volumes
- Data sets
- Dump classes
- Dump volumes
- Host information
- Migration information
- Primary volume information
- Tape volume information
- User authorization

Many options are available for each of the **LIST** commands; we do not show every variation. The commands that we show cover each area of DFSMShsm for which the **LIST** command gathers information. Several of the commands use only the MCDS or BCDS to gather information. Other commands use both the MCDS and BCDS. You might want to investigate the commands further to determine whether additional parameters suit your installation requirements.

To use the commands that are listed in this chapter, you must have the necessary RACF or DFSMShsm AUTH. If you do not have this authority, or do not know how to get it, see Chapter 5, “Implementing DFSMShsm security” on page 79.

Certain **LIST** commands might result in a large amount of data to be displayed, interfering with your terminal displays, or increasing DFSMShsm SYSOUT. You can select the output that you want for the specified command by issuing one of the commands that are shown in Example 13-1.

Example 13-1 Select output to terminal, a data set, or SYSOUT class

```
HSEND request TERM
HSEND request ODS('dsname')
HSEND request SYSOUT(class)
```

The **LIST** command can create a large amount of output, so be sure to direct any output that might be large, for example, a list of all migration volumes, to an output data set.

When you direct the output to DASD, you must specify the fully qualified data set name. If the output data set does not exist, DFSMShsm dynamically allocates it. If the specified output data set exists, DFSMShsm appends the output to the end of the data set.

Aggregate backup and recovery support

If you are running aggregate backup and recovery support (ABARS) in your environment, in certain situations, it will be necessary to list and report ABARS information for accounting or disaster recovery. Use the following command to list information about an ABARS aggregate group and ABACKUP and ARECOVER information:

```
HSEND LIST AGGREGATE(PAY1) VERSION(0001) ODS(MHLRES7.ABARS.LIST)
```

Example 13-2 on page 375 is a sample output from the previous command.

Example 13-2 Sample LIST AGGREGATE output

```
-- DFSMSHSM CONTROL DATASET AGGREGATE BACKUP AND RECOVERY VERSION LISTING -----
      AT 19:39:10 ON 2012/08/31 FOR SYSTEM=SC64

ABR RECORD KEY = MHLRESA.2012240000101
AGGREGATE GROUP NAME = MHLRESA                          DFSMSHSM VERSION = VD.1
AGGREGATE ACCOUNT CODE =
ABACKUP DATE = 2012/08/27     ABACKUP TIME = 12:16:39
VERSION = 0001                COPY NUMBER = 01
UNIT NAME = 3590-1            NUMBER OF COPIES = 01
SOURCE SYSTEM = SC64
MANAGEMENT CLASS = MC54NMIG.....................
REMOTE DESTINATIONS =         DESTINATION ID                TRANS. OK
                              NONE

CONTROL FILE NAME = HSM.DMP.C.C01V0001
DFSMSDSS DATA FILE NAME = HSM.DMP.D.C01V0001
INTERNAL I/O DATA FILE NAME = NONE
ACTIVITY LOG/INSTRUCTION FILE NAME = HSM.DMP.I.C01V0001

ABACKUP ACTIVITY LOG DATA SET NAME = HSMACT.H2.ABACKUP.MHLRESA.D12240.T121639
INSTRUCTION DATA SET NAME = MHLRESA.ABARS.INSTRUCT
FILTER OUTPUT DATA SET NAME = MHLRESA.ABARS.OUTPUT

CONTROL FILE VOLSERS =
 VOLS=  THS014
 LIBS=  LIB2
DFSMSDSS DATA FILE VOLSERS =
 VOLS=  THS006
 LIBS=  LIB2
```

Backup and migration

You can also use the **LIST** command to gather information about backups, migration, and recalls, to help you to identify the high-level qualifiers (HLQs) and applications that perform more backups, or request more recalls. You can use this information to help you to manage retention periods on primary volumes, and to identify data sets that are being backed up incorrectly.

To list the backup volumes that are assigned to DFSMShsm, you can issue the following command:

```
HSEND LIST BACKUPVOLUME ODS(MHLRES5.LBV)
```

To retrieve information about a specific application or HLQ, you can add the **LEVEL** parameter to restrict the listing to a determined mask of data sets:

```
HSEND LIST LEVEL(MHLRES7) BCDS ODS(MHLRES7.DSN3)
```

Note: This command will display all BCDS information about HLQ MHLRES7 and send output to MHLRES7.DSN3. This command can return many entries, and displaying the output directly to your terminal is not recommended.

Example 13-3 on page 376 shows the output from the previous command.

Example 13-3 Sample LIST LEVEL BCDS output

```
- DFSMSHSM CONTROL DATASET - BACKUP DATASET-- LISTING ----- AT 13:49:27 ON
12/09/04 FOR SYSTEM=SC64


DSNAME = MHLRES7.ACSTEST.LISTDS                        BACKUP FREQ = ***, MAX
ACTIVE BACKUP VERSIONS = ***

BACKUP VERSION DATA SET NAME                BACKUP FROM   BACKUP   BACKUP   SYS
GEN  VER  UNS/ RET  BACKUP NEW
                                            VOLUME VOLUME DATE     TIME     CAT
NMBR NMBR RET  DAYS PROF   NAME

HSM.BACK.T011418.MHLRES7.ACSTEST.A0313      VT0032 MLD10C 10/11/09 18:14:01 YES
000  001   NO ***** NO     N**

TOTAL BACKUP VERSIONS = 0000000001
```

The **LIST** command also supports the use of the **SELECT** parameter. With this parameter, you can select only the information that you need from DFSMShsm CDSs. The **SELECT** parameter can include specific volume names, age, tape status, and other information.

In the following command, we show how to use the **LIST LEVEL MCDS** command with the **SELECT** parameter with **VOLUME** to list migration information about a specific volume. We show you how to use **AGE** to determine the last referred to date range that you want to list. This command lists MCDS information for MHLRES5.PAY1 data sets:

```
HSEND LIST LEVEL(MHLRES5.PAY1) MCDS SELECT(VOLUME(HG661A) AGE(10 50))
```

This command will list all migration information about data sets that were originally allocated on HG661A, and not referenced in at least 10 days and no more than 50 days. Example 13-4 shows the example output.

Example 13-4 Sample LIST MCDS LEVEL output

```
- DFSMSHSM CONTROL DATASET - MIGRATED DATASET-- LISTING ----- AT 14:47:52 ON
12/09/04 FOR SYSTEM=SC64

 DATASET NAME                                MIGRATED  LAST REF MIGRATED  TRKS
QTY   TIMES  DS SDSP   QTY    LAST MIG
                                            ON VOLUME   DATE     DATE   ALLOC 2K
BLKS  MIG  ORG  DS  16K BLKS  VOLUME

 MHLRES5.PAY1.SALARY1                          HSM14D  12/08/19 12/08/20 000001
0000001 001   PS  NO   ******   ******
- DFSMSHSM CONTROL DATASET - SUMMARY-- LISTING ----- AT 14:47:52 ON 12/08/30 FOR
SYSTEM=SC64

                                         MIGRATED       TRACKS      K-BYTES
                                         DATA SETS     MIGRATED     MIGRATED
                                      0000000001      000000001     00000001


 ----- END OF - MIGRATED DATASET - LISTING -----
```

To list migration information, including data sets that were recalled to a primary volume, you can use the following command:

```
HSEND LIST DATASETNAME MCDS INCLUDEPRIMARY ODS(MHLRES7.DSN4)
```

Example 13-5 shows the output that is generated by the previous command.

Example 13-5 Sample output from LIST DATASETNAME MCDS command

```
- DFSMSHSM CONTROL DATASET - MIGRATED DATASET-- LISTING ----- AT 14:58:42 ON 12/

DATASET NAME                                     MIGRATED  LAST REF MIGRATED  TRKS
QTY  TIMES  DS SDSP   QTY    LAST MIG
                                                ON VOLUME   DATE     DATE   ALLOC  2K
BLKS MIG  ORG  DS  16K BLKS  VOLUME

CVERNON.EAVTEST.EFSAMO                             ONLINE  09/08/14 09/08/14 0000015
0000000 00002 PS  NO   ******   ******
DB9CD.DSNDBC.DB2R7.DB2R7.I0001.A001                ONLINE  09/10/07 09/10/07 0000005
0000000 00003 VS  NO   ******   ******
HAIMO.HSMMIG.TEST1                                 ONLINE  10/05/20 10/05/20 0000001
0000000 00001 PO  NO   ******   ******
HERING.ALLQUOTA.UNLOAD                             ONLINE  08/03/18 08/03/19 0000001
0000000 00002 PS  NO   ******   ******
HERING.CP.TESTFILE                                 ONLINE  08/03/19 08/03/19 0000001
0000000 00001 PS  NO   ******   ******
```

All recalled data sets show MIGRATED ON VOLUME as ONLINE. This list includes how many times the data set was migrated, migrated date, migrated volume, and other attributes.

The **LIST** command can also help you to identify the data sets that are stored in a specific tape, so you can take the necessary actions if a tape is damaged, or lost. To list data set information from a backup tape, you can use the following command, which lists the tape table of contents (TTOC):

```
HSEND LIST TTOC(volser) DATASETINFORMATION ODS('MHLRES7.DSN8')
```

This command will show all of the data sets that are stored on the specific tape. The data sets that you see in Example 13-6 are backups that are taken by DFSMShsm from production data sets.

Example 13-6 Output from LIST TTOC command

```
- DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 17:10:23 ON 12/09/0

VOLSER    UNIT    VOL      REUSE     VALID   PCT    VOL    RACF  PREV    SUCC
          NAME    TYPE   CAPACITY    BLKS    VALID  STATUS       VOL     VOL
THS000   3590-1  D(01) 0000028900 0000268759 100    PART    NO   *NONE*  *NONE*

            DATA SET NAME                          NUM BLOCKS  RELATIVE FBID  VS
   HSM.BACK.T342400.MHLRES7.NONSMSX.A1001          0000002631     0000008     NO
   HSM.BACK.T552900.MHLRES7.NONSMSZ.A2234          0000002631     0000010     NO
   HSM.BACK.T273100.MHLRES7.NONSMSA.A2235          0000005261     0000011     NO
   HSM.BACK.T184600.MHLRES7.NONSMSA.A2234          0000002631     0000012     NO
   HSM.BACK.T585600.MHLRES7.NONSMSB.A2234          0000002631     0000013     NO
   HSM.BACK.T025700.MHLRES7.NONSMSV.A2234          0000002631     0000014     NO
```

To identify which data sets the backup or migration version refers to, you can use the following command:

```
HSEND FIXCDS C hsmdataset DISPLAY
```

For migration data sets, use D instead of C. Then, look at offset +0000 to check the data set name. See Example 13-7.

Example 13-7 Output from FIXCDS C command

```
MCH=   017C2400 CA0E367A BF26466B CA0E367A BF26466B
 * @                                *
+0000  D4C8D3D9 C5E2F74B D5D6D5E2 D4E2E74B C2D2D740 40404040 40404040 40404040
 *MHLRES7.NONSMSX.BKP               *
+0020  40404040 40404040 40404040 E3C8E2F0 F0F00000 78048083 00243404 0111001F
 *             THS000               *
+0040  40006D2F 00940000 000003E8 02916609 00005233 00012220 0000D4D3 F9F4F0C4
 *            Y               ML940D*
+0060  00000000 3030200F 00000000 0111001F 00000000 00000000 00000A47 40404040
```

Note: For migration data sets, the offset is +0004.

In other situations, you might need to know the data sets that were allocated in a specific primary volume at the time DFSMShsm started the last incremental backup on that volume. With the following command, you can retrieve the data set names on that volume, data set organization, creation date, referenced date, and information about whether the data set was changed. The following command lists the data sets on HG6600 when DFSMShsm performed the last incremental backup:

```
HSEND LIST PVOL(HG6600) BCDS BACKUPCONTENTS ODS(MHLRES5.DSN6)
```

The output from this command is similar to the output that is shown in Example 13-8.

Example 13-8 Output from the LIST PVOL command

```
- DFSMSHSM CONTROL DATASET -PRIMARY VOLUME-BCDS-- BCONTENTS    --- AT  16:42:54 ON
98/07/30 FOR SYSTEM=SC54
 CONTENTS OF BACKUP VTOC COPY # 00 FOR PRIMARY VOLUME HSM14A
DATASET NAME                                  ORG  MULTI  CREATED   REFERENCED
EXP DATE  RACF  PSWD   CHANGED
 HSM.BCDS                                      VS    ***   98/07/23  98/07/29
00/00/00  ***   ***    YES
 HSM.BCDS.DATA                                 VS    ***   98/07/23  98/07/29
00/00/00  ***   ***    YES
 HSM.BCDS.INDEX                                VS    ***   98/07/23  00/00/00
00/00/00  ***   ***    NO
 HSM.MCDS                                      VS    ***   98/07/23  98/07/29
00/00/00  ***   ***    YES
 HSM.MCDS.DATA                                 VS    ***   98/07/23  98/07/29
00/00/00  ***   ***    YES
 HSM.MCDS.INDEX                                VS    ***   98/07/23  00/00/00
00/00/00  ***   ***    NO
 HSM.OCDS                                      VS    ***   98/07/23  98/07/29
00/00/00  ***   ***    YES
 HSM.OCDS.DATA                                 VS    ***   98/07/23  98/07/29
00/00/00  ***   ***    YES
```

```
 HSM.OCDS.INDEX                                VS    ***    98/07/23  00/00/00
00/00/00  ***   ***    NO
 HSMACT.H1.BAKLOG.D98208.T182753               PS    NO     98/07/27  98/07/27
00/00/00  NO    NO     YES
 HSMACT.H1.BAKLOG.D98209.T105715               PS    NO     98/07/28  98/07/28
00/00/00  NO    NO     YES
 HSMACT.H1.BAKLOG.D98210.T154052               PS    NO     98/07/29  98/07/29
00/00/00  NO    NO     YES
 HSMACT.H1.BAKLOG.D98210.T165212               PS    NO     98/07/29  98/07/29
00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98204.T183225               PS    NO     98/07/23  98/07/23
00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98208.T182753               PS    NO     98/07/27  98/07/27
00/00/00  NO    NO     YES
 HSMACT.H1.CMDLOG.D98208.T200850               PS    NO     98/07/27  98/07/27    0
```

Example 13-8 on page 378 shows the data set names, DSORG, multivolume information, creation date, referenced date, and changed attributes among others.

Dumps

The **LIST** command has multiple options to list dump classes and backups. We show several common listings that are used on a daily basis.

The first step when you work with dumps is to generate a list of all dump classes that are defined to DFSMShsm, and its configuration. A simple way to list all of the dump classes that are defined to DFSMShsm is by issuing a **LIST DUMPCLASS** command. If you already know the dump class name, you can put the dump class name in parentheses, and DFSMShsm will display information about the specific dump class only. The following command lists all dump classes that are defined to DFSMShsm:

```
HSEND LIST DUMPCLASS ODS(MHLRES7.DSN1)
```

This command will produce an output similar to the output that is shown in Example 13-9.

Example 13-9 Sample output from LIST DUMPCLASS command

```
--- DFSMSHSM CONTROL DATASET -DUMP CLASS-BCDS--- LISTING --- AT 18:22:36 ON 12/0

DUMP      UNIT     AUTO   DATASET  RESET  CLASS   CP  FRR                    TAPE
VTOC
CLASS     TYPE     REUSE  RESTORE  CHANGE DISABLE REQ AVA DAY FREQ   RETPD
EXPDT  COPIES  STACK  DISPOSITION
CRYPCOPY  3590-1    YES    NO       NO      NO    NO  NO  **   000   000356
*******  000     030  ADDITIONAL TAPE COPY

          HWCOMP  ENCRYPT  ENCTYPE    ICOUNT  RSAKEY/KPWD
          NO      KEYPW    CLRAES128  00100   PASSWORD


DUMP      UNIT     AUTO   DATASET  RESET  CLASS   CP  FRR                    TAPE
VTOC
CLASS     TYPE     REUSE  RESTORE  CHANGE DISABLE REQ AVA DAY FREQ   RETPD
EXPDT  COPIES  STACK  DISPOSITION
DBOATST1  VT3590    YES    YES      NO      NO    **  YES **   000   000001
*******  002     060  ********************
```

```
          HWCOMP  ENCRYPT  ENCTYPE    ICOUNT  RSAKEY/KPWD
          NO      NONE     *********  *****
****************************************************************
```

The output shows the dump classes, unit type that is used, backup frequency, retention period, and stack among other attributes.

You can also create a list of all of the dump volumes that were created in your environment. Issue the following command to list all dump volumes that are recorded on BCDS:

```
HSEND LIST DVOL BCDS ODS('MHLRES7.DNS5')
```

The command output in Example 13-10 shows you extra information about the dumps, including the tape status (AVAIL, EXPIRED, or UNEXPIRED), source volsers, dump date and time, and target tape volser.

Example 13-10   HSEND LIST DVOL BCDS command output

```
-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 19:30:38 ON

DUMP   VOL    UNIT     FILE SOURCE           DUMPED    DUMPED
PCT  HW ENC C SET OF DUMP
VOLSER STATUS TYPE     SEQ  VOLSER SMS CLASS    DATE       TIME      EXP DATE IDRC
LIBRARY FULL        P VOLSERS
COB001 AVAIL  3590-1   ***  ****** *   DB9CTST1 ********** **:**:** *NOLIMIT   **
*NO LIB* **** ** **** *
THS012 EXPIR  3590-1                   SUNDAY                        2008/04/21 Y
LIB2     ***  *  *** N
                001  MLDE65 Y       2008/03/25 15:48:02
THS012
THS015 UNEXP  3590-1                   SUNDAY                        2012/09/23 Y
LIB2     ***  N  *** N
                001  MLDB35 Y       2012/08/27 18:10:29
THS015
THS019 EXPIR  3590-1                   DB0BTST1                      2012/08/30 Y
LIB2     ***  N  *** Y
                001  SB0XJ0 Y       2012/08/29 13:25:44
THS019
                002  SB0XJ1 Y       2012/08/29 13:25:44
THS019
                003  SB0XJ8 Y       2012/08/29 13:25:44
THS019
                004  SB0XJ9 Y       2012/08/29 13:25:44
THS019
```

Note: To list information about a specific volume, use the **HSEND LIST PRIMARY(DACMC1) BCDS ALLDUMPS** command instead.

If you need a list of all of the data sets that are stored in a specific volume when it was dumped, you can use the following command. You need to provide the dump volume name, and the source volume that you want to list. If you do not enter a source volume, DFSMShsm will list the volumes that are recorded on that tape. The following command lists the content of SB0XJ0 on the THS019 dump tape:

```
HSEND LIST DUMPVOLUME(THS019) DCONTENTS(SB0XJ0) ODS('MHLRES7.DSN1')
```

The previous **LIST DUMPVOLUME** command gives you information about the dump tape and the data sets on SB0XJ0. See Example 13-11.

Example 13-11 LIST DUMPVOLUME output

```
DUMP   VOL    UNIT     FILE SOURCE           DUMPED    DUMPED
PCT  HW ENC C SET OF DUMP
VOLSER STATUS TYPE     SEQ  VOLSER SMS CLASS    DATE       TIME      EXP DATE IDRC
LIBRARY FULL       P VOLSERS
THS019 EXPIR  3590-1                 DB0BTST1                        2012/08/30 Y
LIB2     ***  N  *** Y
                001  SB0XJ0 Y        2012/08/29 13:25:44
THS019

       ENCTYPE    RSAKEY/KPWD
       *********  ****************************************************************

DUMP COPY DATA SET NAME = HSM.DMP.DB0BTST1.VSB0XJ0.D12242.T442513
CONTENTS OF VTOC COPY FOR SOURCE VOLUME  SB0XJ0
DATASET NAME                                ORG  MULTI  CREATED   REFERENCED
EXP DATE  RACF  PSWD   CHANGED
DB0BA.ARCHLOG1.A0000001                     PS    NO    11/04/27  11/04/27
38/09/11  NO    NO     YES
DB0BA.ARCHLOG1.A0000002                     PS    NO    11/04/28  11/04/28
38/09/12  NO    NO     YES
```

Users

In addition to the other options, you can also obtain information about DFSMShsm users that are authorized by the DFSMShsm **AUTH** command.

Use the following command to quickly identify users that have or must not have DFSMShsm authorization. This command lists user access that is authorized by the **AUTH** command:

```
HSEND LIST USER ODS('MHLRES7.LISTUSER')
```

The output that is written to the data set looks like Example 13-12.

Example 13-12 Sample LIST USER command output

```
- DFSMSHSM CONTROL DATASET - USER-- LISTING ----- AT 20:12:31 ON 12/09/04 FOR SY

ARC1700I DFSMSHSM COMMANDS ARE RACF PROTECTED

 USERID   AUTH

BYRNEF    CNTL
HAIMO     CNTL
HGPARK    CNTL
JME001    CNTL
KOHJI     CNTL
```

In the command output, a number of users have an authority of “CNTL”. We recommend that you assign this authority only as required and that most users are assigned “USER” authority.

13.2 Query command

As DFSMShsm processes its units of work, it maintains information in its address space. As the **LIST** command obtains its information from the CDSs, another command will interrogate the address space. This command is the **QUERY** command, and it can be used to display real-time information about your DFSMShsm system. The **QUERY** command returns information that includes the following factors:

- The current SETSYS parameters
- The current ABARS parameters
- The status of outstanding DFSMShsm requests
- Volume space usage
- The status of each volume and data set subtask and long-running commands
- The progress of automatic functions
- The parameters that are used at DFSMShsm startup

Unlike the **LIST** command, the **QUERY** command asks the DFSMShsm address space for the requested information. Therefore, you cannot use the **QUERY** command to request information about CDS records and you cannot use the **LIST** command for address space inquires.

Before you issue any **QUERY** command, you must ensure that you have the necessary RACF or DFSMShsm AUTH authorization to perform the queries. If you do not have the authority, or do not know how to check it, see Chapter 5, "Implementing DFSMShsm security" on page 79.

You can issue **QUERY** commands from the console or TSO session, code REXX programs, or add the commands to the DFSMShsm startup procedure. Part of the displayed information might be incorrect because DFSMShsm is not yet fully initialized.

DFSMShsm

When you work with **QUERY** commands, if you issue the **HSEND** command, the output is directed back to your TSO terminal and to the DFSMShsm log. If you issue the **MODIFY** command through the console, the information is returned to the system log.

The following example shows the **QUERY** command to display DFSMShsm active tasks:

```
HSEND QUERY ACTIVE
```

This command displays DFSMShsm active tasks for the specified host. Other hosts might show different results because they might be performing other functions or have different settings.

Example 13-13 shows an output from the **HSEND QUERY ACTIVE** command. You can use this command to check what DFSMShsm is processing, and whether any tasks are on hold.

Example 13-13 Sample output from HSEND QUERY ACTIVE command

```
ARC0101I QUERY ACTIVE COMMAND STARTING ON HOST=2
 ARC0144I AUDIT=NOT HELD AND INACTIVE, LIST=NOT HELD AND INACTIVE, RECYCLE=NOT
 ARC0144I (CONT.) HELD AND INACTIVE, REPORT=NOT HELD AND INACTIVE
 ARC0160I MIGRATION=NOT HELD, AUTOMIGRATION=NOT HELD, RECALL=NOT HELD,
 ARC0160I (CONT.) TAPERECALL=NOT HELD, DATA SET MIGRATION=INACTIVE, VOLUME
 ARC0160I (CONT.) MIGRATION=INACTIVE, DATA SET RECALL=INACTIVE
 ARC0163I BACKUP=NOT HELD, AUTOBACKUP=NOT HELD, RECOVERY=NOT HELD,
 ARC0163I (CONT.) TAPEDATASETRECOVERY=NOT HELD, DATA SET BACKUP=NOT HELD, VOLUME
 ARC0163I (CONT.) BACKUP=INACTIVE, DATA SET RECOVERY=INACTIVE, VOLUME
 ARC0163I (CONT.) RECOVERY=INACTIVE
 ARC0276I DATA SET BACKUP=INACTIVE, DATA SET BACKUP ACTUAL IDLETASKS=(ALLOC=00,
```

```
ARC0276I (CONT.) MAX=00)
ARC1826I FRBACKUP=NOT HELD AND INACTIVE,FRRECOV=NOT HELD AND INACTIVE,FRBACKUP
ARC1826I(CONT.)DUMP=NOTHELDANDINACTIVE,FRRECOV(TAPE)=NOTHELDANDINACTIVE,
ARC1826I (CONT.) FRRECOV(DATASET)=NOT HELD AND INACTIVE
ARC0642I DUMP=NOT HELD, AUTODUMP=NOT HELD, VOLUME DUMP=INACTIVE, VOLUME
ARC0642I (CONT.) RESTORE=INACTIVE, DATA SET RESTORE=INACTIVE
ARC0437I - TAPECOPY NOT HELD AND INACTIVE
ARC0437I - TAPEREPL NOT HELD AND INACTIVE
ARC0415I EXPIREBV=NOT HELD AND INACTIVE, LAST STORED BACKUP VERSION KEY=, LAST
ARC0415I (CONT.) STORED ABARS VERSION KEY=, LAST PLANNED END KEY=
ARC0460I PRIVATE AREA LIMIT=8168K, UNALLOCATED=5428K, LARGEST FREE AREAS=5412K,
ARC0460I (CONT.)  16K
ARC0460I EXTENDED PRIVATE AREA LIMIT=1460M, UNALLOCATED=1429M, LARGEST FREE
ARC0460I (CONT.) AREAS=1428M, 236K
ARC6018I AGGREGATE BACKUP/RECOVERY = INACTIVE
ARC6019I AGGREGATE BACKUP = NOT HELD, AGGREGATE RECOVERY = NOT HELD
ARC1540I COMMON RECALL QUEUE PLACEMENT FACTORS: CONNECTION STATUS=CONNECTED,
ARC1540I (CONT.) CRQPLEX HOLD STATUS=NONE,HOST COMMONQUEUE HOLD STATUS=NONE,
ARC1540I (CONT.) STRUCTURE ENTRIES=000% FULL,STRUCTURE ELEMENTS=000% FULL
ARC1541I COMMON RECALL QUEUE SELECTION FACTORS: CONNECTION STATUS=CONNECTED,
ARC1541I (CONT.) HOST RECALL HOLD STATUS=NONE,HOST COMMONQUEUE HOLD STATUS=NONE
ARC0101I QUERY ACTIVE COMMAND COMPLETED ON HOST=2
```

We recommend that you use the **MVS MODIFY** command for queries that might be long, so you can scroll up and down, and perform searches to find the information that you want in the log. The **MVS MODIFY** command syntax is **/F,*hsmtask*,QUERY ACTIVE**. The **HSEND** command is not necessary when you execute **MVS MODIFY** commands.

If multiple hosts are in your z/OS image, you can issue the command to display information that is necessary to identify each host in the system for future commands. Use this command to query DFSMShsm host images:

```
HSEND Q IMAGE
```

The information returns with the message that is shown in Example 13-14.

Example 13-14 Result from HSEND Q IMAGE command

```
ARC0101I QUERY IMAGE COMMAND STARTING ON HOST=1
 ARC0250I HOST PROCNAME JOBID ASID MODE
 ARC0250I    1     HSM1 02980 0049 MAIN
 ARC0250I    2     HSM2 02981 004A  AUX
 ARC0101I QUERY IMAGE COMMAND COMPLETED ON HOST=1
```

You can also list all DFSMShsm control parameters that are set by using a **LIST** command. It will give you a list of ABARS, backup, migration, dump, and other settings, so you can verify that all parameters are set correctly. Use the following command to display all DFSMShsm control settings:

```
HSEND Q SETSYS
```

The output from the previous command is large. We recommend that you issue this command by using **MVS MODIFY**, so you can page up and down, or search on the command output.

For a list of the parameters that were specified on the PROC statement in the DFSMShsm startup procedure, issue the following command:

```
HSEND Q STAR
```

The information is returned in the messages that are shown in Example 13-15.

Example 13-15 Output from the HSEND Q STAR command

```
ARC0101I QUERY STARTUP COMMAND STARTING ON HOST=2
ARC0143I PARMLIB MEMBER=ARCCMD64, DFSMSHSM AUTHORIZED USERID=HSM, HOSTID=2,
ARC0143I (CONT.) PRIMARY HOST=NO, LOGSW=NO, STARTUP=NO, EMERGENCY=NO, CDSQ=NO,
ARC0143I (CONT.) CDSR=NO, PDA=YES, RESTART=NOT SPECIFIED, CDSSHR=RLS,
ARC0143I (CONT.) RNAMEDSN=NO, STARTUP PARMLIB MEMBER=ARCSTR00
ARC0249I CELLS=(200,100,100,50,20),HOSTMODE=MAIN
ARC0101I QUERY STARTUP COMMAND COMPLETED ON HOST=2
```

In certain cases, you will need information about daily statistics for DFSMShsm processing. DFSMShsm tracks activity on a daily basis in the daily statistics record (DSR).

To display the DFSMShsm daily statistics information at any time throughout the day, issue the following command:

```
HSEND Q STAT
```

DFSMShsm returns a summary of all its processing for the current day. The output includes CPU time, and data sets that were migrated. See Example 13-16.

Example 13-16 Statistics from the QUERY STAT command

```
ARC0101I QUERY STATISTICS COMMAND STARTING ON HOST=2
ARC0155I DFSMSHSM STATISTICS FOR 12/09/05
ARC0156I STARTUPS=01, SHUTDOWNS=00, ABENDS=00, MWES=0038, CPU TIME=00001.14
ARC0156I (CONT.) SECONDS
ARC0157I DS MIGRATED L1=00000001, DS MIGRATED L2=00000000, DS EXTENT
ARC0157I (CONT.) REDUCTIONS=000000, DS MIGRATED FAIL=001, TRKS
ARC0157I (CONT.) MIGRATED=00000380,   BYTES MIGRATED=000080596
ARC0158I DS RECALLED L1=00000000, DS RECALLED L2=00000000, DS RECALL FAIL=000,
ARC0158I (CONT.)   BYTES RECALLED=000000000, RECALL MOUNTS AVOIDED=00000, EXTRA
ARC0158I (CONT.) ABACKUP MOUNTS=00000
ARC0159I DS BACKUP=00000000, DS BACKUP FAIL=000, DS RECOVER=00000000, DS
ARC0159I (CONT.) RECOVER FAIL=000, RECOVER MOUNTS AVOIDED=00000
ARC0641I VOL DUMP=0, VOL DUMP FAIL=0, VOL RESTORE=0, VOL RESTORE FAIL=0, DS
ARC0641I (CONT.) RESTORE=0, DS RESTORE FAIL=0
ARC0145I DS DELETED=00000000, DS DELETE FAILED=000
ARC0146I RECYCLED BACKUP VOLUMES=0000, DS=00000000, BLOCKS=000000
ARC0146I RECYCLED MIGRATION VOLUMES=0000, DS=00000000, BLOCKS=000000
ARC1825I FAST REPLICATION VOLUME BACKUPS=0 REQUESTED, 0 FAILED; VOLUME
ARC1825I (CONT.) RECOVERIES=0 REQUESTED, 0 FAILED
ARC0101I QUERY STATISTICS COMMAND COMPLETED ON HOST=2
```

DFSMShsm automatically records certain errors that it receives and records other errors by using the diagnosis command, **TRAP**. To see which TRAP activity occurred, issue the following command:

```
HSEND Q T
```

The information that is returned in the messages is shown in Example 13-17.

Example 13-17 Output from the HSEND Q TRAP command

```
ARC0101I QUERY TRAPS COMMAND STARTING ON HOST=2
ARC0205I TRAP IN MODULE ARCLBUC FOR CODE 00404, TIMES=0001, TYPE=BY OCCURRENCE
ARC0205I TRAP IN MODULE ARCALVOL FOR CODE 00004, TIMES=0002, TYPE=BY OCCURRENCE
ARC0101I QUERY TRAPS COMMAND COMPLETED ON HOST=2
```

Aggregate backup and recovery support

If you want to query the current DFSMShsm control parameters that apply to aggregate backup and recovery (ABARS), issue the following command:

```
HSEND QUERY ABARS
```

The information is returned in the messages that are shown in Example 13-18.

Example 13-18 Sample return from the QUERY ABARS command

```
ARC0101I QUERY ABARS COMMAND STARTING ON HOST=2
ARC6008I AGGREGATE BACKUP/RECOVERY PROCNAME = DFHSMABR
ARC6009I AGGREGATE BACKUP/RECOVERY MAXADDRESSSPACE = 01
ARC6366I AGGREGATE BACKUP/RECOVERY UNIT NAME = 3490
ARC6367I AGGREGATE BACKUP/RECOVERY EXITS = NONE
ARC6368I AGGREGATE BACKUP/RECOVERY ACTIVITY LOG MESSAGE LEVEL IS FULL
ARC6371I AGGREGATE RECOVERY ML2 TAPE UNIT NAME = 3490
ARC6372I NUMBER OF ABARS I/O BUFFERS = 02
ARC6373I ABARS ACTIVITY LOG OUTPUT TYPE = DASD
ARC6033I AGGREGATE RECOVERY UNIT NAME = 3490
ARC6036I AGGREGATE BACKUP OPTIMIZE = 3
ARC6036I AGGREGATE RECOVERY TGTGDS = SOURCE
ARC6036I AGGREGATE RECOVERY ABARSVOLCOUNT = ANY
ARC6036I AGGREGATE RECOVERY PERCENTUTILIZED = 090
ARC6036I AGGREGATE BACKUP/RECOVERY ABARSDELETEACTIVITY = NO
ARC6036I AGGREGATE BACKUP/RECOVERY ABARSTAPES = STACK
ARC6036I AGGREGATE BACKUP ABARSKIP = NOPPRC, NOXRC
ARC0101I QUERY ABARS COMMAND COMPLETED ON HOST=2
```

You can also query automatic functions that are being performed by DFSMShsm to retrieve information about the processed volumes, and volumes to be processed by the automatic function. This information might be helpful for you to determine how long it will take to finish the current processing.

The following command shows you how to query the DFSMShsm automatic functions:

```
HSEND QUERY AUTOP
```

The information is returned in the messages that are shown in Example 13-19.

Example 13-19 Output from previous example

```
ARC0101I QUERY AUTOPROGRESS COMMAND STARTING ON HOST=1
ARC0247I PRIMARY SPACE MANAGEMENT IS CURRENTLY
ARC0247I (CONT.) PROCESSING DFSMSHSM MANAGED VOLUMES
ARC0246I SMS VOLUMES RESTRICTED TO PROCESSING BY THIS
ARC0246I (CONT.) PROCESSING UNIT: NOT PROCESSED=0, TOTAL=0, SMS
ARC0246I (CONT.) VOLUMES NOT RESTRICTED TO PROCESSING BY ANY
ARC0246I (CONT.) PROCESSING UNIT: NOT PROCESSED=0, TOTAL=4, NON-SMS
```

```
ARC0246I (CONT.) VOLUMES: NOT PROCESSED=0, TOTAL=0
ARC0101I QUERY AUTOPROGRESS COMMAND COMPLETED ON HOST=1
```

The same command gives the result that is shown in Example 13-20 when automatic functions are not in progress.

Example 13-20 Output from previous command if no automatic function is running

```
ARC0101I QUERY AUTOPROGRESS COMMAND STARTING ON HOST=3
ARC0247I NO AUTOMATIC FUNCTION IS CURRENTLY PROCESSING DFSMSHSM MANAGED VOLUMES
ARC0246I ODM VOLUMES RESTRICTED TO PROCESSING BY THIS PROCESSING UNIT: NOT
ARC0246I (CONT.) PROCESSED=0, TOTAL=0, ODM VOLUMES NOT RESTRICTED TO PROCESSING
ARC0246I (CONT.) BY ANY PROCESSING UNIT: NOT PROCESSED=0, TOTAL=0, NON-SMS
ARC0246I (CONT.) VOLUMES: NOT PROCESSED=0, TOTAL=0
ARC0101I QUERY AUTOPROGRESS COMMAND COMPLETED ON HOST=3
```

Backup and migration

Querying backup is necessary to identify possible tasks on hold, or to check and decide how many tasks need to run concurrently for each function to avoid over-allocating tape drives and avoid wasting tapes during backup processing.

To list information about your backup configuration, use the **BACK** parameter, which will give you all of the DFSMShsm backup configurations that are active. Use the following command to display the backup configuration:

```
HSEND Q BACK
```

The previous command will display output that is similar to Example 13-21.

Example 13-21 Sample output from the HSEND Q BACK command

```
ARC0101I QUERY BACKUP COMMAND STARTING ON HOST=3
ARC0638I MAXDUMPTASKS=03, ADSTART=(0000 0000 0000), DUMPIO=(3,2),
ARC0638I (CONT.) VOLUMEDUMP=(STANDARD), MAXDUMPRECOVERTASKS=01
ARC0273I DUMP CYCLE LENGTH=7 DAY(S), CYCLE=NNNNNNY, TODAY IS DAY=3, CYCLE
ARC0273I (CONT.) START DATE=98/03/02, AUTODUMP/LEVEL FUNCTIONS NOT ELIGIBLE TO
ARC0273I (CONT.) BE STARTED, CYCLE START TIME NOT SPECIFIED
ARC0274I BACKUP=YES(TAPE(VT3590G2)), SPILL=YES(ANY), MAXDSRECOVERTASKS=03,
ARC0274I (CONT.) MAXDSTAPERECOVERTASKS=03
ARC0154I MAXBACKUPTASKS=03, ABSTART= (0000 0000 0000), VERSIONS=001,
ARC0154I (CONT.) FREQUENCY=000, SKIPABPRIMARY=NO, BACKUP PREFIX=HSM,
ARC0154I (CONT.) INCREMENTALBACKUP=CHANGEDONLY, PROFILEBACKUP=YES,
ARC0154I (CONT.) INUSE=(RETRY=NO, DELAY=015, SERIALIZATION=REQUIRED)
ARC0269I DS DASD BACKUP TASKS=02, DS TAPE BACKUP TASKS=02, DEMOUNTDELAY=0060,
ARC0269I (CONT.) MAXIDLETASKS=00, DS BACKUP MAX DASD SIZE=000003000, DS BACKUP
ARC0269I (CONT.) STD DASD SIZE=000003000, SWITCHTAPES TIME=0000,
ARC0269I (CONT.) PARTIALTAPE=MARKFULL, GENVSAMCOMPNAMES=YES
ARC0271I BACKUP CYCLE LENGTH=01 DAY(S), CYCLE=Y, TODAY IS DAY=01, VOLUME
ARC0271I (CONT.) LIMIT/DAY=0001, AVAILABLE BACKUP VOLUMES=00008, CYCLE START
ARC0271I (CONT.) DATE=98/03/02
ARC0101I QUERY BACKUP COMMAND COMPLETED ON HOST=3
```

You can also use the words ALL, DAILY, SPILL, and UNASSIGNED between parenthesis to display information that relates to specific backup processing. The following command displays information about spill backup processing:

```
HSEND Q BACK(SPILL)
```

Part of the information that is displayed in this command output is also available from the general **HSEND Q BACK** command. Also, ARC0164I is added to indicate the backup volumes. See Example 13-22.

Example 13-22 Output from the QUERY BACK(SPILL) command

```
ARC0101I QUERY BACKUP COMMAND STARTING ON HOST=3
ARC0638I MAXDUMPTASKS=03, ADSTART=(0000 0000 0000), DUMPIO=(3,2),
ARC0638I (CONT.) VOLUMEDUMP=(STANDARD), MAXDUMPRECOVERTASKS=01
ARC0273I DUMP CYCLE LENGTH=7 DAY(S), CYCLE=NNNNNNY, TODAY IS DAY=3, CYCLE
ARC0273I (CONT.) START DATE=98/03/02, AUTODUMP/LEVEL FUNCTIONS NOT ELIGIBLE TO
ARC0273I (CONT.) BE STARTED, CYCLE START TIME NOT SPECIFIED
ARC0274I BACKUP=YES(TAPE(VT3590G2)), SPILL=YES(ANY), MAXDSRECOVERTASKS=03,
ARC0274I (CONT.) MAXDSTAPERECOVERTASKS=03
ARC0154I MAXBACKUPTASKS=03, ABSTART= (0000 0000 0000), VERSIONS=001,
ARC0154I (CONT.) FREQUENCY=000, SKIPABPRIMARY=NO, BACKUP PREFIX=HSM,
ARC0154I (CONT.) INCREMENTALBACKUP=CHANGEDONLY, PROFILEBACKUP=YES,
ARC0154I (CONT.) INUSE=(RETRY=NO, DELAY=015, SERIALIZATION=REQUIRED)
ARC0269I DS DASD BACKUP TASKS=02, DS TAPE BACKUP TASKS=02, DEMOUNTDELAY=0060,
ARC0269I (CONT.) MAXIDLETASKS=00, DS BACKUP MAX DASD SIZE=000003000, DS BACKUP
ARC0269I (CONT.) STD DASD SIZE=000003000, SWITCHTAPES TIME=0000,
ARC0269I (CONT.) PARTIALTAPE=MARKFULL, GENVSAMCOMPNAMES=YES
ARC0271I BACKUP CYCLE LENGTH=01 DAY(S), CYCLE=Y, TODAY IS DAY=01, VOLUME
ARC0271I (CONT.) LIMIT/DAY=0001, AVAILABLE BACKUP VOLUMES=00008, CYCLE START
ARC0271I (CONT.) DATE=98/03/02
ARC0164I SPILL      VOLS = VT0006-A VT0005-A THS016-A
ARC0101I QUERY BACKUP COMMAND COMPLETED ON HOST=3
```

If you discover that certain data sets are not migrated correctly, you can issue the following command. To display all data sets that are being prevented from migrating because a **SETMIG LEVEL(*hlq*)** command was issued, use the following command to query the migration restrictions on DFSMShsm:

```
HSEND Q RET
```

DFSMShsm returns all data sets that are prevented from migrating by the **SETMIG LEVEL(*hlq*)** command. See Example 13-23.

Example 13-23 Sample output from the HSEND Q RET command

```
ARC0101I QUERY RETAIN COMMAND STARTING ON HOST=2
ARC0175I LEVEL QUALIFIER AND MIGRATION RESTRICTION TYPE
ARC0176I QUALIFIER=SYS1. RESTRICTION TYPE=NOMIGRATION
ARC0176I QUALIFIER=SYSCTLG. RESTRICTION TYPE=NOMIGRATION
ARC0176I QUALIFIER=HSM. RESTRICTION TYPE=NOMIGRATION
ARC0101I QUERY RETAIN COMMAND COMPLETED ON HOST=2
```

The command to list the space usage of volumes can be applied to all your volumes or a specific volser that is coded. Use the following command to list the space information for the volume SBXH56:

```
HSEND Q SPACE(SBXHS6)
```

DFSMShsm includes information about the free space, fragmentation, largest extents, and data set control block (DSCB) usage. See Example 13-24.

Example 13-24 Sample output from the HSEND Q SPACE command

```
ARC0400I VOLUME SBXHS6 IS 99% FREE, 0000000110 FREE TRACK(S), 000009972 FREE
ARC0400I (CONT.) CYLINDER(S), FRAG .001
ARC0401I LARGEST EXTENTS FOR SBXHS6 ARE CYLINDERS      9952, TRACKS     149280
ARC0402I VTOC FOR SBXHS6 IS 00090 TRACKS(0004500 DSCBS), 0004464 FREE
ARC0402I (CONT.) DSCBS(99% OF TOTAL)
```

Note: If we omitted the volser, DFSMShsm returns this information for all non-storage management subsystem (SMS) primary and migration-level 1 (ML1) volumes. Additionally, no ARC0401I message is issued.

Control data sets

Also, use the **QUERY** command to check CDS and journal usage, so you can schedule changes to reorganize and increase CDS journals ahead of time, issue **BACKVOL CDS** commands to the null journal data set, and avoid journal-full issues. Use this command to display CDS information:

```
HSEND Q CDS
```

The output in Example 13-25 shows information about all CDS usage, space allocation, and the threshold that is used for warning messages.

Example 13-25 Sample output from the HSEND Q CDS command

```
ARC0101I QUERY CONTROLDATASETS COMMAND STARTING ON HOST=2
ARC0947I CDS SERIALIZATION TECHNIQUE IS RLS
ARC0148I MCDS TOTAL SPACE=72000 K-BYTES, CURRENTLY ABOUT 28% FULL, WARNING
ARC0148I (CONT.) THRESHOLD=80%, TOTAL FREESPACE=94%, EA=YES, CANDIDATE
ARC0148I (CONT.) VOLUMES=4
ARC0948I MCDS INDEX TOTAL SPACE=210 K-BYTES, CURRENTLY ABOUT 27% FULL, WARNING
ARC0948I (CONT.) THRESHOLD=80%, CANDIDATE VOLUMES=4
ARC0148I BCDS TOTAL SPACE=72000 K-BYTES, CURRENTLY ABOUT 48% FULL, WARNING
ARC0148I (CONT.) THRESHOLD=80%, TOTAL FREESPACE=95%, EA=YES, CANDIDATE
ARC0148I (CONT.) VOLUMES=4
ARC0948I BCDS INDEX TOTAL SPACE=210 K-BYTES, CURRENTLY ABOUT 46% FULL, WARNING
ARC0948I (CONT.) THRESHOLD=80%, CANDIDATE VOLUMES=4
ARC0148I OCDS TOTAL SPACE=72000 K-BYTES, CURRENTLY ABOUT 34% FULL, WARNING
ARC0148I (CONT.) THRESHOLD=80%, TOTAL FREESPACE=96%, EA=YES, CANDIDATE
ARC0148I (CONT.) VOLUMES=4
ARC0948I OCDS INDEX TOTAL SPACE=210 K-BYTES, CURRENTLY ABOUT 33% FULL, WARNING
ARC0948I (CONT.) THRESHOLD=80%, CANDIDATE VOLUMES=4
ARC0148I JOURNAL TOTAL SPACE=3875053 K-BYTES, CURRENTLY ABOUT 000% FULL,
ARC0148I (CONT.) WARNING THRESHOLD=080%, TOTAL FREESPACE=100%, EA=NO, CANDIDATE
ARC0148I (CONT.) VOLUMES=0
ARC0101I QUERY CONTROLDATASETS COMMAND COMPLETED ON HOST=2
```

You can also display information about CDS backup data sets. The following command shows how to use the **CDSV** command to query CDS backup data sets:

```
HSEND Q CDSV
```

The **HSEND Q CDSV** command output includes backup data set names, the latest valid backup version (LATESTFINALQUALIFIER), and the data mover that was used for backup. See Example 13-26.

Example 13-26 Sample output from the QUERY CDSV command

```
ARC0101I QUERY CDSVERSIONBACKUP COMMAND STARTING ON HOST=2
ARC0375I CDSVERSIONBACKUP, MCDSBACKUPDSN=HSM.MCDS.BACKUP,
ARC0375I (CONT.) BCDSBACKUPDSN=HSM.BCDS.BACKUP, OCDSBACKUPDSN=HSM.OCDS.BACKUP,
ARC0375I (CONT.) JRNLBACKUPDSN=HSM.JRNL.BACKUP
ARC0376I BACKUPCOPIES=0004, BACKUPDEVICECATEGORY=DASD,
ARC0376I (CONT.) LATESTFINALQUALIFIER=D0000040, DATAMOVER=DSS
ARC0101I QUERY CDSVERSIONBACKUP COMMAND COMPLETED ON HOST=2
```

Requests

Displaying requests is the easiest way to identify the type of requests that are waiting on the queue. You can increase the number of tasks that are running concurrently if enough resources are available, identify users that are sending multiple requests, and cancel or give a request a higher priority, depending on the user's need and the available resources.

You can use many options with **QUERY** commands. The first command displays a summary of all requests that are being processed, or are waiting to be processed by DFSMShsm:

```
HSEND Q WAIT
```

DFSMShsm will sum all migration, backup, recall, and other requests, and display the results in an easy-to-use format. See Example 13-27.

Example 13-27 Sample output from the HSEND Q WAIT command

```
ARC0101I QUERY WAITING COMMAND STARTING ON HOST=2
ARC1542I WAITING MWES ON COMMON QUEUES: COMMON RECALL QUEUE=00000000,
ARC1542I (CONT.) TOTAL=00000000
ARC0168I WAITING MWES: MIGRATE=00000000, RECALL=00000000, DELETE=00000000,
ARC0168I (CONT.) BACKUP=00000001, RECOVER=00000000, COMMAND=00000000,
ARC0168I (CONT.) ABACKUP=00000000, ARECOVER=00000000, FRBACKUP=00000000,
ARC0168I (CONT.) FRRECOV=00000000, TOTAL=00000001
ARC0101I QUERY WAITING COMMAND COMPLETED ON HOST=2
```

To produce a detailed list of all requests on the DFSMShsm queue, you can issue the following command:

```
HSEND Q REQ
```

The output from the previous command is a list of all requests that re being processed, or waiting to be processed by DFSMShsm. See Example 13-28.

Example 13-28 Output from the previous QUERY command

```
ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=2
ARC0167I BACKUP MWE FOR DATA SET MHLRES7.LISTUSER FOR USER MHLRES7, REQUEST
ARC0167I (CONT.) 00000145, WAITING TO BE PROCESSED, 00000 MWE(S) AHEAD OF THIS
ARC0167I (CONT.) ONE
ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=2
```

Note: The previous command can result in large output if many requests exist for DFSMShsm.

You can also query for specific users to determine the requests that were sent for processing by DFSMShsm by a single user. The following command is used to display requests from a specific user:

```
HSEND Q USER(user)
```

In addition, you can query for a data set name to retrieve information about this specific request. The following command queries for information about requests that involve a specific data set:

```
HSEND Q DATASETNAME('dsname')
```

The output includes the user who requested the action, and the number of requests that are ahead for processing. See Example 13-29.

Example 13-29 Sample output from the Q DATASETNAME command

```
ARC0101I QUERY DATASETNAME COMMAND STARTING ON HOST=2
ARC0167I BACKUP MWE FOR DATA SET MHLRES7.LISTUSER FOR USER MHLRES7, REQUEST
ARC0167I (CONT.) 00000155, WAITING TO BE PROCESSED, 00000 MWE(S) AHEAD OF THIS
ARC0167I (CONT.) ONE
ARC0101I QUERY DATASETNAME COMMAND COMPLETED ON HOST=2
```

13.3 Report command

The **REPORT** command is used to gather and consolidate statistics about DFSMShsm operations and functions. DFSMShsm during its processing stores daily function and volume-related information in the MCDS. Use the **REPORT** command to gather information that is generated at various levels:

- Gather the following information at a function level:
  - Backup
  - Migration
  - Delete
  - Recall
  - Recover
  - Recycle
  - Spill
- At a daily level
- At a volume level
- For statistics before or after a certain date
- For statistics between certain dates
- For a summary of all statistics reports

We show several **REPORT** commands that you can use to give you an idea of the type of information that is available. We also introduce utilities in SYS1.SAMPLIB(ARCTOOLS).

Before you issue any **REPORT** command, you must ensure that you have the necessary RACF or DFSMShsm AUTH authorization to perform the queries. If you do not have the authority, or do not know how to check it, see Chapter 5, “Implementing DFSMShsm security” on page 79.

You also must specify DFSMShsm for how long you want to keep statistics information. The following command sets this retention on DFSMShsm and can be added to DFSMShsm PARMLIB:

```
HSEND SETSYS MIGRATIONCLEANUPDAYS(recall stats reconnect)
```

The preceding command has three parameters that you can use. The **recall** parameter tells DFSMShsm how long to keep MCDS information about a recalled data set. The **stats** parameter defines the number of days that DFSMShsm will keep statistics information. The **reconnect** parameter sets the number of days that DFSMShsm holds MCDS information for recalled data sets that are candidates for reconnection. The defaults are 10 (for *recall*), 30 (for *stats*), and 3 (for *reconnect*).

As in the **LIST** command, you can direct the **REPORT** command output to different places, such as a terminal, SYSOUT, or data set, by using the **ODS** parameter. If the data set exists, the REPORT output will be appended at the end of the data set. Otherwise, DFSMShsm will create the data set.

To obtain a summary of all function-related activity that occurred today, issue the following command:

```
HSEND REPORT DAILY FUNCTION ODS(MHLRES5.DAILY.STATS)
```

The output that is written to the data set looks similar to the output that is shown in Example 13-30.

Example 13-30 Output from the REPORT DAILY FUNCTION command

```
--DFSMSHSM STATISTICS REPORT ------- AT 16:55:21 ON 2012/09/05 FOR SYSTEM=SC64

DAILY STATISTICS REPORT FOR 12/09/05

STARTUPS=001, SHUTDOWNS=000, ABENDS=000, WORK ELEMENTS PROCESSED=000055, BKUP VOL
RECYCLED=00000, MIG VOL RECYCLED=00000
DATA SET MIGRATIONS BY VOLUME REQUEST= 0000001, DATA SET MIGRATIONS BY DATA SET
REQUEST= 00000, BACKUP REQUESTS= 0000000
EXTENT REDUCTIONS= 0000000 RECALL MOUNTS AVOIDED= 00000 RECOVER MOUNTS AVOIDED=
00000
FULL VOLUME DUMPS= 000000 REQUESTED, 00000 FAILED; DUMP COPIES= 000000 REQUESTED,
00000 FAILED
FULL VOLUME RESTORES= 000000 REQUESTED, 00000 FAILED; DATASET RESTORES= 000000
REQUESTED, 00000 FAILED
ABACKUPS= 00000 REQUESTED,00000 FAILED; EXTRA ABACKUP MOUNTS=00000
DATA SET MIGRATIONS BY RECONNECTION =  000000, NUMBER OF TRACKS RECONNECTED TO
TAPE = 00000000
FAST REPLICATION VOLUME BACKUP = 00000000 REQUESTED, 00000000 FAILED
FAST REPLICATION VOLUME RECOVER = 00000000 REQUESTED, 00000000 FAILED

                    NUMBER  ------READ--------  -----WRITTEN------
------REQUESTS---- AVERAGE ------AVERAGE TIME-------
   HSM FUNCTION    DATASETS TRK/BLK    BYTES    TRK/BLK    BYTES    SYSTEM USER
FAILED   AGE   QUEUED WAIT PROCESS TOTAL

MIGRATION
 PRIMARY - LEVEL 1 0000001  00000380 000080596  00000001 000032768  000001 00000
00000  00001  0000  00000  00000  00000
 SUBSEQUENT MIGS   0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
```

```
 PRIMARY - LEVEL 2 0000000  00000000 000000000  00000000 000000000  000000 00001
00001  00000  0000  00000  00000  00000
RECALL
 LEVEL 1 - PRIMARY 0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
 LEVEL 2 - PRIMARY 0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
DELETE
 MIGRATE DATA SETS 0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
 PRIMARY DATA SETS 0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
BACKUP
 DAILY BACKUP      0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
 SUBSEQUENT BACKUP 0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
 DELETE BACKUPS    0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
RECOVER
 BACKUP - PRIMARY  0000000  00000000 000000000  00000000 000000000  000000 00000
00000  00000  0000  00000  00000  00000
RECYCLE
 BACKUP - SPILL    0000000  00000000            00000000            000000 00000
00000  00000  0000  00000  00000  00000
 MIG L2 - MIG L2   0000000  00000000            00000000            000000 00000
00000  00000  0000  00000  00000  00000
```

The header of the REPORT output contains the number of mounts that were saved by RECALL and RECOVER. This information is useful for addressing the changes in mounts when you use larger capacity tapes.

It is also possible to define specific date periods that you want the reports, on the condition that they are stored by DFSMShsm. To specify a period for the report, you can issue the following command:

```
HSEND REPORT DAILY FROMDATE(12/08/12) FUNCTION TODATE(12/09/05)
```

The output is similar to the DSR. The difference is that you now get information at a function level:

- A function summary for the period 24 July to 31 July inclusive
- A function summary report that totals all activity for the requested period

You can use the **FUNCTION** parameters to select only the necessary information you need, for example:

- BACKUP
- DELETE
- MIGRATION
- RECALL
- RECOVER
- RECYCLE
- SPILL

If you issue the following command, you obtain a report that details the data sets that are backed up to daily volumes:

```
HSEND REPORT DAILY FUNCTION(BACKUP) ODS(MHLRES5.BACKUP.REPORT)
```

The output is similar to the output that was produced previously but restricted to the backup function. Certain statistics are always reported, and these statistics will appear before the requested function report.

It is also possible to create reports for all DFSMShsm-managed volumes, or specific volumes. DFSMShsm will retrieve information about backup, migration, recover, and recycle. The following command lists statistics for DFSMShsm-managed volumes:

```
HSEND REPORT VOLUMES FUNCTION ODS('MHLRES7.VOLUME.STATS')
```

Example 13-31 shows us the output from the previous command. If you intend to display data about a single volume, put the *volser* between parentheses after the **VOLUMES** parameter.

Example 13-31 Output from the REPORT VOLUMES command

```
UNIT TYPE = 3390    , HSM VOLUME TYPE = LEVEL 1
MIGRATED DATA SETS BY VOLUME REQUEST=000001, DATA SET MIGRATIONS BY DATA SET REQ
         MINIMUM AGE = 000, TOTAL TRACKS = 00003243, FREE TRACKS = 00147012, FRA
VOLUME DUMP=  NOT DONE;  DUMP COPIES=000000  REQUESTED, FAILED=000000
VOLUME RESTORE=  NOT DONE;  DATASET RESTORES=000000  REQUESTED, FAILED=000000
DATA SET MIGRATIONS BY RECONNECTION = 000000 , NUMBER OF TRACKS RECONNECTED TO T
FAST REPLICATION BACKUP = 00000 REQUESTED, 00000 FAILED
FAST REPLICATION RECOVER = 00000 REQUESTED, 00000 FAILED

                    NUMBER  ------READ--------  -----WRITTEN------  ------REQUES
   HSM FUNCTION    DATASETS TRK/BLK    K-BYTES  TRK/BLK    K-BYTES  SYSTEM USER

MIGRATION
 PRIMARY - LEVEL 1 0000001  00000000 000000000  00000001 000000032  000001 00000
 SUBSEQUENT MIGS   0000000  00000000 000000000  00000000 000000000  000000 00000
 PRIMARY - LEVEL 2 0000000  00000000 000000000  00000000 000000000  000000 00000
RECALL
 LEVEL 1 - PRIMARY 0000000  00000000 000000000  00000000 000000000  000000 00000
 LEVEL 2 - PRIMARY 0000000  00000000 000000000  00000000 000000000  000000 00000
DELETE
 MIGRATE DATA SETS 0000000  00000000 000000000  00000000 000000000  000000 00000
 PRIMARY DATA SETS 0000000  00000000 000000000  00000000 000000000  000000 00000
BACKUP
 DAILY BACKUP      0000000  00000000 000000000  00000000 000000000  000000 00000
 SUBSEQUENT BACKUP 0000000  00000000 000000000  00000000 000000000  000000 00000
 DELETE BACKUPS    0000000  00000000 000000000  00000000 000000000  000000 00000
RECOVER
 BACKUP - PRIMARY  0000000  00000000 000000000  00000000 000000000  000000 00000
RECYCLE
 BACKUP - SPILL    0000000  00000000            00000000            000000 00000
 MIG L2 - MIG L2   0000000  00000000            00000000            000000 00000
```

We do not show all of the **REPORT** command parameters. The information that is available within the DSRs volume statistics records (VSRs) is enough to give you a good idea of the activity. You might be able to spot patterns for particular days and functions, or identify unusually high activity against specific volumes and take action to spread the work more evenly.

13.4 Using the DFSMSrmm reporting tool

In certain cases, the standard reports and lists from DFSMShsm do not provide the necessary information for the storage manager to determine DFSMShsm usage or to create reports with specific information about data sets, such as the creation date or the number of data sets that were migrated by using a specific management class.

To create these reports, you can use the DFSMSrmm report generator.

Although DFSMSrmm might not be installed on your system, you can still use report generator to create your reports. They are available from ISMF panel option G.

We show how to get used to the report generator, run several predefined reports, and alter the displayed fields.

When you get into the DFSMSrmm report generator, you will see options to work with reports, report types, and the reporting tool, as shown in Figure 13-1.

```
                         DFSMSrmm Report Generator
Option ===>

0  OPTIONS         - Specify dialog options and defaults
1  REPORT          - Work with reports
2  REPORT TYPE     - Work with report types
3  REPORTING TOOL  - Work with reporting tools
4  MIGRATION       - Migration tasks for reporting
X   EXIT           - Exit DFSMSrmm dialog
Enter selected option or END command. For more info., enter HELP or PF1.
```

Figure 13-1 DFSMSrmm Report Generator panel

By entering option `1`, REPORT, you can search for predefined user, installation, or product reports. You can use this window to display the existing reports, or to create a new report. If you search for a report name that does not exist, report generator will return an empty list, so that you can create a new report that you want.

We selected all predefined DFSMShsm reports, starting with ARC* from user and product reports. Figure 13-2 on page 395 shows all of the DFSMShsm-related reports that are defined to the report generator.

```
                        DFSMSrmm Report Definitions        Row 1 to 15 of 16
Command ===>                                                Scroll ===> PAGE

The following line commands are valid: A,D,G,H,J,L,M,N,S, and T

S Name     Report title                   Report type                  User id
- -------- ------------------------------ ---------------------------- -------
  ARCGAB01 ABARS ABACKUP Statistics       DFSMShsm ABARS Report        HSM
  ARCGAR01 ABARS ARECOVER Statistics      DFSMShsm ABARS Report        MHLRES7
  ARCGDB01 DCOLLECT BACKUP DATA           DFSMShsm DCOLLECT BACKUP     MHLRES7
  ARCGDD01 DCOLLECT DASD CAPACITY PLANNIN DFSMShsm DCOLLECT DASD CAP   MHLRES7
  ARCGDM01 DCOLLECT MIGRATION DATA        DFSMShsm DCOLLECT MIGRATION  HSM
  ARCGDT01 DCOLLECT TAPE CAPACITY PLANNIN DFSMShsm DCOLLECT TAPE CAP   HSM
  ARCGS001 Statistics for DFSMShsm        DFSMShsm FSR-SMF Records     MHLRES7
  ARCGS002 Statistics for Backup          DFSMShsm FSR-SMF Records     HSM
  ARCGS003 Statistics for Migration       DFSMShsm FSR-SMF Records     HSM
  ARCGS004 Statistics for Recall          DFSMShsm FSR-SMF Records     HSM
  ARCGS005 Statistics for Recovery        DFSMShsm FSR-SMF Records     HSM
  ARCGS006 Statistics for Volume Dump     DFSMShsm FSR-SMF Records     HSM
  ARCGS007 Statistics for Restore from Du DFSMShsm FSR-SMF Records     MHLRES7
  ARCGS008 Statistics for FRBACKUP        DFSMShsm FSR-SMF Records     HSM
  ARCGS009 Statistics for FRRecover       DFSMShsm FSR-SMF Records     HSM
```

Figure 13-2 Searching for all DFSMShsm-related predefined reports

Several DFSMShsm reports are predefined to the report generator. Each report uses a different input, and produces a different output. From this panel, you can add, copy, update, or submit reports for processing.

To display information about the data that will be displayed on the report, enter `S` to select the report. If you plan to add or remove fields on the report, you can create a new report or document the changes in a place that other users of this report can access.

We selected report ARCGMD01 to collect information about data set migration. The report collects information about migration date and time, data set name, original volume, number of times migrated, and other information.

By entering `G` to generate the report, a new configuration window opens. In this window, we need to provide the necessary information so that the report generator can create the JCL to extract the report.

The input data set field is the data set that is used by the report generator as a source to extract the information. Use a data set that your job has access to READ if the data set exists, or ALTER if the data set will be created by the report generator.

If you do not have the DCOLLECT created for this run, or records that were extracted from System Management Facilities (SMF) for other reports, you can set “Create report data” to `Y` (on Figure 13-3 on page 396). The report generator will add a step to create the input file and gather the DCOLLECT or SMF data. Use the skeleton variables to control your extract and report processing, such as the values that are used in SORT include conditions, and data set names.

Figure 13-3 on page 396 shows a sample configuration to create the report JCL.

```
                      DFSMSrmm Report Generation - ARCGDB01
Command ===>

Enter or change the skeleton variables for the generated JCL:

Input data set  . . . . 'MHLRES7.DCOLLECT.INPUT'

Date format . . . . . . YYYYDDD
    (American, European, Iso, Julian, or free form)
    Required if you use variable dates (&TODAY) in your selection criteria.

Create report data  . . Y  (Y/N)
    Choose Y if you want an extract step included into your generated JCL.

Additional skeleton variables, for example if an extract step is included:
Skeleton Variable_1 . .
Skeleton Variable_2 . .
Skeleton Variable_3 . .
    The skeleton selection depends on the reporting macro . . . : ARCUTILP
                                          and macro keyword . . : TYPE=B
  Enter END command to start the report generation or CANCEL
```

Figure 13-3 Setting up the report variables

When all of the definitions are set, you can use PF3 to exit this window. A message that is similar to “`Report JCL ARCGDB01 stored on 'MHLRES7.REPORT.JCL'`” displays at your terminal.

Before you submit your job, consider checking the DFSMShsm CDS names or SMF data sets that report generator used to create the extract data to confirm that they are accurate.

Example 13-32 shows a sample output from the report that was created in the previous steps.

Example 13-32 Sample output from the report generator tool

```
DCOLLECT MIGRATION DATA        - 1 -        2012/09/07        14:23:51

 MIG DATE   MIG TIME   DSN                                            1ST SRC VO
---------   --------   --------------------------------------------   ----------
  2012236   12353803   MHLRES7.HSM.BCDS                               SBOX75
  2012237   22200769   MHLRES7.BV                                     MLD10C
  2012242   16573407   MHLRES7.NONSMSE                                MLD20C
  2012249   00050086   MHLRES7.NONSMSA.DSN1                           ML940D
  2012250   10105729   MHLRES7.ACSTEST.LISTDS                         MLD83A
```

If you determine that none of the existing reports provide the exact information that you want, you can copy a similar report, and update it to include the information that you need.

To copy an existing report to a new report, you can search for it on the report option, and then enter option `N` to copy the report definition. You must define a new report name, which will be stored in your user report library.

In Figure 13-4 on page 397, we show the process to create a new report that is based on an existing report.

```
                      DFSMSrmm Report Definitions        Row 1 to 14 of 16
Command ===>                                               Scroll ===> PAGE

The following line commands are valid: A,D,G,H,J,L,M,N,S, and T

S Name     Report title                  Report type                 User id
- -------- EssssssssssssssssssssssssssssssssssssssssssssssssssssN ----- -------
N ARCGAB01 e                                                   e       MHLRES7
  ARCGAR01 e                                                   e       MHLRES7
  ARCGDB01 e                                                   e P     MHLRES7
  ARCGDD01 e Enter the report name . . . . ARCABNEW            e CAP   MHLRES7
  ARCGDM01 e                                                   e TION  MHLRES7
  ARCGDT01 e                                                   e CAP   HSM
  ARCGS001 DssssssssssssssssssssssssssssssssssssssssssssssssssssM s     MHLRES7
  ARCGS002 Statistics for Backup         DFSMShsm FSR-SMF Records     HSM
  ARCGS003 Statistics for Migration      DFSMShsm FSR-SMF Records     HSM
  ARCGS004 Statistics for Recall         DFSMShsm FSR-SMF Records     HSM
  ARCGS005 Statistics for Recovery       DFSMShsm FSR-SMF Records     HSM
  ARCGS006 Statistics for Volume Dump    DFSMShsm FSR-SMF Records     HSM
  ARCGS007 Statistics for Restore from Du DFSMShsm FSR-SMF Records    MHLRES7
  ARCGS008 Statistics for FRBACKUP       DFSMShsm FSR-SMF Records     HSM
```

Figure 13-4 Copying an existing report to a new report

After you copy the report, you can perform a new search by using its report name, and use S to select the report to update. This option shows you all of the available fields on the report. If the field that you need is not listed in this report, you can try to use another predefined report as source, or code your own program to read the DCOLLECT or SMF data, and extract the information that you need.

In Figure 13-5, we updated the ARCABNEW report to *not* include a return code and reason code for ABARS ABACKUP statistics.

```
                   DFSMSrmm Report Definition - ARCABNEW   Row 15 to 21 of 48
Command ===>                                               Scroll ===> PAGE

Report title . . . ABARS ABACKUP Statistics                                +
Report footer  . .
Reporting tool . : ICETOOL                                Report width:  267
Enter "/" to select option
   Edit the help information for this report

Use END to save changes or CANCEL
The following line commands are valid: S, and R

S CO SO  Field name           Column header text                  CW  Len Typ
- -- --- -------------------- ----------------------------------- --- --- ---
  14     WFSR2ABACKUP_TOTALSP TOTAL KB                              8   8 N
  15     WFSR2CPUTIME         CPU TIME                              8   4 N
  16     WFSR2ACCOUNT_CODE    ACC CODE                             32  32 C
         WFSR2RC              RC (hex)                              8   4 B
         WFSR2REAS            RSN (hex)                             9   4 B
*        WFSR2TYPE            TYPE - Use browse macro (M)           9   1 B
```

Figure 13-5 Updating the ARCABNEW report

Exit the edit window by pressing PF3 to save changes.

13.5 Utilities in ARCTOOLS for DFSMShsm reporting

DFSMShsm ships utilities for reporting and other uses. All of the utilities are contained in the ARCTOOLS member in SYS1.SAMPLIB.

To obtain these utilities, use the following steps:

1. Edit the ARCTOOLS member of SYS1.SAMPLIB and change the fields as instructed.
2. Run ARCTOOLS to allocate the HSM.SAMPLE.TOOL data set with several members that can be used for reporting.

We introduce only the SCANBLOG, SCANMLOG, SCANFSR, FSRSTAT, and QUERYSET utilities:

SCANBLOG utility This sample REXX exec scans several days' worth of backup activity logs and summarizes the results.

SCANMLOG utility This sample REXX exec scans several days' worth of migration activity logs and summarizes the results.

SCANFSR utility This sample REXX exec scans the FSR data that was extracted from SMF (by using the IFASMFDP program) and summarized by FSR type and management class.

FSRSTAT utility This sample REXX program reads FSR data and presents statistical results. In certain cases, the data is presented as a histogram to group data in certain categories for further analysis.

QUERYSET utility You cannot use the `QUERY` command to send its output to a data set. By using TSO/E extended consoles, this sample REXX allows programs to be submitted that can issue a console command, such as `QUERY`, and to get the results back for interpretation.

13.6 Using DCOLLECT

Another method of gathering data that relates to space utilization and capacity planning is to use *DCOLLECT*, an access method services (AMS) function that is run in a batch environment. With DCOLLECT, you can obtain information about the following features:

- Active data sets
- Volumes
- Migrated data sets
- Backup data sets
- DASD capacity planning
- Tape capacity planning

For a detailed description of DCOLLECT, see *DFSMS Access Method Services for Catalogs,* SC26-7394.

We show how you can produce an output data set that can be used later as input to a report creation package.

You can produce the output data set by invoking IDCAMS DCOLLECT or by using the ARCUTIL utility, which is a DFSMShsm-related program that is used to capture DFSMShsm information.

If DFSMShsm information is requested from within a DCOLLECT batch job, the ARCUTIL load module is called, which is almost transparent to the user. The parameters, although specified in a different place, are the same parameters that you use on the DCOLLECT SYSIN DD card.

The sample DCOLLECT JCL that is shown in Example 13-33 calls ARCUTIL transparently.

Example 13-33 Sample DCOLLECT JCL to call ARCUTIL

```
//DCOLLECT   JOB   ,'DCOLLECT RUN',CLASS=Z,MSGCLASS=H,REGION=4M
//*
//STEP1   EXEC   PGM=IDCAMS
//*
//SYSPRINT  DD  SYSOUT=*
//ARCSNAP   DD  SYSOUT=*
//MCDS      DD  DSN=HSM.MCDS,DISP=SHR
//BCDS      DD  DSN=HSM.BCDS,DISP=SHR
//DCOUT     DD  DSN=MHLRES5.DCOLLECT.OUTPUT,
//              DISP=(NEW,CATLG,DELETE),
//              SPACE=(1,(859,429)),AVGREC=K,
//              DSORG=PS,RECFM=VB,LRECL=264
//SYSIN     DD  *
     DCOLLECT -
           OUTFILE(DCOUT) -
           NODATAINFO -
           NOVOLUMEINFO -
           MIGRATEDATA -
           BACKUPDATA -
           CAPPLANDATA -
           MIGRSNAPERR
```

If you want to invoke ARCUTIL directly by using the same DCOLLECT specified parameters to capture DFSMShsm information, use the JCL that is shown in Example 13-34.

Example 13-34 Sample ARCUTIL JCL

```
//JOB2    JOB 'ARCUTIL RUN',REGION=4M
//STEP2   EXEC PGM=ARCUTIL,PARM='DCOLLECT MIGD CAPD BACD MSERR'
//ARCSNAP SYSOUT=*
//ARCTEST SYSOUT=*
//ARCDATA DD DSN=MY.COLLECT.DATA,DISP=(,CATLG),
//           SPACE=(CYL,(5,10)),UNIT=SYSDA
//MCDS    DD DSN=HSM.MCDS,DISP=SHR
//BCDS    DD DSN=HSM.BCDS,DISP=SHR
```

This JCL gathers the same information as the IDCAMS DCOLLECT execution, but this time you directly invoke ARCUTIL.

The DCOLLECT parameters are specified on the EXEC parm statement. DCOLLECT must be the first command that is specified. It can be followed by any of the other `DCOLLECT` keywords.

The output that is generated by IDCAMS DCOLLECT, or ARCUTIL, is not a report-like format. Use another program to read this data and generate the reports that you require. You can use DFSORT or ICETOOL to create the necessary reports.

14

Problem determination

A correctly configured installation of DFSMShsm will execute without any issues most of the time. However, if an issue occurs, be prepared to collect the correct documentation on the first occurrence of the issue. This way, diagnosing the root cause of the issue begins immediately, preventing the need to re-create or to wait for a recurrence of the issue to collect the appropriate documentation.

In this chapter, we describe the most important sources of problem determination documentation for DFSMShsm. We go through sample **AUDIT** commands and provide scenarios about when to use them.

14.1 Problem determination documentation

The most important sources of problem determination documentation for DFSMShsm are described:

- Problem Determination Aid (PDA)
- DFSMShsm dumps
- SYSLOG
- JOBLOG
- Activity logs

A few common problem scenarios and the actions to address them are described. An overview of the DFSMShsm **AUDIT** command and how it can be used during both problem determination and recovery are provided.

14.1.1 Problem Determination Aid

PDA is the single most important diagnostic aid that is used to determine the root cause of most of the issues that occur in DFSMShsm.

PDA is a trace facility that is built into DFSMShsm that traces the module flow in DFSMShsm. This tracing is accomplished by incorporating PDA trace points at important locations in the DFSMShsm modules (for example, when a module is entered or exited). When a task in DFSMShsm encounters one of these trace points, the information that is contained in the trace point is recorded in an internal trace buffer in the DFSMShsm address space. When the internal trace buffer is full, the data in the trace buffer is written (optionally) to the PDA data sets that the user defined on DASD.

The PDA facility is automatically enabled at DFSMShsm startup. You enable or disable PDA processing with the **SETSYS PDA(NONE|ON|OFF)** command. We highly recommend that you leave PDA tracing enabled when DFSMShsm is active. The PDA trace options are described:

- **SETSYS PDA(NONE)**, when specified at DFSMShsm startup, causes no data to be gathered; the trace log data sets are not opened.
- **SETSYS PDA(ON)** causes DFSMShsm to request storage for data accumulation and opens the DASD trace log data sets if they are allocated. If the trace log data sets are not allocated, data is accumulated in internal storage only.
- **SETSYS PDA(OFF)** stops data accumulation. The trace log data sets remain open.

Two PDA output data sets need to be created on DASD (if you choose to retain the PDA data on DASD). For DFSMShsm to use these data sets after they are created, they must be defined to the ARCPDOX and ARCPDOY DD names in the DFSMShsm startup procedure.

Writing to the PDA data sets

DFSMShsm always writes to the data set name that is defined to the ARCPDOX DD name. For example, if data set HSM.PDOX is defined to the ARCPDOX DD name, DFSMShsm will always write to HSM.PDOX. When this data set becomes full, either an ABENDB37 or an ABENDD37 is generated against the data set, which is normal. DFSMShsm uses these ABENDs to determine that it needs to swap the PDA data set that it is writing to.

This swap is why a user must define two PDA data sets in the DFSMShsm startup procedure: one to the ARCPDOX DD name and one to the ARCPDOY DD name. When the ARCPDOX data set becomes full, DFSMShsm swaps the data set names that are associated with the ARCPDOX and ARCPDOY DD names. That is, DFSMShsm renames each data set by using the other data set's name. For example, if data set HSM.PDOX is defined to the ARCPDOX DD name and HSM.PDOY is defined to the ARCPDOY DD name, HSM.PDOX is renamed to HSM.PDOY and HSM.PDOY is renamed to HSM.PDOX.

After the swap occurs, DFSMShsm starts writing to the data set name that is associated with the ARCPDOX DD name. By using the preceding example, DFSMShsm starts writing to the HSM.PDOX data set (which was previously named HSM.PDOY). When this swap occurs, HSM identifies an ARC0037I message.

Allocating the PDA data sets

Your PDA data sets are usually allocated as part of the starter set. Example 14-1 contains sample JCL that can be used to allocate PDA data sets if the initial allocations are too small. These data sets must be created on the same volume.

Example 14-1 Sample JCL to allocate PDA data sets

```
//MHLRES4J JOB MSGLEVEL=1,CLASS=A
//STEP1 EXEC PGM=IEFBR14
//ARCPDOX DD DSN=HSM.HSMPDOX,DISP=(,CATLG),VOL=SER=HSM14C,
// UNIT=3390,SPACE=(CYL,(30,2))
//ARCPDOY DD DSN=HSM.HSMPDOY,DISP=(,CATLG),VOL=SER=HSM14C,
// UNIT=3390,SPACE=(CYL,(30,2))
/*
```

The larger you define the PDA data sets, the longer the time period of trace data the data sets are able to contain. When the ARCPDOX data set becomes full and the data set names are swapped, any previously recorded data in the data sets now being written to is overlaid. Therefore, unless you archive the ARCPDOY data set after the ARC0037I is issued, you will begin to lose this data the next time the ARCPDOX data set fills and another swap occurs. You need to copy the ARCPDOY data set to tape or DASD before the ARCPDOX data set becomes full to archive it.

Enabling and switching the PDA logs

If the PDA trace is not enabled at DFSMShsm startup, or you disabled it, issue the following command at a system console to enable PDA tracing:

```
f hsmprocname,SETSYS PDA(ON)
```

The PDA data sets are automatically swapped when DFSMShsm is started. To manually switch the data sets, issue the following command at a system console:

```
f hsmprocname,SWAPLOG PDA
```

Archiving PDA data

A generation data group (GDG) can be used to archive the PDA trace data. You can then copy the generation data set (GDS) to tape or direct it to a DASD volume where it will be migrated to tape.

Example 14-2 on page 404 shows sample JCL to create a GDG to archive the PDA trace data.

Example 14-2 Sample JCL to define a GDG base for archiving PDA

```
//MHLRES4G JOB  MSGLEVEL=1,MSGCLASS=A
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
  DEFINE GDG (NAME('HSM.HSMTRACE') LIMIT(30) (SCRATCH)/*
```

Sample JCL to define a GDG base for archiving PDA

Example 14-3 shows sample JCL for copying the PDA data set to tape as a GDS for archival.

Example 14-3 Sample JCL to copy PDA to tape

```
//MHLRES4C JOB  MSGLEVEL=1,MSGCLASS=A
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   DUMMY
//SYSUT1   DD   DSN=HSM.HSMPDOY,DISP=SHR
//SYSUT2   DD   DSN=HSM.HSMTRACE(+1),
//         UNIT=TAPE,
//         DISP=(NEW,CATLG,CATLG),VOL=(,,1),
//         DCB=(HSM.HSMPDOY)
```

When the PDA trace log switch occurs, DFSMShsm issues message ARC0037I. You can use an automation product to trap this message and automatically submit a copy job to archive the PDA trace data. This practice provides a sequential history of trace data over time so that the data is available when needed for diagnosing problems.

Although PDA provides an excellent overview of the module flow in DFSMShsm over the time frame that an issue occurs, it is not able to provide a snapshot of the state of DFSMShsm exactly when an abend occurs. This functionality is provided by supervisor call (SVC) dumps.

14.1.2 DFSMShsm dumps

One of the most important diagnostic features that MVS provides is the ability to take a snapshot of an address space when an error is detected or when a user believes that an error is present. The following facilities, which exist in MVS, allow this data capture:

- Automatic dumps
- Console dumps
- Serviceability level indication processing (SLIP) traps

How these facilities can be used to capture errors in DFSMShsm is described.

Automatic dumps

DFSMShsm provides two **SETSYS** parameters that control how and what type of dumps are taken when an abend occurs in the DFSMShsm address space:

- SYS1DUMP
- NOSYS1DUMP

SETSYS SYS1DUMP

When SETSYS SYS1DUMP is specified and an abnormal end of task (abend) occurs in the DFSMShsm address space, MVS requests a dump to be written to either a pre-allocated SYS1.DUMPxx data set or to a dynamically allocated dump data set that is named according to your installation standards. The dump is unformatted and requires the use of Interactive Problem Control System (IPCS) to be analyzed. This dump type will be requested by the IBM Support Center to diagnose the root cause of an abend in the DFSMShsm address space. This setting is recommended.

SETSYS NOSYS1DUMP

When SETSYS NOSYS1DUMP is specified and an abend occurs in the DFSMShsm address space, the dump that is produced is directed to the dump-related DD statements that are specified in the DFSMShsm startup procedure. The following DD statements are applicable:

- SYSABEND
- SYSUDUMP
- SYSMDUMP

SYSABEND and SYSUDUMP dumps are both *formatted*. They are user-readable and do not require IPCS for analysis. SYSMDUMP dumps are unformatted and require IPCS for analysis.

We recommend that you specify the **`SETSYS SYS1DUMP`** command in the ARCCMDxx member of SYS1.PARMLIB. However, if you prefer to use SETSYS NOSYS1DUMP, ensure that you specify that a SYSMDUMP is taken. The IBM Support Center prefers that clients gather unformatted dumps as opposed to formatted dumps. Formatted dumps do not contain the same amount of information as unformatted dumps and decrease the likelihood that the root cause of an issue can be determined from them.

Console dumps

In addition to automatically generating dumps when abend is detected, MVS provides the ability for an operator to issue a manual dump command to capture a snapshot of an address space at any time. The main reason that we recommend taking a console dump is if you suspect that DFSMShsm processing is stopped. A console dump is the only way to determine exactly what is causing a problem in the DFSMShsm address space to occur.

Although the dump command and its associated parameters can be issued at the same time, we recommend that you create IEADMCxx members in SYS1.PARMLIB in advance that contain the dump parameters to use to capture dumps that are associated with DFSMShsm processing. After these IEADMCxx members in SYS1.PARMLIB are created, you only need to execute the following command at the system console to create a dump:

```
DUMP COMM=(text),PARMLIB=xx
```

Text is the title that you want for the dump. The *xx* is the PARMLIB member that contains the address spaces and parameters that you want dumped.

You need to create IEADMCxx PARMLIB members for three sets of console dump parameters if a problem is experienced in DFSMShsm:

1. A dump of only the DFSMShsm address space, as shown in Example 14-4.

Example 14-4 Sample IEADMCxx parameters to dump only the DFSMShsm address space

```
JOBNAME=(DFHSM),
   SDATA=(ALLNUC,CSA,LPA,LSQA,PSA,RGN,SQA,SWA,TRT,SUM,GRSQ),END
```

2. A dump of DFSMShsm and other address spaces, as shown in Example 14-5.

Example 14-5 Sample IEADMCxx parameters to dump DFSMShsm and other address spaces

```
JOBNAME=(DFHSM,CATALOG,GRS),
   SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SWA,TRT,SUM),END
```

3. A dump of DFSMShsm and the common recall queue (CRQ) structure (if CRQ is implemented in your environment), as shown in Example 14-6.

Example 14-6 Sample IEADMCxx parameters to dump the DFSMShsm address space and CRQ

```
JOBNAME=(DFHSM),
STRLIST=(STRNAME=SYSARC_xxxxx_RCL,(LNUM=ALL,ADJ=CAP,EDATA=UNSER),
LOCKE,(EMC=ALL),ACC=NOLIM),
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SWA,TRT,SUM,XESDATA,COUPLE),
END
```

When you dump the CRQ, you need to replace *xxxxx* with the name of the CRQ structure as it was specified in the following command:

`SETSYS COMMONQUEUE(RECALL(CONNECT(`*xxxxx*`)))`

For example, if the CRQ name that is used in this command is PLEX1, the **STRNAME** in the parameter in the dump is **STRNAME=SYSARC_PLEX1_RCL**.

SLIP traps

Like console dumps, SLIP traps allow an unformatted dump of DFSMShsm to be taken when an error condition is detected. Unlike console dumps however, with SLIP traps, you can specify the system conditions that must be met for an automatic dump to be taken.

Just as console dumps allow the specification of the dump parameters in advance in an IEADMCxx PARMLIB member, with SLIP traps, you can specify the SLIP trap parameters in advance in PARMLIB member IEASLPxx.

For example, you might want a dump of DFSMShsm to be automatically generated if one of the control data sets (CDSs) or the journal fails to be backed up during CDS backup. This event generates an ARC0744E message. You can write the following SLIP in its own IEASLPxx member in SYS1.PARMLIB to automatically capture a dump of DFSMShsm if this message is issued. See Example 14-7.

Example 14-7 Sample IEASLPxx parameters to dump the DFSMShsm address space on ARC0744E

```
SLIP SET,A=SVCD,MSGID=ARC0744E,
JOBLIST=(DFHSM),
SDATA=(ALLNUC,ALLPSA,CSA,LPA,LSQA,NUC,PSA,RGN,SUM,SQA,SWA,TRT,GRSQ),END
```

After this SLIP is created, execute the following command at the system console to set the SLIP trap:

```
SET SLIP=xx
```

Where *xx* is the value of the newly created IEASLP*xx* member. For more information about how to write SLIP trap commands, see z/OS *MVS System Commands,* SA22-7627.

Dump analysis and elimination

Dump analysis and elimination (DAE) is an MVS function that eliminates duplicate dumps in MVS systems or across MVS systems.

DFSMShsm generates dumps to gather first-failure diagnostic information. Problems that occur multiple times, perhaps on different hosts, typically generate a storage dump. The initial dump is helpful for diagnosing the problem; additional dumps for the same problem usually are not needed.

Controlling DFSMShsm storage dumps is a major issue with users. The ability to correctly suppress duplicate dumps is an important step toward successfully managing dump data sets.

DAE implementation

To implement DAE for DFSMShsm, the following tasks must be completed:

- SETSYS SYS1DUMP must be specified. SYS1DUMP is the DFSMShsm default.
- PARMLIB member ADYSETxx must be coded with the **SUPPRESSALL** keyword, for example:

  ```
  DAE=START,RECORDS(400),SVCDUMP(MATCH,UPDATES,SUPPRESSALL)
  ```

- A single DAE data set can be shared across systems in a sysplex. The coupling services of the cross-system coupling facility (XCF) and GRS must be enabled for the DAE data set to be shared in a sysplex environment and for dumps to be suppressed across MVS systems.

DFSMShsm uses DAE for functions that run in both the primary address space and the aggregate backup and recovery support (ABARS) secondary address space.

DAE does not suppress SYSABEND, SYSUDUMP, SYSMDUMP, or SNAP dumps or dumps that originate from **SLIP** or **DUMP** operator commands. Because these dumps are taken only on demand, suppression is not desirable.

This support does not apply to dumps that are produced by DFSMShsm as a result of the **TRAP** command. For more information about setting up the DAE data set, see *z/OS MVS Diagnosis: Tools and Service Aids*, GA22-7589.

14.1.3 SYSLOG

The SYSLOG reports events that are routed to the system operator to ensure the continued successful functioning of the z/OS operating system and the programs that execute in it. It is important to monitor the SYSLOG for any DFSMShsm messages that indicate an error condition, and for any messages that are displayed outside of DFSMShsm that affect DFSMShsm processing.

For example, if an I/O error occurred on a drive that was in use by DFSMShsm, an IOS000I message that indicates this error is only written to the SYSLOG; it is not sent to any DFSMShsm-specific log. For this reason, if an issue occurs in DFSMShsm processing, it is important to retain SYSLOGs that lead up to and include the time frame in which the issue occurred.

14.1.4 JOBLOG

The DFSMShsm job log (JOBLOG) contains messages that span from the time that an instance of DFSMShsm starts through the time that it is shut down. The messages that it contains are also output to the SYSLOG, but unlike the SYSLOG, the job log contains messages that are specific to only a single instance of DFSMShsm.

The job log is useful for reviewing the messages that are issued during the startup process of an instance of DFSMShsm to ensure that all of the setup commands that are contained in the ARCCMDxx PARMLIB member were processed successfully. For this reason, it is important to retain the job log for review of an instance of DFSMShsm that experiences an issue.

14.1.5 Activity logs

The activity logs report on the backup, dump, migration, ABARS, and command processing of DFSMShsm in your system. DFSMShsm has five activity logs:

- The migration activity log provides information about space management activity, including **MIGRATE** commands for volumes and levels, interval migration, and automatic primary and automatic secondary space management.
- The backup activity log provides information about automatic backup, volume command backup, FRBACKUP and FRRECOV activities, and volume command recovery activities.
- The dump activity log provides information about automatic dump, command volume dump, and command volume restore activities.
- The ABARS activity log provides information about aggregate backup and recovery activities.
- The command activity log provides information about TAPECOPY and TAPEREPL activity and records error or informational messages that occur during low-level internal service processing.

With the **SETSYS ACTLOGTYPE** command, you can direct DFSMShsm activity logs to DASD or to a SYSOUT class. For more information, see Chapter 3, “Getting started” on page 29. If you specify **ACTLOGTYPE(DASD)** on the **SETSYS** command, DFSMShsm dynamically allocates DASD data sets for the activity logs with names in the following format:

`HSMACT.`*`Hmcvthost.function.agname.Dyyddd.Thhmmss`*

The following definitions refer to the previous naming format:

Hmcvthost DFSMShsm host ID from the DFSMShsm startup PROC statement, which is preceded by `H`

function Either ABACKUP, ARECOVER, BAKLOG, CMDLOG, DMPLOG, or MIGLOG

agname Aggregate group name (only present if function is ABACKUP or ARECOVER)

Dyyddd Year and day of allocation, which are preceded by `D`

Thhmmss Hour, minute, and second of allocation, which are preceded by `T`

Although the activity logs are lower in importance to problem diagnosis than the documentation that was previously described, they still provide value because the messages that are sent to them can indicate potential issues in DFSMShsm. We recommend that you monitor the activity logs for any indications of a potential issue.

14.2 Problem recognition, documentation collection, and recovery

After a problem occurs in DFSMShsm and you collect the appropriate documentation to enable the diagnosis of the root cause of the issue, begin the recovery process to bring your DFSMShsm back to its fully functional state. Although various issues can arise during DFSMShsm processing, those issues fall into one of two categories: Hung situations (stoppages) in the DFSMShsm address space and the DFSMShsm ARC*xxxx* messages that identify an undesirable condition.

14.2.1 Hung situations in the DFSMShsm address space

If DFSMShsm processing hangs or stops, the processing of any job in the z/OS environment that requires DFSMShsm functionality to succeed might be held.

For example, if a job tries to access a data set but that data set was migrated by DFSMShsm, the data set must be recalled before the job can access the data set. If DFSMShsm hangs before the data set finished recalling, the job cannot continue processing until the hung situation is recovered from.

For this reason, it is important for a DFSMShsm administrator to determine when DFSMShsm processing stopped, to collect the appropriate documentation to diagnose the root cause of the stoppage, and to recover from this condition.

Recognizing a hung condition

Stoppages or hung conditions in DFSMShsm processing are caused by DFSMShsm tasks that cannot obtain the required resources to continue processing. DFSMShsm has two variations of the **QUERY** command that can be used to recognize that DFSMShsm processing might be hung: **QUERY WAITING** and **QUERY ACTIVE(TCBADDRESS)**.

Example 14-8 shows the output of the **QUERY WAITING** command in an instance of DFSMShsm that does not have any work that is waiting to be executed.

Example 14-8 Output from the QUERY WAITING command

```
09.24.40 f dfhsm,query waiting
09.24.40 STC00018  ARC0101I QUERY WAITING COMMAND STARTING ON HOST=1
09.24.40 STC00018  ARC0168I WAITING MWES: MIGRATE=00000000,
ARC0168I (CONT.) RECALL=00000000, DELETE=00000000, BACKUP=00000000,
ARC0168I (CONT.) RECOVER=00000000, COMMAND=00000000, ABACKUP=00000000,
ARC0168I (CONT.) ARECOVER=00000000, FRBACKUP=00000000,
ARC0168I (CONT.) FRRECOV=00000000, TOTAL=00000000
09.24.40 STC00018  ARC0101I QUERY WAITING COMMAND COMPLETED ON HOST=1
```

We recommend that you issue the **QUERY WAITING** command periodically to verify that DFSMShsm is continuing to process the Management Work Elements (MWEs) that are listed in the ARC0168I message. If this command is issued every 15 minutes, and over a few hour period, all of the MWEs either stay at the same number or increase, this situation might indicate that DFSMShsm processing is held up for an unknown reason and further analysis is required.

If you suspect that DFSMShsm processing is hung, issue the **QUERY ACTIVE(TCBADDRESS)** command. Example 14-9 shows the output of this command in an instance of DFSMShsm that has five active recall tasks and CDS backup executing.

Example 14-9 Output from the QUERY ACTIVE(TCBADDRESS) command

```
09.24.44          f dfhsm,query active(tcbaddress)
09.24.44 STC00016  ARC0101I QUERY ACTIVE COMMAND STARTING ON HOST=1
09.24.44 STC00016  ARC0142I CDS BACKUP, CURRENTLY IN PROCESS,
ARC0142I (CONT.) TCB=x'009B4C68'
09.24.44 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST9 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000058 ON HOST 1 , TCB=x'009A7E88'
09.24.44 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST1 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000060 ON HOST 1 , TCB=x'009A6238'
09.24.44 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST2 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000061 ON HOST 1 , TCB=x'009B25D0'
09.24.44 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST3 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000062 ON HOST 1 , TCB=x'009B2438'
09.24.44 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST4 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000063 ON HOST 1 , TCB=x'009B22A0'
09.24.44 STC00016  ARC0101I QUERY ACTIVE COMMAND COMPLETED ON HOST=1
```

When you review the output from the **QUERY ACTIVE(TCBADDRESS)** command on your own system, look for any messages that indicate a unit of work, for example the recall of a data set, that you think is finished is still active. This situation indicates that the unit of work itself might cause the hung status of the rest of DFSMShsm, or the unit of work might be a victim of another unit of work that caused the hung situation.

To further narrow down the unit of work that might cause the hung situation that DFSMShsm is experiencing, you can use the following Global Resource Serialization (**GRS**) command to display any resource contention that is detected in the system:

```
D GRS,CONTENTION
```

Example 14-10 shows an example of a system that is experiencing resource contention on the ARCGPA/ARCCAT resource.

Example 14-10 Output from the D GRS,CONTENTION command

```
09.25.45          D GRS,CONTENTION
09.25.45          ISG343I 09.25.45 GRS STATUS 022
S=SYSTEM  ARCGPA   ARCCAT
SYSNAME        JOBNAME        ASID     TCBADDR   EXC/SHR    STATUS
SYSTEM1   DFHSM              0030      009A7E88   SHARE      OWN
SYSTEM1   DFHSM              0030      009B4C68 EXCLUSIVE    WAIT
SYSTEM1   DFHSM              0030      009A6238   SHARE      WAIT
SYSTEM1   DFHSM              0030      009B25D0   SHARE      WAIT
SYSTEM1   DFHSM              0030      009B2438   SHARE      WAIT
SYSTEM1   DFHSM              0030      009B22A0   SHARE      WAIT
NO LATCH CONTENTION EXISTS
```

If the output from this command indicates contention for a DFSMShsm resource, you might be able to match the TCBs that are identified in this output as holding the DFSMShsm resource that is under contention with the TCBs that are identified as a result of the **QUERY ACTIVE(TCBADDRESS)** command. If you can, you might then use the functionality of the **CANCEL TCBADDRESS** command to attempt to free the resources that are hanging up the rest of the DFSMShsm processing. Example 14-10 on page 410 shows that TCB 009A7E88 holds ARCGPA/ARCCAT, which matches recall request number 58 in Example 14-9 on page 410.

> **Note:** The **CANCEL TCBADDRESS** command is only designed to cancel TCBs that are identified by way of the **QUERY ACTIVE(TCBADDRESS)** command. If it is used to cancel a TCB that is *not* identified in the **QUERY ACTIVE(TCBADDRESS)** command output, the task will be canceled, but the results are unpredictable.

In the example that is shown in Example 14-10 on page 410, the DFSMShsm resource ARCGPA/ARCCAT is under contention by six TCBs. TCB 009A7E88 holds a SHARED enqueue on the resource, preventing TCB 009B4C68 from obtaining the EXCLUSIVE enqueue that it needs to continue processing. Any TCB that requests an enqueue on ARCGPA/ARCCAT after TCB 009B4C68 must wait until TCB 009B4C68 enqueues and dequeues the resource for the enqueue to succeed and that task to continue processing.

> **Note:** Do not issue the **CANCEL TCBADDRESS** command without first taking a console dump of DFSMShsm by using the instructions in Example 14-4 on page 406 and without following the correct procedure for using the **CANCEL TCBADDRESS** command that we describe next.

Collecting the documentation of a hung situation

Before you act to resolve a hung condition, you must take a console dump of DFSMShsm by using the parameters that are described in Example 14-4 on page 406. If you suspect or are aware that the hung condition involves other address spaces, use the parameters that are described in Example 14-5 on page 406. If the hung condition involves data set recalls that are not processing and you are using CRQ, use the parameters that are described in Example 14-6 on page 406 to take a console dump that includes a dump of the CRQ structure.

After console dumps of the involved address spaces are collected, use the **CANCEL TCBADDRESS** command to attempt to free the resources that are causing the hung situation in DFSMShsm processing. In addition to the console dumps, we recommend that you retain the PDA and SYSLOG for at least the 2-hour period that leads up to and includes the time that the dump was taken.

Recovering from a hung situation

DFSMShsm provides the **CANCEL TCBADDRESS** command to allow the administrator to cancel tasks that are hung and that prevent other tasks from processing normally. **CANCEL TCBADDRESS** is a powerful command that you use only to cancel a task that you suspect is hung and that prevents other tasks from processing normally. This command must *not* be used as a regular maintenance command to cancel tasks that are actively processing to completion (for example, canceling the migration or recall of a large data set because it is taking a long time to move the data). If you use this command to cancel a task that is actively nearing completion, the results are unpredictable.

The TCBADDRESS that you choose to cancel with the **CANCEL** command must be obtained from the **QUERY ACTIVE(TCBADDRESS)** command. Before you issue the **CANCEL** command, you must first issue the **HOLD** command to hold the function that contains the task that you are canceling.

Example 14-11 shows an example of using the **HOLD** command to hold the recall function.

Example 14-11 Output from the HOLD RECALL command

```
09.26.16          f dfhsm,hold recall
09.26.16 STC00017  ARC0100I HOLD COMMAND COMPLETED
```

After the function that you are holding receives the ARC0100I message, issue the **QUERY ACTIVE(TCBADDRESS)** command one more time to verify that the task that you want to cancel is still identified. Example 14-12 shows the output from the **QUERY ACTIVE(TCBADDRESS)** command.

Example 14-12 Output from the QUERY ACTIVE(TCBADDRESS) command

```
09.27.32          f dfhsm,query active(tcbaddress)
09.27.32 STC00016  ARC0101I QUERY ACTIVE COMMAND STARTING ON HOST=1
09.27.32 STC00016  ARC0142I CDS BACKUP, CURRENTLY IN PROCESS,
ARC0142I (CONT.) TCB=x'009B4C68'
09.27.32 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST9 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000058 ON HOST 1 , TCB=x'009A7E88'
09.27.32 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST1 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000060 ON HOST 1 , TCB=x'009A6238'
09.27.32 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST2 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000061 ON HOST 1 , TCB=x'009B25D0'
09.27.32 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST3 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000062 ON HOST 1 , TCB=x'009B2438'
09.27.32 STC00016  ARC0162I RECALLING DATA SET DERDMANN.MIGTST4 FOR USER
ARC0162I (CONT.) HSMATH0, REQUEST 00000063 ON HOST 1 , TCB=x'009B22A0'
09.27.32 STC00016  ARC0101I QUERY ACTIVE COMMAND COMPLETED ON HOST=1
```

One enhancement in z/OS Data Facility Storage Management Subsystem (DFSMS) V2.1 was to include the host IDs, tape volsers, and device addresses for those tasks that are processing tape in the ARC0162I message, as shown in Example 14-13.

Example 14-13 ARC0162I message text

```
ARC0162I {MIGRATING | BACKING UP | RECALLING | RECOVERING
 | DELETING | RESTORING | FRRECOV OF} DATA SET dsname
 FOR USER userid, REQUEST request ON HOST hostid <<-- host ID
[,TCB=X'tcbaddress'']:
VOL = {volser1 | NONE}, ADDR = {address1 | NONE};  <-- volser
VOL = {volser2 | NONE}, ADDR = {address2 | NONE};  <-- devaddr
VOL = {volser3 | NONE}, ADDR = {address3 | NONE}]

Where volser1, volser2 or volser3 are the tape volume serial numbers
Where address1, address2 or address3 are the device addresses of the tape volumes.
```

If the TCB address is identified, you can issue the **CANCEL TCBADDRESS(x'aaaaaaaa')** command to cancel the task, as shown in Example 14-14 on page 413.

Example 14-14 Output from the CANCEL TCBADDRESS(x'aaaaaaaa') command

```
09.28.50 f dfhsm,CANCEL TCBADDRESS(X'009A7E88')
09.28.50 STC00018  ARC0931I (H)CANCEL COMMAND COMPLETED, NUMBER OF TASKS
ARC0931I (CONT.) CANCELLED=1
09.28.50 STC00018  ARC0003I ARCRSTR1 TASK ABENDED, CODE 047D0000 IN
ARC0003I (CONT.) MODULE UNKNOWN AT OFFSET 0000, STORAGE LOCATION
ARC0003I (CONT.) 80E27174
```

When the **CANCEL TCBADDRESS** command is executed, DFSMShsm issues an ABEND7D0 for the task that is being canceled. When this ABEND7D0 occurs, DFSMShsm error recovery processing is invoked to clean up the task to the best of its ability.

The error recovery can take time to process to completion, so it is not unusual for a subsequent **QUERY ACTIVE(TCBADDRESS)** command to still identify the TCB that the **CANCEL** command was issued against. If the TCB is still identified in the **QUERY ACTIVE(TCBADDRESS)** command after 10 minutes, issue a second **CANCEL** command of this task. If this subsequent **CANCEL** command does not cancel the task successfully and the DFSMShsm processing still appears to be hung, contact the IBM Support Center for assistance.

After the task is canceled, be sure to issue a **RELEASE** command of that function to allow DFSMShsm to resume processing that function. Example 14-15 shows the output of releasing the recall function.

Example 14-15 Output from the RELEASE RECALL command

```
09.29.50 f dfhsm,release recall
09.29.50 STC00026  ARC0100I RELEASE COMMAND COMPLETED
```

You can also release just the DASD recall function with the new **RECALL(DASD)** parameter on the **RELEASE** command, as shown in Example 14-16.

Example 14-16 Output from the RELEASE RECALL(DASD) command

```
09.45.50 f dfhsm,release recall(dasd)
09.45.50 STC00026 ARC0100I RELEASE COMMAND COMPLETED
```

Addressing issues with the common recall queue

If an issue is experienced in DFSMShsm that involves recalls that are not processing normally and the CRQ is in use, the first step that you must take to address the issue is to dump DFSMShsm and the CRQ by using the parameters in Example 14-6 on page 406. After a dump is taken of the CRQ, disconnect each instance of DFSMShsm from the CRQ by issuing the following command on each DFSMShsm host in an HSMplex that connects to the CRQ:

```
SETSYS COMMONQUEUE(RECALL(DISCONNECT))
```

When this command is specified in an DFSMShsm host, the host performs the following functions:

- Directs all new recall requests to the local recall queue
- Moves all recall requests that originated on that host from the CRQ to the local queue
- Completes any remote requests that were previously selected from the CRQ
- Stops selecting requests from the CRQ

The host disconnects from the CRQ when the following conditions are met:

- The host finishes processing all of the remote requests that were previously selected from the CRQ.
- All requests that originated on that host moved from the CRQ to the local queue.
- Processing is complete for all requests that originated on that host that were selected for remote processing.

After all of the hosts are disconnected from the CRQ, check to see whether the recall issues remain:

- If the recall issues remain, capture the PDA and SYSLOG from the system or systems that are still experiencing the issue and contact the IBM Support Center for assistance.
- If the recall issues are gone, execute the **`AUDIT COMMONQUEUE(RECALL) FIX`** command to instruct DFSMShsm to fix any issues that it detects in the CRQ:
  - If the resulting ARC1544I message indicates that errors were found and fixed, you can reconnect each instance of DFSMShsm to the CRQ and verify that recalls are processing successfully by using the CRQ. If the recall issues return, the CRQ structure might be corrupted beyond the capability of the **`AUDIT`** command to detect and fix problems. In this case, you need to follow the procedure that is entitled "Case 10: Correcting errors in the common recall queue" in the *DFSMShsm Storage Administration Guide,* SC26-0421, to delete and rebuild the CRQ structure.
  - If the ARC1544I message indicates that no errors were identified and fixed, you need to follow the procedure that is entitled "Case 10: Correcting errors in the common recall queue" in the *DFSMShsm Storage Administration Guide,* SC26-0421, to delete and rebuild the CRQ structure.

Note: The CRQ structure must be deleted and rebuilt only after you take a dump of the structure, as described in Example 14-6 on page 406.

14.2.2 DFSMShsm messaging

Throughout DFSMShsm processing, DFSMShsm highlights messages that indicate the success or failure of its many functions. These messages are sent to the logs that are described in 14.1, "Problem determination documentation" on page 402.

DFSMShsm messages are identified to these logs by using the prefix "ARC". DFSMShsm identifies three types of messages:

- Action messages
- Error messages
- Informational messages

Action messages

Action messages indicate that an action must be taken by an operator to address the condition that is identified in the message. Example 14-17 shows an example of an action message that might be issued to the operator.

Example 14-17 Output from a DFSMShsm action message

```
11.15.40 STC00016 *0010 ARC0366A REPLY Y ONLY WHEN ALL 003 TAPE VOLUME(S)
 IS/ARE COLLECTED, N IF ANY NOT AVAILABLE
```

This message indicates that a reply from the operator is required before this DFSMShsm function can continue.

Error messages

Error messages indicate that an error occurred in DFSMShsm processing that must be addressed. Example 14-18 shows an example of an error message that might be issued to the operator.

Example 14-18 Output from a DFSMShsm error message

```
*11.25.42 STC00016 *ARC0744E MCDS COULD NOT BE BACKED UP, RC=0036,
*ARC0744E (CONT.) REAS=0000. MIGRATION, BACKUP, FRBACKUP, DUMP, AND
*ARC0744E (CONT.) RECYCLE HELD.
```

Error messages need to be reviewed to determine why the error occurred and what action needs to be performed to correct the error.

Informational messages

Informational messages show the user the functions that are processing in DFSMShsm. Example 14-19 shows an example of an informational message that might be issued to the operator.

Example 14-19 Output from a DFSMShsm informational message

```
11.43.01 STC00016  ARC0101I QUERY SETSYS COMMAND COMPLETED ON HOST=1
```

Informational messages need to be monitored to ensure that DFSMShsm is functioning correctly.

Reviewing DFSMShsm messages

We recommend that you review the logs for these messages regularly and become familiar with how to look up the meanings of the identified DFSMShsm messages. The DFSMShsm messages are documented in *z/OS MVS System Messages Vol 2 (ARC-ASA),* SA22-7632. This manual provides a description of each message and the actions, if any, that are recommended to address the condition that is described in the message. Example 14-20 shows an example of an ARC1001I informational message.

Example 14-20 Output of an ARC1001I message

```
ARC1001I DERDMANN.MIGTST MIGRATE FAILED, RC=0019, REAS=0001
ARC1219I DATA SET IN USE BY ANOTHER USER OR JOB, MIGRATION REJECTED
```

The Explanation section for this message, which is shown in Figure 14-1 on page 416, indicates that the attempted migration of data set DERDMANN.MIGTST failed with an RC19, REAS=0001 and that the ARC1219I message must be referenced to determine the meaning of the RC19.

```
ARC1001I {dsname | VOL=volser | command} operation FAILED, RC=retcode
       REAS=reascode
```

Explanation: A DFSMShsm request has failed. The following describes the message variables:

- *dsname* indicates the name of the data set.
- *volser* indicates the volume serial number.
- *command* indicates the command and its associated command parameters.
- *operation* indicates the requested operation.
- *retcode* indicates the return code.
- *reascode* indicates the reason code.

A following related message provides you with further information about the failure. The two digits of the message number specified as *nn*, indicate the *retcode*. The related message number also indicates the DFSMShsm function processing when the failure has occurred, as shown on the following table.

Table 1. Return Code Identifier

Function	Message Group
Recovery/Recall	ARC11*nns*
Space Management	ARC12*nns*
Backup	ARC13*nns*
Aggregate Backup/Recovery	ARC6*nnns*
Fast Replication	ARC18*nns*

Figure 14-1 Explanation of an ARC1001I message

As shown in Figure 14-2, the ARC1219I message indicates that the migration failed because the data set was in use by another user or job. As the explanation indicates, the REAS=0001 from the ARC1001I must be used to match to the corresponding reason code meaning or “Reascode Meaning” that is described in the ARC1219I message.

```
ARC1219I DATA SET IN USE BY ANOTHER USER OR JOB, MIGRATION REJECTED
```

Explanation: During DFSMShsm space management processing, an attempt was made to allocate a data set. The allocation failed because:

- The data set was already allocated to another user or job.
- The data set was already allocated to another step within the batch job which issued the DFSMShsm command.
- The migration control data set data set record (MCDS) was updated by another processor in a multiple processing unit environment.
- The data set VTOC entry was modified by another program during DFSMShsm migration processing because the data set was updated by another processor in a multiple-processor environment.

The preceding ARC1001I message or the associated ARC0734I message gives the name of the data set. In message ARC1001I or ARC0734I, the values for *reascode* are:

Reascode Meaning

1

An error occurred in allocating the non-VSAM data set to be migrated.

Figure 14-2 Explanation of an ARC1219I message

In this example, the REAS=0001 indicates that the migration failed because an error occurred in allocating the non-VSAM data set to be migrated.

The System Action and Application Programmer Response show at the end of each message. The System Action indicates the action that DFSMShsm takes when the condition that is described by this message occurs. The Application Programmer Response indicates the action that the user must take when the condition that is described by this message occurs. Figure 14-3 shows that when this condition occurs, DFSMShsm reacts by ending the space management operation of the data set and that if that data set requires migration, the migration must be tried again when the data set is not in use.

System Action: The space management operation of the data set ends. DFSMShsm processing continues.

Application Programmer Response: If forced migration is required, retry the request when the data set is not in use.

Figure 14-3 System Action and Application Programmer Response of an ARC1219I message

If any DFSMShsm messages are identified in these logs that you do not think your DFSMShsm environment needs to be receiving, contact the IBM Support Center for assistance.

z/OS DFSMS V2.1 VSAM record-level sharing error recovery

If VSAM record-level sharing (RLS) is managing the DFSMShsm CDSs, DFSMShsm needs the SMSVSAM server to be up and running. If the SMSVSAM server is unavailable, updates to the CDSs will not occur. In previous releases, z/OS DFSMShsm was shut down and the ARC0061I message was issued:

```
ARC0061I DFSMSHSM SHUTTING DOWN DUE TO SMSVSAM SERVER ERROR
```

Figure 14-4 explains why the message occurred.

```
Explanation: An error with the SMSVSAM server has caused DFSMShsm to lose
access to its control data sets. Either the SMSVSAM server did not initialize
in the allotted amount of time, the SMSVSAM server was repeatedly terminating,
or DFSMShsm was unable to open one or more of the CDSs after a server error
occurred. All attempts to read, write, delete or update control data set
records will fail. Most functions currently being processed will fail. Only
those functions that are allowed to continue while DFSMShsm is in emergency
mode will continue to be processed. To regain access to the control data sets,
DFSMShsm must shut down and be restarted.
System action: DFSMShsm is placed into emergency and shutdown modes. An abend
is issued.
```

Figure 14-4 Explanation of ARC0061I

DFSMShsm will quiesce all CDS I/O activity. ARC0064E is issued, as seen in Figure 14-5 on page 418. Automation can be used to monitor for this message so that storage administrators can be notified that this situation occurred and determine what happened. Both SMSVSAM and DFSMShsm can be restarted.

```
ARC0064E SMSVSAM SERVER ERROR OCCURRED
Explanation: An SMSVSAM server error caused DFSMShsm to lose access to its
control data sets, and the CDS's had to be closed and reopened. One or more
DFSMShsm requests may have failed due to the server error.
System action: DFSMShsm processing continues.
System programmer response: Search for DFSMShsm failure messages that were
issued soon after the SMSVSAM server initialized. If errors occurred, we
recommend you run AUDIT for the function(s) that failed. For example, for data
set migration, backup or recycle errors, run AUDIT DATASETCONTROLS, for Fast
Replication errors, run AUDIT COPYPOOLCONTROLS, etc. After the AUDIT is
complete, failed requests can be reissued if necessary.
Programmer response: None.
Source: DFSMShsm
```

Figure 14-5 New message ARC0064E

14.3 Data recovery scenarios

Because data loss is always a possibility in computing, whether it is due to a user error or a software defect, ensure that data recovery procedures are in place in case data loss occurs.

The *DFSMShsm Storage Administration Guide,* SC26-0421, provides the following scenarios to assist you with your data recovery procedures if data loss occurs in DFSMShsm:

- Damaged CDS and full journal
- Damaged journal and undamaged CDS
- Full journal and undamaged CDS
- Structurally damaged CDS and missing journal records
- Time sequence gaps in CDS records and journal records also missing
- Overwritten migration tape
- Overwritten backup tape
- Damaged ML1 volumes
- Reestablish access to previously deleted migrated data sets (no backup exists, migration-level 2 (ML2) only)
- Correcting errors in the common recall queue
- Recovering a deleted ML1 data set without a backup

14.4 Auditing DFSMShsm

You can use the **AUDIT** command not only to detect and report discrepancies between CDSs, catalogs, and DFSMShsm-owned volumes, but also to diagnose and often provide repairs for these discrepancies. The **AUDIT** command is an effective tool for problem determination and recovery.

14.4.1 Introduction to the AUDIT command

To ensure data integrity, DFSMShsm uses numerous data set records to track individual data sets. These records are contained in the following places:

- Master catalog, which is a list of data sets for the entire system
- User catalog, which is a list of data sets that are accessible from that catalog
- Journal, which keeps a running record of backup and migration transactions
- Small-data-set packing (SDSP) data sets on migration volumes
- Migration control data set (MCDS), which is an inventory of migrated data sets and migration volumes
- Backup control data set (BCDS), which is an inventory of backed-up data sets and volumes, dumped volumes, and backed-up aggregates
- Offline control data set (OCDS), which contains a tape table of contents (TTOC) inventory of migration and backup tape volumes

In normal operation, these records stay in synchronization. However, because of data errors, hardware failures, or human errors, these records can become unsynchronized. The **AUDIT** command allows the system to cross-check the various records that relate to data sets and DFSMShsm resources. The **AUDIT** command can list errors and propose diagnostic actions or, at your option, complete certain repairs itself.

Consider the use of the **AUDIT** command for the following reasons:

- After any CDS restore (highly recommended)
- After an ARC184I message (error when reading or writing DFSMShsm CDS records)
- Errors on the RECALL or DELETE of migrated data sets
- Errors on BDELETE or RECOVER of backup data sets
- DFSMShsm tape-selection problems
- RACF authorization failures
- Power or hardware failure
- Consistency checking of fast replication CDS record relationships
- Periodic checks
- You can use the **AUDIT** command to cross-check the following sources of control information:
  - MCDS or individual migration data set records
  - BCDS or individual backup data set records, ABARS records, or fast replication records
  - OCDS or individual DFSMShsm-owned tapes
  - DFSMShsm-owned DASD volumes
  - Migration-volume records
  - Backup-volume records
  - Recoverable-volume records (from dump or incremental backup)
  - Contents of SDSP data sets
  - Fast replication dump, volume pair, version, and volume copy pool association records

Use the **AUDIT** command at times of low system activity because certain executions of the **AUDIT** command run a long time.

Original and enhanced audit commands

Two categories of audit commands exist: enhanced audit commands and original audit commands. The *enhanced audit commands* are designed to provide more comprehensive auditing than the original audit commands. However, the *original audit commands* are still available to provide compatibility with CLISTs or job streams that were developed for use with older versions of DFSMShsm. Because we recommend the use of the enhanced audit commands over the original audit commands, the remainder of this chapter focuses on the use of the enhanced audit commands.

Required parameters and optional parameters

When you issue the **AUDIT** command, you must specify one of the required parameters to indicate a primary category of control information. This control information will direct or "drive" the invocation of the audit. The optional parameters qualify what to audit, the output destination, and other qualifying information.

For an enhanced audit, you must choose one of the following required parameters to drive the audit:

- ABARSCONTROLS
- COMMONQUEUE
- COPYPOOLCONTROLS
- DATASETCONTROLS
- DIRECTORYCONTROLS VOLUMES
- MEDIACONTROLS
- VOLUMECONTROLS

Many optional parameters can be used when you execute an **AUDIT** command. For the list of optional parameters that apply to each required **AUDIT** parameter, see the *DFSMShsm Storage Administration Guide,* SC26-0421.

> **Important:** The two most important optional parameters are the **NOFIX** and **FIX** parameters.
>
> Use **NOFIX** if you want the **AUDIT** command to report any error that it detects and the actions it can take to fix them but to *not* take any of the fix actions.
>
> To fix the errors that the **AUDIT** command can fix, you must specify the **FIX** parameter on the **AUDIT** command.
>
> For planned use of the **AUDIT** command, we recommend that you first execute the **AUDIT** command with the **NOFIX** parameter. Then, after you review the errors that the **AUDIT** command detects and the actions that the **AUDIT** command can take to fix them, execute the **AUDIT** command with the **FIX** parameter.

Another important optional parameter is the **RESUME** parameter. Use the **RESUME** parameter only with the **AUDIT MEDIACONTROLS** and **DATASETCONTROLS** parameters. The **RESUME** parameter instructs DFSMShsm to start where it ended on a previous **AUDIT** command if that **AUDIT** command ended before completion for a non-abnormal reason. For example, if an **AUDIT DATASETCONTROLS(MIGRATION)** command is executing but causing contention that prevents other DFSMShsm tasks from executing, you might want to issue the **HOLD AUDIT** command. This command will stop the audit as soon as it finishes processing the current data set and allow the other DFSMShsm functions to proceed.

When the other DFSMShsm functions finish processing, you might want the audit to continue where the previous audit ended rather than start the audit from the beginning. To continue the previous audit, specify the **RESUME** parameter on the next issuance of the **AUDIT DATASETCONTROLS(MIGRATION)** command.

Note: The **AUDIT MEDIACONTROLS RESUME** command can be executed only against a tape and is only valid if the **FIX** parameter is specified. That is, both the **AUDIT** command that was ended prematurely and the **AUDIT** command that is issued to resume processing must specify the **FIX** parameter.

14.4.2 Auditing copy pools

DFSMShsm provides two methods of auditing the copy pools that it manages to ensure that the fast replication CDS record relationships are correct:

- COPYPOOLCONTROLS

  Audits all copy pool records
- COPYPOOLCONTROLS(*cpname*)

  Audits the copy pool records for a specific copy pool

If any issues are experienced during fast replication and you decide to execute an audit of the copy pool records to ensure that the fast replication CDS records exist and are accurate, be sure to execute both forms of this **AUDIT** command. The **AUDIT** command of all of the copy pools and the individual copy pool. These **AUDIT** commands execute different code paths and detect and fix different discrepancies in the fast replication CDS records.

14.4.3 Auditing common queues

Common queues have interrelated entries that can become corrupted due to abends or unexpected losses of connectivity. A corrupted common queue can cause certain requests to not be processed. DFSMShsm automatically corrects certain inconsistencies, but for other inconsistencies, it is necessary to issue the **AUDIT COMMONQUEUE** command. The **AUDIT** command enables DFSMShsm to dynamically correct inconsistencies with minimal impact on processing.

Consider the use of the **AUDIT COMMONQUEUE** command in the following situations:

- After receiving an ARC1506E message
- After receiving an ARC1187E message
- When recall requests are unexpectedly not being selected for processing

The internal structure of common queues is not externalized. An ARC1544I message is the only output that the **AUDIT COMMONQUEUE** command returns. It does not return a specific message for each error because individual error messages are of no value. The **OUTDATASET**, **SYSOUT**, and **TERMINAL** parameters are not used with **AUDIT COMMONQUEUE**.

Unlike other **AUDIT** command functions, auditing a common queue is not time-intensive and can be performed at any time.

14.4.4 AUDIT command examples

The following examples show how to instruct DFSMShsm to execute various **AUDIT** commands and the output that is produced when different errors are detected. For a full list of the errors that are detected by each **AUDIT** command, the fix actions that DFSMShsm can take to address those errors, and troubleshooting hints for those errors that DFSMShsm cannot fix on its own, see the *DFSMShsm Storage Administration Guide,* SC26-0421.

AUDIT DATASETCONTROLS(MIGRATION)

When you specify the **AUDIT DATASETCONTROLS(MIGRATION)** command, DFSMShsm uses each MCD record of a validly migrated data set in the MCDS to drive the audit. Example 14-21 shows that DFSMShsm detected two errors during this execution of the **AUDIT DATASETCONTROLS(MIGRATION)** command: ERR 03 and ERR 22.

Example 14-21 Output from the AUDIT DATASETCONTROLS(MIGRATION) command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 11:32:01 ON 12/08/21 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT DATASETCONTROLS(MIGRATION)ODS(DERDMANN.DSC.MIGOUT) NOFIX

/* ERR 03 DERDMANN.MIGTST1 NOT CATALOGED, HAS MIGRATION COPY + */
/* DFHSM.HMIG.T041811.DERDMANN.MIGTST1.A2234 */
/* ERR 22 DERDMANN.MIGTST2 HAS NO ALIAS RECORD */
/* FIXCDS A DFHSM.HMIG.T121811.DERDMANN.MIGTST2.A2234 CREATE(X'00000004' + */
/*
X'4040404040404040404040404040404040404040404040404040404040404040404040404040
40404040') */
/* FIXCDS A DFHSM.HMIG.T121811.DERDMANN.MIGTST2.A2234 PATCH(X'00000004'
DERDMANN.MIGTST2) */
/* FIXCDS A DFHSM.HMIG.T121811.DERDMANN.MIGTST2.A2234 PATCH(X'00000000' M) */
- END OF -     ENHANCED AUDIT - LISTING -
```

When DFSMShsm can fix the errors that it detects, one or more **FIXCDS** commands will be generated after the ERR number that is detected and before the next ERR number that is listed. In Example 14-21, only the ERR 22 can be repaired by DFSMShsm; ERR 03 cannot. Specifying the **FIX** parameter will apply the correct CDS updates to repair the ERR 22. To address the ERR 03, review the troubleshooting hints that are listed in the *DFSMShsm Storage Administration Guide,* SC26-0421.

AUDIT MEDIACONTROLS VOLUMES(volser)

When you specify the **AUDIT MEDIACONTROLS VOLUMES(*volser*)** command, DFSMShsm uses the information in the common data set descriptor (CDD), which is required at the start of every DFSMShsm-generated copy, in each DFSMShsm-generated copy of a user data set on the specified volume to drive the audit.

Example 14-22 shows that DFSMShsm detected one error during this execution of the **AUDIT MEDIACONTROLS VOLUMES(*volser*)** command: ERR 140.

Example 14-22 Output from the AUDIT MEDIACONTROLS VOLUMES(volser) command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 14:45:18 ON 12/08/21 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT MEDIACONTROLS VOLUMES(MIG102) ODS(DERDMANN.MC.OUT) NOFIX

/* ERR 140 MIG102 - DERDMANN.MIGTST2 HAS NO ALIAS RECORD */
/* FIXCDS A DFHSM.HMIG.T121811.DERDMANN.MIGTST2.A2234 CREATE */
- END OF -     ENHANCED AUDIT - LISTING -
```

The **FIXCDS** command that is identified after the ERR 140 indicates that the ERR 140 can be repaired by DFSMShsm. Specifying the **FIX** parameter will apply the correct CDS update to repair the ERR 140.

AUDIT VOLUMECONTROLS(MIGRATION)

When you specify the **AUDIT VOLUMECONTROLS(MIGRATION)** command, DFSMShsm uses the MCV records of all of the migration volumes in the MCDS to drive the audit. Example 14-23 shows that DFSMShsm detected one error during this execution of the **AUDIT VOLUMECONTROLS(MIGRATION)** command: ERR 56.

Example 14-23 Output from the AUDIT VOLUMECONTROLS(MIGRATION) command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 15:05:24 ON 12/08/22 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT VOLUMECONTROLS(MIGRATION) ODS(DERDMANN.VC.OUT2) NOFIX

/* ERR 56 MC1 EXTENSION 01 MISSING */
/* FIXCDS 1 L1VOL-00 VERIFY(X'00000001' BITS(1.......)) PATCH(X'00000001'
BITS(0.......)) */
- END OF -     ENHANCED AUDIT - LISTING -
```

The **FIXCDS** command that is identified after the ERR 140 indicates that the ERR 56 can be repaired by DFSMShsm. Specifying the **FIX** parameter will apply the correct CDS update to repair the ERR 56.

AUDIT COMMONQUEUE(RECALL)

When you specify the **AUDIT COMMONQUEUE(RECALL)** command, DFSMShsm scans the entries in the CRQ for logical inconsistencies in the structure.

Figure 14-6 shows the ARC1544I message that is issued only to the console. The ARC1544I indicates that DFSMShsm did not detect any errors in the CRQ.

```
13.52.42 SYSTEM1           f dfhsm,AUDIT COMMONQUEUE(RECALL) FIX
13.52.42 SYSTEM1 STC00036  ARC0300I **OPER** ISSUED===>AUDIT
 COMMONQUEUE(RECALL) FIX
13.52.42 SYSTEM1 STC00036  ARC1544I AUDIT COMMONQUEUE HAS COMPLETED, 0000
 ERRORS
ARC1544I (CONT.) WERE DETECTED FOR STRUCTURE SYSARC_PLEX1_RCL, RC=00
```

Figure 14-6 Output from the AUDIT COMMONQUEUE(RECALL) command

AUDIT COPYPOOLCONTROLS

When you specify the **AUDIT COPYPOOLCONTROLS** command, DFSMShsm reads sequentially through all existing fast replication records and confirms the existence and accuracy of all DFSMShsm records that are associated with copy pools. This command can detect orphaned records. Example 14-24 shows that DFSMShsm detected four errors during this execution of the **AUDIT COPYPOOLCONTROLS** command: ERR 180, ERR 178, and ERR 202.

Example 14-24 Output from the AUDIT COPYPOOLSCONTROLS command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 11:59:56 ON 12/09/04 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT COPYPOOLCONTROLS ODS(DERDMANN.AUDIT.CPC.AFTER2)

/* ERR 180 J RECORD NOT FOUND FOR VOLUME SRC003, IT BELONGS TO COPY POOL CP1 ,
VER=001*/
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME TAR001, EXPECTS FRSV (J) RECORD FOR
SOURCE VOLUME SRC003, WHICH WAS NOT FOUND */
/* ERR 202 ORPHANED I (FRTV) RECORD FOUND FOR TARGET VOLUME ABC123, SOURCE VOLUME
...... */
/* FIXCDS I ABC123 DELETE */
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME ABC123, EXPECTS FRSV (J) RECORD FOR
SOURCE VOLUME ......, WHICH WAS NOT FOUND */
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME TAR001, EXPECTS FRSV (J) RECORD FOR
SOURCE VOLUME SRC003, WHICH WAS NOT FOUND */
- END OF -     ENHANCED AUDIT - LISTING -
```

The **FIXCDS** command that is identified after the ERR 202 indicates that the ERR 202 can be repaired by DFSMShsm. Specifying the **FIX** parameter will apply the correct CDS update to repair the ERR 202. To address the ERR 180 and ERR 178, you must review the troubleshooting hints that are listed in the *DFSMShsm Storage Administration Guide,* SC26-0421.

AUDIT COPYPOOLCONTROLS(copypoolname)

When you specify the **AUDIT COPYPOOLCONTROLS(*copypoolname*)** command, the **AUDIT** command reads the fast replication control data set (CDS) records to confirm the existence and accuracy of all DFSMShsm records that are associated with the copy pool.

> **Note:** This form of the **AUDIT COPYPOOLCONTROLS** command cannot detect orphaned records.

Example 14-25 shows that DFSMShsm detected two errors during this execution of the **AUDIT COPYPOOLCONTROLS(*cpname*)** command: ERR 180 and ERR 178.

Example 14-25 Output from the AUDIT COPYPOOLCONTROLS(copypoolname) command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 12:01:28 ON 12/09/04 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT COPYPOOLCONTROLS(CP1) ODS(DERDMANN.AUDIT.CP1.OUTPUT)

/* ERR 180 J RECORD NOT FOUND FOR VOLUME SRC003, IT BELONGS TO COPY POOL CP1 ,
VER=001                                */
/* ERR 178 FRTV (I) RECORD FOR TARGET VOLUME TAR001, EXPECTS FRSV (J) RECORD FOR
SOURCE VOLUME SRC003, WHICH WAS NOT FOUND */
- END OF -     ENHANCED AUDIT - LISTING -
```

To address the ERR 180 and ERR 178, you must review the troubleshooting hints that are listed in the *DFSMShsm Storage Administration Guide,* SC26-0421.

AUDIT ABARSCONTROLS(agname)

When you specify the **AUDIT ABARSCONTROLS(*agname*)** command, DFSMShsm uses the ABR record for the aggregate that is specified to drive the audit. If you do not specify an aggregate, all of the aggregates in the BCDS are audited. Example 14-26 shows that DFSMShsm detected one error during this execution of the **AUDIT** command: ERR 168.

Example 14-26 Output from the AUDIT ABARSCONTROLS(agname) command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 14:26:14 ON 12/09/04 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT ABARSCONTROLS(DEREK) ODS(DERDMANN.AUDIT.ABARS.OUTPUT)

/* ERR 168 - CONTROL FILE NAME = DEREK.C.C01V0001 , + */
 INDICATED IN THE ABR RECORD, BUT IS NOT CATALOGED. ABR RECORD KEY =
DEREK.2012248000101 */
```

To address the ERR 168, you must review the troubleshooting hints that are listed in the *DFSMShsm Storage Administration Guide,* SC26-0421.

AUDIT DIRECTORYCONTROLS VOLUMES(volser)

When you specify the **AUDIT DIRECTORYCONTROLS VOLUMES(*volser*)** command, DFSMShsm uses the data set control blocks (DSCBs) on the specified volume with the attributes that DFSMShsm assigns to its data sets to drive the audit. Other DSCBs are ignored.

Example 14-27 on page 426 shows that DFSMShsm detected one error during this execution of the **AUDIT DIRECTORYCONTROLS VOLUMES(*volser*)** command: ERR 140.

Example 14-27 Output from the AUDIT DIRECTORYCONTROLS VOLUMES(volser) command

```
-DFSMSHSM AUDIT-       ENHANCED AUDIT -- LISTING - AT 15:40:58 ON 12/09/04 FOR
SYSTEM=3090
COMMAND ENTERED:
AUDIT DIRECTORYCONTROLS VOLUMES(MIG101) ODS(DERDMANN.AUDIT.DC.OUTPUT)

/* ERR 140 MIG101 - DFHSM.HMIG.T120115.DERDMANN.AUDIT.A2248 HAS NO ALIAS RECORD
/* AUDIT DATASETCONTROLS(MIGRATION) LEVELS(DERDMANN.AUDIT)
- END OF -     ENHANCED AUDIT - LISTING -
```

The output after the ERR 140 indicates that you must execute the **AUDIT DATASETCONTROLS(MIGRATION) LEVELS(DERDMANN.AUDIT)** command to further diagnose the ERR 140.

15

IBM Health Checker update

This chapter addresses a Data Facility Storage Management Subsystem (DFSMS) update that relates to health checks and also summarizes the available products for DFSMS and storage-related health checks.

15.1 Health Checker function

The IBM Health Checker examines the health of selected system components at regular intervals. It monitors settings, looks for single points of failure in the setup, and issues warnings that are based on its findings. The settings for the Health Checker are in the SYS1.PARMLIB member HZSPRMxx. To change the settings in the SYS1.PARMLIB member HZSPRMxx, use System Display and Search Facility (SDSF) statements or the **F hzsproc** command for dynamic updates. You can activate, deactivate, or update health checks by using the **F hzsproc** command.

Example 15-1 lists examples of operating the Health Checker.

Example 15-1 Examples of operating the Health Checker by command

```
S hzsproc /* Start health checker */
F hzsproc,RUN,CHECK=(ibmhsm,*) /* Run a group of health checks */
F hzsproc,RUN,CHECK=(ibmsms,sms_cds_reuse_option) /* Run individ. health check */
F hzsproc,UPDATE,CHECK=(ibmsms,sms_cds_reuse_option),severity=(med) /*Update */
```

The interval between checks is set by the **INTERVAL** parameter or by a policy. All health checks have a severity (LOW, MEDIUM, or HIGH) that is defined on the check. You can set this severity to suit your installation's needs. Currently, 500 health checks are available, but this section focuses only on the checks that relate to DFSMS.

15.1.1 New DFSMS health check

The device support health check on tape libraries was added to the z/OS Health Checker reporting on tape library errors that occur during IPL. This health check routine, DMO_TAPE_LIBRARY_INIT_ERRORS, was designed to include more device support health checks in the future. It is the only new health check that relates to DFSMS that was added in DFSMS V1.13. The report shows explanations and suggested remedies. For more information about this check, see the *IBM Health Checker for z/OS User's Guide,* SA22-7994. The following messages might be displayed in the report:

```
DMOH0101I CHECK(DMO_TAPE_LIBRARY_INIT_ERRORS) ran successfully and found no
exceptions.
DMOH0102I text.
DMOH0104E CHECK(DMO_TAPE_LIBRARY_INIT_ERRORS) determined that library device
initialization errors occurred during IPL.
DMOH0105I The check is not applicable in the current environment because there are
no ape libraries defined.
```

For more information, see Appendix A of *MVS System Messages and Codes, Volume 4,* SA22-7634.

Overview of current health checks that relate to DFSMS

The list of current DFSMS and storage-related health checks spans multiple areas:

- Catalog
- Tape libraries
- Data Facility Hierarchical Storage Management System (DFHSM)
- Data Facility Removable Media Manager (DFRMM)
- z/OS file system (zFS)
- Systems-managed buffering
- DFSMS

- VSAM
- VSAM record-level sharing (RLS)

The following list shows the currently available health checks:

- CATALOG_IMBED_REPLICATE
- DMO_TAPE_LIBRARY_INIT_ERRORS.
- HSM_CDSB_BACKUP_COPIES
- HSM_CDSB_DASD_BACKUPS
- HSM_CDSB_VALID_BACKUPS
- PDSE_SMSPDSE1
- ZOSMIGV1R10_RMM_REJECTS_DEFINED
- ZOSMIGV1R10_RMM_VOL_REPLACE_LIM
- ZOSMIGV1R10_RMM_VRS_DELETED
- ZOSMIGV1R11_RMM_DUPLICATE_GDG
- ZOSMIGV1R11_RMM_REXX_STEM
- ZOSMIGV1R11_RMM_VRSEL_OLD
- SMB_NO_ZFS_SYSPLEX_AWARE
- ZOSMIGREC_SMB_RPC.
- SMS_CDS_REUSE_OPTION
- SMS_CDS_SEPARATE_VOLUMES.
- VSAM_INDEX_TRAP
- VSAMRLS_CFCACHE_MINIMUM_SIZE
- VSAMRLS_CFLS_FALSE_CONTENTION
- VSAMRLS_DIAG_CONTENTION
- VSAMRLS_QUIESCE_STATUS
- VSAMRLS_SHCDS_CONSISTENCY
- VSAMRLS_SHCDS_MINIMUM_SIZE
- VSAMRLS_SINGLE_POINT
- ZOSMIGV1R11_ZFS_INTERFACELEVEL
- ZOSMIGREC_ZFS_RM_MULTIFS
- ZOSMIGV1R11_ZFS_RM_MULTIFS
- ZOSMIGV1R13_ZFS_FILESYS

For more information about health checks, see *IBM Health Checker for z/OS: User's Guide,* SA22-7994.

SMS health check function from SDSF

To access the IBM Health Checker from the TSO SDSF primary menu, enter `CK` from the command line or select **`CK`** from the displayed list (Figure 15-1 on page 430). SDSF is a separately priced program.

```
HQX7780 -----------------  SDSF PRIMARY OPTION MENU  --------------------------
COMMAND INPUT ===> CK SCROLL ===> CSR

DA    Active users                      INIT  Initiators
I     Input queue                       PR    Printers
O     Output queue                      PUN   Punches
H     Held output queue                 RDR   Readers
ST    Status of jobs                    LINE  Lines
                                        NODE  Nodes
LOG   System log                        SO    Spool offload
SR    System requests                   SP    Spool volumes
MAS   Members in the MAS                NS    Network servers
JC    Job classes                       NC    Network connections
SE    Scheduling environments
RES   WLM resources                     RM    Resource monitor
ENC   Enclaves                          CK    Health checker
PS    Processes
                                        ULOG  User session log
END   Exit SDSF
```

Figure 15-1 Accessing Health Checker with CK

If you want to filter by CheckOwner, enter `FILTER CHECKOWNER EQ IBMDMO`. Figure 15-2 shows the new health check on tape libraries on the startup after the IPL.

```
SDSF HEALTH CHECKER DISPLAY  (ALL)                  LINE 44-61 (489)
COMMAND INPUT ===>                                         SCROLL ===> CSR
PREFIX=ANDRE*  DEST=(ALL)  OWNER=*  SYSNAME=*
NP   NAME                            CheckOwner       State              Status
S    DMO_TAPE_LIBRARY_INIT_ERRORS    IBMDMO           ACTIVE(ENABLED)     SUCCE
     GRS_AUTHQLVL_SETTING            IBMGRS           ACTIVE(ENABLED)     EXCEP
     GRS_CONVERT_RESERVES            IBMGRS           ACTIVE(DISABLED)    GLOBA

.......... more health checks ...
```

Figure 15-2 New health check on tape libraries on the startup after the IPL

In Figure 15-2, note the headings: Name, CheckOwner, State, and Status. You can also scroll to the right for more headings. The health checks are grouped as shown in the column CheckOwner. You can activate or run a health check with a command against the individual health check or group. You can verify at the far right side that this health check ran successfully. Select the check DMO_TAPE_LIBRARY_INIT_ERRORS by entering `S` under the heading NP. Figure 15-3 on page 431 shows the verification results.

```
SDSF OUTPUT DISPLAY DMO_TAPE_LIBRARY_INIT_ERRORS  LINE 0      COLUMNS 02- 81
 COMMAND INPUT ===>                                          SCROLL ===> CSR
********************************* TOP OF DATA**********************************
CHECK(IBMDMO,DMO_TAPE_LIBRARY_INIT_ERRORS)
START TIME: 09/21/2011 12:42:11.448711
CHECK DATE: 20100128  CHECK SEVERITY: LOW

DMOH0101I CHECK(DMO_TAPE_LIBRARY_INIT_ERRORS) ran successfully and found
no exceptions

END TIME: 09/21/2011 12:42:11.449351  STATUS: SUCCESSFUL
```

Figure 15-3 Results of health check verification

To demonstrate a health check with exceptions, select **SMS_CDS_SEPARATE_VOLUMES**, as shown in Figure 15-4. In this case, an exception was identified and explained, and a recommendation to improve the setup was offered.

```
CHECK(IBMSMS,SMS_CDS_SEPARATE_VOLUMES)
START TIME: 10/06/2011 13:50:46.469308
CHECK DATE: 20090303  CHECK SEVERITY: MEDIUM

* Medium Severity Exception *

IGDH1001E CHECK(IBMSMS,SMS_CDS_SEPARATE_VOLUMES) detected the
ACDS (SYS1.SMS.ACDS)
and COMMDS (SYS1.SMS.COMMDS)
allocated on the same volume.

  Explanation:  As a best practice, an ACDS/COMMDS must reside on a
    volume, accessible from all systems in the SMS complex. To ease
    recovery in case of failure, the ACDS should reside on a different
    volume than the COMMDS. Also, you should allocate a spare ACDS on a
    different volume. The control data set (ACDS or COMMDS) must reside
    on a volume that is not reserved by other systems for a long period
    of time because the control data set (ACDS or COMMDS) must be
    available to access for SMS processing to continue.
```

Figure 15-4 Health check with exceptions

15.1.2 Upgrade and coexistence considerations

No particular upgrade action is required. The IBM Health Checker is an established, standard product that adds new health checks automatically. The follow-up can be automated through your automation package, which is based on the severity settings that you choose for your environment. For more information about receiving email alerts, see *Exploiting the IBM Health Checker for z/OS Infrastructure*, REDP-4590.

16

Common DFSMShsm functions

In this chapter, we describe functions that are common to multiple areas of DFSMShsm, which allows us to document these functions in one place only.

16.1 Changing priorities in DFSMShsm

In this section, we describe changing priorities in DFSMShsm.

16.1.1 Priority processing

You might see NOWAIT recalls and deletions queue up because many of these NOWAIT recalls and deletions were submitted at the same time. These NOWAIT recalls and deletions end up in the same DFSMShsm Management Work Elements (MWEs) queue. DFSMShsm will give all of the WAIT requests higher priority than the NOWAIT requests. All users will start one NOWAIT request before another user starts a second. The prioritization sequence will be from the oldest to the newest request.

Recalls from tape are an exception. When a migration-level 2 (ML2) tape is mounted, all recall requests that are going to this tape will be processed before this tape is unmounted. Recalls from the tape are in prioritization order and continue on the condition that the TAPERECALLLIMITS value is not exceeded.

When the TAPERECALLLIMITS value is reached and a high priority recall toward the same tape is pending on another system, the recall process in this tape will end, even though not all recalls were processed. If no higher priority recalls are in the queue, recalls will continue on the tape that is mounted. However, a check is made after each recall to see whether a new, higher-priority recall appeared.

Using the ARCRPEXT

The Return Priority installation exit can be installed to change priority when requests are added to the queue. Prioritization can be performed for WAIT requests relative to other WAIT requests and similarly for NOWAIT requests. You can also give recall requests a higher priority than delete requests. In the same way, you can give interactive requests a higher priority than batch-initiated requests.

DFSMShsm will prime the priority field with the value of `50`. With the ARCRPEXT, you can change this value to a value 0 - 100 (100 is the highest priority). You can change this value for both WAIT and NOWAIT requests. Two requests with the same priority will be processed in first-in first-out (FIFO) order.

For RECALL and DELETE requests, “priority order” on the recall queue has the following meaning:

- WAIT requests are queued before NOWAIT requests.
- Priorities of WAIT requests can be relative to other WAIT requests.
- Priorities of NOWAIT requests are relative to other NOWAIT requests (but after WAIT requests):
  - NOWAIT requests (if any) with a relative priority of *50* are interleaved by the user ID of the submitter with other NOWAIT requests of priority 50.
  - NOWAIT requests with a relative priority *greater than 50* are queued before those NOWAIT requests with a priority of 50, without interleaving.
  - NOWAIT requests with a relative priority *less than 50* are queued after those NOWAIT requests with a priority of 50, without interleaving.

You can assign priority values other than 50 for those NOWAIT requests that you do not want interleaved by user ID.

For data set RECOVER requests, the following prioritization applies:

- WAIT requests are queued before NOWAIT requests.
- WAIT requests can be prioritized relative to other WAIT requests.
- NOWAIT requests can be prioritized after other NOWAIT requests, but after WAIT requests.

For volume recover requests, the following priority sequence on the queue applies:

- WAIT requests are queued ahead of NOWAIT requests.
- WAIT requests can be prioritized relative to other WAIT requests.
- NOWAIT requests can be prioritized relative to other NOWAIT requests, but after WAIT requests.

16.1.2 Tape Take Away in a single DFSMShsm host environment

The term *Tape Take Away* in DFSMShsm is used to cover the situation where a long-running function is preventing a request from being processed, for example, an ML2 recall from being processed because the tape request for the recall is used by a long-running function.

Tape Take Away can be requested by the recall function, and Tape Take Away occurs fast and smoothly. (The tape is released for the recall.)

For migration and recycle output, a tape might be needed that is in use by a long-running function. In this case, migration or recycle will select a different tape for output.

If a recall needs a migration volume, Tape Take Away will start and the migration task will close the requested volume and continue on a different output volume. This situation is valid for any type of migration: primary, secondary, interval, and command-initiated migration.

Tape Take Away for recalls can also occur for long-running tasks, such as RECYCLE and TAPECOPY input tapes.

Recycle occurs by releasing the connected set that is being processed and by moving on to the next set. TAPECOPY is slightly different because a time gap exists before Take Away occurs and TAPECOPY might finish successfully within this time.

Contention is also possible with recall itself. If many recalls are being processed, a higher priority recall might suffer, due to missing available recall tasks. In this case, Tape Take Away can also be scheduled.

For an ABACKUP contention with recall, two scenarios exist. If the recall is type NOWAIT, Tape Take Away does not occur. If the recall request type is WAIT, the Tape Take Away starts only after 10 minutes, which leaves the ABACKUP task a chance to finish before the tape is requested by recall.

For ARECOVER, the recall request will not schedule a Tape Take Away. It will try to execute the recall only within the specified limit for retries.

In general, if a tape is not freed up after 15 retries, DFSMShsm will send message ARC0380A to the operator console with the options of responding: WAIT, CANCEL, or MOUNT.

The WAIT option will reset the count and a new series of 15 retries can start with a two-minute interval.

ABACKUP can also invoke a Tape Take Away from RECYCLE or TAPECOPY input tapes. For RECYCLE, the approach will be as for a recall Tape Take Away: Recycle will end on the current connected set and will move on to the next one. For TAPECOPY, a time delay occurs before Take Away occurs, leaving TAPECOPY the opportunity to finish before the tape is taken away.

Tape Take Away in a multiple DFSMShsm host environment

The Tape Take Away process is similar in a single DFSMShsm host and a multi-DFSMShsm environment. Therefore, this section will explain only the difference.

If a recall request to an ML2 tape is requested, and this ML2 tape is already in use by another DFSMShsm host, TAPERECALLLIMITS sets the time value for when a Tape Take Away can occur. Tape Take Away will only occur however if the upcoming recall has a higher priority than the one already in process. The behavior differs in a common recall queue (CRQ) environment, where all recalls in the CRQ queue to the same tape occur in the same tape mount.

If a RECYCLE requests an input or output tape that is being used in a RECYCLE on a different DFSMShsm host, the request will fail.

16.1.3 Altering priority by using the ALTERPRI command

There is a new DFSMShsm command, **ALTERPRI**, to alter the priority of queued requests.

You can alter the priority of the following request types:

- ABACKUP
- ARECOVER
- BACKDS
- BACKVOL
- DELETE
- FRBACKUP
- FREEVOL
- FRRECOV
- MIGRATE
- RECALL
- RECOVER

Use the **ALTERPRI** command to alter the priority of queued requests on an as-needed basis. You must not use this command as the primary means of assigning priority values to new requests.

Two options of setting priority are available with the **ALTERPRI** command:

- The **HIGH** parameter, which is the default, alters the specified request so that it has the highest priority on its respective queue.
- Conversely, the **LOW** parameter alters the request so that it has the lowest priority on its respective queue.

If you do not give any parameter, **HIGH** will be the default.

> **Note:** You cannot alter the priority of **BACKVOL CDS** commands and requests that are already selected for processing.

The mutually exclusive **REQUEST**, **USER**, or **DATASETNAME** parameters indicate the requests that DFSMShsm will reprioritize. Use the **QUERY REQUEST** command to determine the request number to issue on the **ALTERPRI** command. DFSMShsm reprioritizes all queued requests that match the REQUEST, USERID, or DATASETNAME criteria that are specified on the **ALTERPRI** command.

To reprioritize a recall request on the CRQ, issue the **ALTERPRI** command on the same host that originated the recall request. The syntax for the command is shown in Example 16-1.

Example 16-1 and Figure 16-1 provide examples of changing priority on DFSMShsm requests.

Example 16-1 Changing the DFSMShsm priority by using the ALTERPRI command

```
ALTERPRI DATASETNAME(USERA.TEST01) /* default used = HIGH */
ALTERPRI REQUEST(nnnn) HIGH
ALTERPRI USERID(USERA) LOW
```

The possible (optional) values of priority setting are HIGH or LOW (`HIGH` is the default).

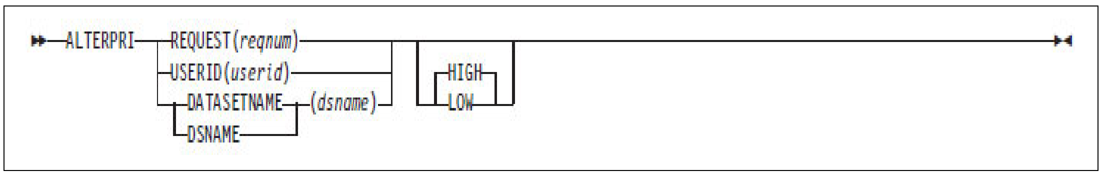

Figure 16-1 ALTERPRI command syntax

Setting these values will move a request to the highest priority on the queue (if HIGH is requested) or to the lowest priority on the queue (if LOW is requested).

Certain commands will generate multiple requests with the same request number:

- BACKVOL STORAGEGROUP
- BACKVOL VOLUMES
- FRBACKUP DUMP
- FRBACKUP DUMPONLY
- FRRECOV DSNAME

If you change the priority on one of these requests by request number, all requests will get the new priority.

Example of changing priority by using ALTERPRI

In this scenario, we show a DFSMShsm recall queue with a number of data sets that are waiting. The lowest priority data set recall is at the bottom of the list in Example 16-2.

Example 16-2 QUERY REQUEST on a DFSMShsm recall queue

```
-F DFHSM64,Q REQUEST
STC20554 ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=2
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#1,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000211, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000000 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#2,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000212, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000001 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#3,
```

```
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000213, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000002 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.TCPIP.PROFILE ,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000214, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000003 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
```

The lowest priority recall is marked in bold in Example 16-3. Now, ALTERPRI is executed with the REQUEST number to change the priority to HIGH (equivalent to moving the recall request to the top of the queue). See the command in Example 16-3.

Example 16-3 Change priority by using ALTERPRI on the lowest priority recall

```
F DFHSM64,ALTERPRI REQUEST(00000214) HIGH
ARC0980I ALTERPRI REQUEST COMMAND STARTING
ARC0982I RECALL MWE FOR DATA SET MHLRES4.TCPIP.PROFILE 546
ARC0982I (CONT.) FOR USER MHLRES4, REQUEST 00000214, REPRIORITIZED TO
ARC0982I (CONT.) HIGH
ARC0981I ALTERPRI REQUEST COMMAND COMPLETED, RC=0000
```

The priority is successfully changed as we can see from the message in the command output. Performing a new **QUERY REQUEST** command also confirms that the request number that was changed moved to the top of the list. See the output from the query in Example 16-4.

Example 16-4 QUERY REQUEST that shows the changed priority after the ALTERPRI command

```
-F DFHSM64,Q REQUEST
STC20554 ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=2
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.TCPIP.PROFILE ?,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000214, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000000 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#1,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000211, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000001 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#2,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000212, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000002 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC1543I RECALL MWE FOR DATASET MHLRES4.DUMP.OUT.#3,
ARC1543I (CONT.) FOR USER MHLRES4, REQUEST 00000213, WAITING TO BE
ARC1543I (CONT.) PROCESSED ON A COMMON QUEUE,00000003 MWES AHEAD OF
ARC1543I (CONT.) THIS ONE
STC20554 ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=2
```

Protecting an ALTERPRI command

Each storage administrator command can be protected through the following fully qualified discrete FACILITY class profile: STGADMIN.ARC.*command*.

In this case, a security administrator can create the fully qualified, discrete profile to authorize this command to storage administrators. You can obtain more details, such as the entire command list and specific RACF profiles, in the *DFSMShsm Implementation and Customization Guide*, SC35-0418.

17

Hints, tips, and suggested practices

In this chapter, we provide useful hints and tips, and recommend the preferred practices for implementing DFSMShsm and the daily administration of DFSMShsm.

17.1 Hints and tips

We provide hints and tips that were useful in our environment.

17.1.1 Use REGION=0M when you start DFSMShsm

Specify 0M for SIZE, which directs DFSMShsm to allocate the largest possible region size. This specification reduces the chance that DFSMShsm will end abnormally because of an out-of-space condition (ABEND878).

17.1.2 Large format journal

If you must back up the control data sets (CDSs) more than once a day due to the journal filling up, consider allocating the journal as a large format sequential data set. Because the journal can be allocated only on a single volume, allocating the journal as a large format data set allows the journal to grow beyond 65535 tracks in size on that volume. Allocating the journal as a large format data set allows more DFSMShsm activity to take place between journal backups and helps to avoid journal full conditions.

17.1.3 CDS recovery

You do not want to be unprepared to recover one or more CDSs if one or more of the CDSs is corrupted. The *DFSMShsm Storage Administration Guide,* SC26-0421, contains step-by-step instructions in the enhanced CDS recovery function section to guide you through the complete recovery of a CDS from a backup copy. We recommend that you execute these steps on a test system to become familiar with the enhanced CDS recovery function if you need to use the procedure to recover from an actual corrupted CDS condition.

17.1.4 Delete migrated data sets without unnecessary recall

When users are deleting data sets with DFSMShsm, users can now bypass the unnecessary recall for |PGM=IEFBR14 job steps by updating the ALLOCxx PARMLIB member to include a new statement, `SYSTEM IEFBR14_DELMIGDS(NORECALL)`, or by using the new **`SETALLOC`** command to enable no recall.

17.1.5 Using ARCCATGP

Issuing the **`UNCATALOG`**, **`RECATALOG`**, or **`DELETE NOSCRATCH`** command against a migrated data set causes the data set to be recalled before the operation is performed. You can authorize certain users however to issue these commands without recalling the migrated data sets by connecting the user to the RACF group ARCCATGP. When a user is logged on under RACF group ARCCATGP, DFSMShsm bypasses the automatic recall for UNCATALOG, RECATALOG, and DELETE NOSCRATCH requests for migrated data sets.

The following tasks are used to enable DFSMShsm to bypass automatic recall during catalog operations:

- Define RACF group ARCCATGP by using the following RACF command:

  `ADDGROUP (ARCCATGP)`

- Connect users who need to perform catalog operations without automatic recall to ARCCATGP by using the following RACF command:

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

- Each user who needs to perform catalog operations without automatic recall must log on to TSO specifying the **GROUP(ARCCATGP)** parameter on the TSO logon panel or the **GROUP=ARCCATGP** parameter on the JOB statement of a batch job. See Example 17-1.

Example 17-1 How to code a jobcard by using ARCCATGP

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

17.1.6 DFSMShsm using 3592 model E05

The following hints and tips are imported from the book: *DFSMS Software Support for IBM System Storage TS1130 and TS1120 tape drives (3592)*, SC26-7514:

- In a non-storage management subsystem (SMS) mixed tape hardware environment, where multiple types of tape hardware are used to emulate 3590 devices, it is recommended that you define unique esoterics for each type of tape hardware. This action is necessary to avoid mixing incompatible recording technologies. You can define an esoteric to DFSMShsm through the **SETSYS USERUNITTABLE** command, for example: **SETSYS UUT(3592E05:3592E05 3590H:3590H)**. With esoterics defined, you can then direct output to the set of drives that you want through the **SETSYS** command, for example: **SETSYS BACKUP(TAPE(3592E05))**
- If your installation has an excessive number of spanning data sets, consider specifying a larger value in the **SETSYS TAPESPANSIZE** command. A larger absolute value is needed to represent the same amount of unused capacity on a percentage basis when the tape has a larger total capacity. For example, if you allow 2% of unused tape to avoid tape spanning for a 3590-H*xx* device that uses enhanced media, specify a TAPESPANSIZE of 1200 MB. To allow 2% unused tape for a MEDIA5 tape on a 3592 Model E05 device (no performance scaling), specify a TAPESPANSIZE of 9999 MB. All size calculations for scaled tapes are based on the scaled size and not the unscaled size.
- If the speed of data access on MEDIA5 or MEDIA9 tape is more important than the full use of the capacity, consider the use of performance scaling. Performance scaling uses 20% of the physical capacity on each tape and keeps all data sets closer together and closer to the initial tape load point. If you use performance scaling with the DFSMShsm duplex function, be sure that the original tape and the alternate tape both use performance scaling. Similarly, make sure that tapecopy input tapes and output tapes have the same performance scaling attributes.

Note: Performance scaling is not available on these tape cartridge media: MEDIA6, MEDIA7, MEDIA8, and MEDIA10. If your installation is using MEDIA5 tapes with performance scaling, consider the use of MEDIA7 tapes for high performance functions. The available MEDIA5 tapes can then be used to their full capacity. Consider performance segmentation as a compromise solution. Performance segmentation increases the performance of data sets in the first 20% of the tape's capacity, but also uses the remaining capacity as a slower access segment. The average performance for the tape is increased at the expense of losing a percentage of the MEDIA5 or MEDIA9 overall tape capacity. (You cannot determine which data sets reside in which segment.)

DFSMShsm recycle processing of 3592 Model E05 tapes can take significantly longer than with smaller tapes because the amount of data that is moved at the same RECYCLEPERCENT can be much larger. In addition to moving more data, the likelihood of a tape takeaway for recall processing increases with the number of data sets that still remain on the tape. One option for controlling overall recycle run time is the **LIMIT(*nnnn*)** parameter of recycle. Recycle returns no more than the specified number of tapes to scratch during the current recycle run. Because recycle sorts the tapes based on the amount of valid data still on each volume, the tapes that are recycled require the least processing time.

Another option to consider is decreasing the **ML2RECYCLEPERCENT** parameter, the **RECYCLEPERCENT** parameter, or both. Assume, for example, that your installation uses MEDIA7 tape for migration-level 2 (ML2) and MEDIA5 tape for backup. If the EFMT1 format is used and you want no more than 6 GB of data to be moved when an ML2 tape is recycled, set ML2RECYCLEPERCENT(10) because the MEDIA7 can hold 60 GB of data in EFMT1. If your installation uses full capacity for backup tapes and you want no more than 6 GB of data to be moved when a backup tape is recycled, set RECYCLEPERCENT(2) because a MEDIA5 tape can hold 300 GB of data in EFMT1.

These examples assume that the ML2 and backup tapes in the installation are filled to capacity because the calculations are based on the average fullness of marked full tapes on your system (the reuse capacity). To determine how much data your current recycle threshold implies, use the reuse capacity that is associated with the tapes. The current recycle threshold percent multiplied by the reuse capacity gives the maximum amount of data on any of the tapes when they are recycled. Although lowering the recycle threshold reduces recycle processing time and decreases the number of times that each tape must be recycled, it might also increase the overall number of tapes that are needed in your installation. Also, if you have a mix of ML2 or backup tape capacities in need of recycle processing, you might want to recycle tapes with the **RANGE** parameter and use the appropriate recycle threshold for the tape capacities in the range.

In a storage management subsystem (SMS) tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select Write Once Read Many (WORM) tapes for ABACKUP processing. The output data set prefix that is specified in the aggregate group definition can be used by the automatic class selection (ACS) routines to select a WORM data class. Set up the ACS routine and the output data set name to uniquely identify the aggregate backup and recovery support (ABARS) output files that must go to WORM tape. In a non-SMS tape environment, the default allows tape pooling to determine whether ABARS data sets go to WORM or read/write media.

Optionally, if the **DEVSUP** parameter **ENFORCE_DC_MEDIA=ALLMEDIATY** or **ENFORCE_DC_MEDIA=MEDIA5PLUS** is used, the data class must request the appropriate media type for it to be successfully mounted:

- Consider using the Fast Subsequent Migration function to reduce the need to RECYCLE these high-capacity tapes.
- For a sysplex environment, consider the use of the common recall queue (CRQ) to optimize mounts of migration tapes.
- AUDIT MEDIACONTROLS for a FAILEDCREATE situation usually needs to look at only the last few files on a tape. If it is available for your system level, ensure that Audit APAR OA04419 is applied.
- The 3592 Model E05 tape drive is used in 3590 emulation mode only, never 3490. The 3592 Model J1A can operate in 3490 emulation mode only when it uses MEDIA5 for output.

How to activate performance segmentation or scaling

The following tasks must be complete to activate performance segmentation or scaling:

- A data class must exist with the *performance segmentation* or *performance scaling* attribute set to Y.
- The ACS routine must map a DFSMShsm single file tape data set name to the data class.
- Because data class now determines whether a performance option is used, non-SMS tape needs ACS routines if you want a 3592 performance option.

IEC205I at close of tape will indicate whether performance segmentation or scaling was used. Performance segmentation and performance scaling apply to MEDIA5 only.

Note: Performance segmentation and performance scaling are mutually exclusive.

Reorganizing your CDSs

Reorg with FREESPACE(0 0) and let DFSMShsm split midsection intervals:

- Performance is degraded for about 2 - 3 weeks during this process.
- Do not panic when you see the HURBA/HARBA ratio increase during first several days.

Execute the REORG job only when you are increasing size allocation. Ensure that *all* hosts are shut down before you attempt to reorganize any CDSs. Use DISP=OLD on a REORG job to prevent accidentally bringing up DFSMShsm.

Error handling in the CRQ environment

If a z/OS image fails, the DFSMShsm host on that system also fails, but all of the recall requests that originated on that host remain intact on the CRQ:

- The coupling facility notifies the remaining connected hosts of the failure.
- In-process requests on the failed host remain on the queue and are available for other hosts to restart them.

  If the failing host was processing a request from an ML2 tape, recall requests for data on that tape cannot be selected.
- The tape is marked as “in-use” by the failing host. The “in-use” indicator can be reset by restarting the failed host or by using the **LIST HOSTID(*hostid*) RESET** command.

17.1.7 Experiencing a space problem during the recall process

If you are experiencing difficulty finding space to recall a large data set, you have two options. You can force it to go multivolume:

```
HSEND RECALL datasetname DFDSSOPTION(VOLCOUNT(ANY))
```

Or, you can direct it to an empty, non-SMS disk:

```
HSEND RECALL datasetname FORCENONSMS UNIT(3390) VOLUME(nonSMSvol)
```

17.1.8 Replacing a lost or damaged ML2 tape

Hopefully, you duplex your ML2 tapes? First, identify the alternate volume with the following **LIST TTOC** command:

```
HSEND LIST TTOC (volser) DSI TERM
```

Then, from the TTOC listing, check that this tape is not part of a spanned set. If this tape is part of a spanned set, find the spanned data set by running a **LIST TTOC** command on the previous volume and recall the spanned data set.

Check that the "link" is broken by repeating the **LIST TTOC (*volser*)** command for the original volume. It now shows no previous volume if the link is broken.

You can then replace the broken cartridge with an alternate cartridge, with the **TAPEREPL** command:

```
HSEND TAPEREPL ORIGINALVOLUMES(volser) ALTERNATEUNITNAME(esoteric)
```

The *esoteric* is the correct tape unit for your alternate ML2 tapes.

17.1.9 ML1 pool full

If the ML1 pool is full, consider adding more volumes to the migration-level 1 (ML1) pool, empty out older data, or move out the larger files.

The following command will move data that is one day old or older on ML1 to ML2:

```
MIGRATE MIGRATIONLEVEL1 DAYS(1)
```

You can also use the following JCL to generate **MIGRATE** command cards for any data set that fits the selection criteria, as shown in Example 17-2.

Example 17-2 Sample JCL to move data sets from ML1 to ML2

```
//MHLRESAR JOB (ACCOUNT),'NAME',CLASS=A,MSGCLASS=X
//*
//*  LIST DATASETS ON ML1 AND MOVE THEM TO ML2
//*
//STEP01  EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//MCDS     DD  DSN=HSM.MCDS,DISP=SHR
//DCOLLECT DD  DSN=&&DCOLDATA,UNIT=SYSALLDA,SPACE=(CYL,(150,100)),
//             DISP=(NEW,PASS),DCB=(LRECL=700,RECFM=VB,DSORG=PS)
//SYSIN    DD  *
 DCOLLECT -
         OFILE(DCOLLECT) -
         MIGRATEDATA
```

```
/*
//DSNINFO EXEC PGM=ICETOOL,REGION=4M
//TOOLMSG  DD  SYSOUT=*
//DFSMSG   DD  SYSOUT=*
//DCOLLECT DD  DISP=OLD,DSN=&DCOLDATA
//TYPED    DD  DISP=(,PASS),SPACE=(TRK,(10,5)),UNIT=SYSALLDA,
//             DSN=&&DCOLTYPD
//SORTED   DD  DISP=(,CATLG),SPACE=(TRK,(2,1)),UNIT=SYSALLDA,
//             DSN=MHLRESA.ML12ML2,
//             DCB=(LRECL=80,RECFM=VB,BLKSIZE=0,DSORG=PS)
//TOOLIN   DD  *
 COPY FROM(DCOLLECT) TO(TYPED) USING(CPY1)
 SORT FROM(TYPED) TO(SORTED) USING(SRT1)
/*
//CPY1CNTL DD  *
*                   ML1 DATASET
*                          |
 SORT FIELDS=COPY
 INCLUDE COND=(74,1,CH,NE,C'T',AND,
*              ALLOCATE SPACE IS MORE THAN 1MB
               77,4,BI,GT,X'00001000',AND,
*              MIG DATE IS MORE THAN 4 DAYS AGO
               85,4,PD,LT,DATE3P-4)
 RECORD TYPE=V
 OPTION VLSHRT
/*
//SRT1CNTL DD  *
* SORT BY SPACE ALLOCATION
 SORT FIELDS=(93,4,BI,D)
 OUTREC  FIELDS=(1,4,C' HSEND WAIT MIG DSN(''',29,44,C''') ML2')
 RECORD TYPE=V
/*
```

In the sample JCL in Example 17-2 on page 444, we use DCOLLECT on migration data and select only any data set that is on an ML1 volume, the migration date is older than four days, and also, its space allocation is larger than 1 MB.

17.1.10 Identify ML1 volumes

ML1 volumes are DFSMShsm-managed if those ML1 volumes are defined and exist in the ARCCMDxx parmlib by using the **ADDVOL** command at the time that DFSMShsm starts up.

If you added more volumes to the ML1 pool dynamically and forgot to define them in ARCCMDxx, they cannot be used until after the next DFSMShsm startup.

> **Important:** The **LIST ML1** command is based on the migration control data set (MCDS) to extract the information. This command gives you the false sense that those listed ML1 volumes are actually in use, but in fact they might not be DFSMShsm-managed at all.

You must use the **QUERY SPACE(*volser*)** command to actually discover whether a particular ML1 volume is DFSMShsm-managed, as shown in Example 17-3 on page 446.

Example 17-3 QUERY SPACE command output

```
ARC0400I VOLUME SBXHS4 IS 97% FREE, 0000000042 FREE TRACK(S), 000009798 FREE
ARC0400I (CONT.) CYLINDER(S), FRAG .014
ARC0401I LARGEST EXTENTS FOR SBXHS4 ARE CYLINDERS      9507, TRACKS     142605
ARC0402I VTOC FOR SBXHS4 IS 00090 TRACKS(0004500 DSCBS), 0004483 FREE
ARC0402I (CONT.) DSCBS(99% OF TOTAL)
***
```

If the ML1 volume is not DFSMShsm-managed, you get a message that indicates that the ML1 volume is not DFSMSHSM-managed, as shown in Example 17-4.

Example 17-4 QUERY SPACE command message

```
ARC0407I QUERY SPACE FAILED, VOLUME SBXHS5 NOT CURRENTLY MANAGED BY DFSMSHSM
***
```

17.1.11 RECALL failures on ML1 volumes

The following steps correct RECALL failures on ML1 volumes:

1. List the data set by using TSO 3.4 and issue the **HLIST DSN(/) MCDS TERM** command to find the migration volume, migration date, and small-data-set packing (SDSP), if used.
2. Look at the migration volume through TSO 3.4 to find the HSM name, by using the migration date to find I*nnnn* (where *nnnn* is the Julian date).

> **Note:** The HSM name will be 'prefix.HMIG.T*nnnnnn.xx.yyyyyy*.I*nnnn*' where T*nnnnnn* is the migration time backwards.

3. Issue the **HSEND FIXCDS D / DISPLAY ODS('HSM.HMIG.T*nnnnnn.xx.yyyyyy*.I*nnnn*')** command.
4. Issue the **HSEND FIXCDS A / DISPLAY ODS('HSM.HMIG.T*nnnnnn.xx.yyyyyy*.I*nnnn*')** command.
5. Run an **IEHLIST** command to print a dump of each record.
6. Run a DSS dump of HSM.HMIG.T*nnnnnn.xx.yyyyyy*.I*nnnn.*
7. After all diagnostics are taken, list the data set by using TSO 3.4 and issue the **HSEND FIXCDS D / DELETE** command. Run this command to delete the migration control record. Run the **HLIST DSN(/) MCDS TERM** command to check that the entry was deleted.
8. Issue the **DELETE/NOSCRATCH** command to delete the catalog entry.
9. Try to recover data from a backup.

17.1.12 Dump stacking and a storage group

If you are planning to use the full volume dump with stacking feature, it is better to group all volumes with the same volume capacity in the same storage group and assign DUMPCLASS with the appropriate stack number.

For example, if you set up a pool of 100 volumes to have full volume dump, which consists of 80 model 9 volumes and 20 model 27 volumes, dumpclass is defined with DUMPSTACK=20, and also MAXDUMPTASKS is 5. DFSMShsm might possibly assign all or most of 20 volumes of model 27 into one dump task, and divide the other volumes evenly to the remaining four dump tasks. As a result, it will take longer to finish the dump for all 20 model 27 volumes. Therefore, auto dump might not complete within its defined window.

Alternatively, if you put all of the 80 model 9 volumes into one storage group and assign DUMPCLASS=FDVMOD9 with DUMPSTACK=20, and put all of the 20 model 27 volumes into another storage group and assign DUMPCLASS=FDVMOD27 with DUMPSTACK=5, each of the dump tasks will either get 20 model 9 volumes or 5 model 27 volumes. Therefore, the dump window is shorter.

17.1.13 Schedule automatic backup in multiple DFSMShsm host environments

Automatic backup is performed in four phases:

- Backing up the CDSs
- Moving backup versions from ML1 to tape
- Backing up migrated data sets
- Backing up DFSMShsm-managed volumes from a storage group with the automatic backup attribute

Only the PRIMARY HOST performs the first three phases. Therefore, it is best to schedule auto backup for the primary host earlier than for the rest of the hosts, which allows the CDSs to be backed up, and the other processors do not have to wait for the CDSs. Another tip is to run automatic backup before space management so that backup versions that are created with the **BACKDS** command are moved off the ML1 volumes where they temporarily reside.

17.1.14 Schedule primary space management and secondary space management

Secondary space management (SSM) manages the migration and cleanup from ML1 volumes, including moving data sets from ML1 to ML2. Therefore, it is best to schedule SSM before primary space management (PSM) so that SSM will free up more space on the ML1 volume for the PSM function.

17.1.15 Migration and storage group thresholds

Establishing unrealistically high and low thresholds in a storage group leads to excessive cycles and missed space management windows.

Primary space management attempts to process down to a low threshold.

Interval migration attempts to process starting from the midway between the high and low thresholds down to the low threshold.

If you define a low threshold of 1%, this threshold will be difficult to achieve and DFSMShsm will work excessively to gain little benefit.

17.1.16 Using INCREMENTALBACKUP(ORIGINAL)

Using INCREMENTALBACKUP(ORIGINAL) depends on the value of SETSYS INCREMENTALBACKUP(ORIGINAL | CHANGEDONLY), which is defined in ARCCMDxx. If you specify ORIGINAL, DFSMShsm will back up the data sets with no backup copies (regardless of its change bit status) or a data set with the change bit set to `on`. Specifying CHANGEDONLY means that only data sets with the data set changed indicator set to `on` are backed up.

We recommend that you specify ORIGINAL from time to time (monthly or quarterly) to ensure that all data sets that require backup are backed up.

17.1.17 Delete No Recall with IEFBR14

Production jobs often use IEFBR14 with DISP=(x,DELETE) as first step. If the data set is migrated, DFSMShsm will recall it in order to delete.

You can avoid this situation by coding a Delete No Recall parameter in SYS1.PARMLIB(ALLOCxx), for example:

```
SYSTEM IEFBR14_DELMIGDS(NORECALL)
```

17.1.18 Tuning patches that are supported by DFSMShsm

The **PATCH** command changes storage within the DFSMShsm address space. You can identify the storage location to change with an absolute address or a qualified address.

For sites with unique requirements of DFSMShsm that are not supported by existing DFSMShsm commands and parameters, these tuning patches might offer a solution. In cases where the unique requirements remain the same from day to day, you might choose to include the patches in the DFSMShsm startup member. These DFSMShsm-supported patches remain supported from release to release without modification.

Before you apply any patches to your system, consider the following information:

- You must be familiar with the DFSMShsm **PATCH** command.
- To check for any errors when you use the **PATCH** command, you must specify the **VERIFY** parameter.
- To see the output from a **PATCH** command, specify the **ODS** parameter.

Too many patches are available to be covered in this book. However, if you need to implement any patches for your particular needs, all of the complete patches are listed and described in Chapter 16 “Tuning DFSMShsm” of the *DFSMShsm Implementation and Customization Guide,* SC35-0418.

A

Sample ACS routines

This appendix contains sample automatic class selection (ACS) routines that are used to manage DFSMShsm data.

Sample storage class ACS routine

In our system, we chose a high-level qualifier (HLQ) of HSM for the DFSMShsm data sets, which is reflected in the ACS routines. Example A-1 shows the following information:

- A storage class for high performance and with guaranteed space (SC54GRT) is assigned to the DFSMShsm control data sets (CDSs) and journal. This storage class is explicitly specified in the allocation of these data sets. Only the storage administrators (MHLRES3, MHLRES4, and MHLRES5) can use the SC54GRT storage class.
- The data that is allocated on the volumes that are owned by DFSMShsm (migration and backup) is excluded from DFSMSdfp management.
- Other DFSMShsm data sets, such as logs and CDS backups, are recognized and allocated in the DFSMSdfp volume pools.
- When a data set is recalled or recovered by DFSMShsm, the storage class is retained if one exists.

Examples of the parameters that are used in the DFSMSdfp classes that are referenced in these routines are presented in 4.5, “Storage class” on page 72.

Example A-1 Storage class ACS routine for DFSMShsm data

```
PROC &STORCLAS
/*******************************************************************/
/* START OF STORAGE CLASS PROC */
/*******************************************************************/
/* CHANGE HISTORY */
/* DATE PERSON DESCRIPTION OF CHANGE */
/*******************************************************************/
/* 95/05/06 DEF INITIAL IMPLEMENTATION FOR STAGE I */
/* DATA */
/* 95/02/05 DEF MINIMAL CONFIGURATION */
/*******************************************************************/
/* YOU NEED TO MOVE ALL THE FILTLISTS TO THE */
/* LOCATION BETWEEN THE TWO LINES OF SPECIAL CHARACTERS SHOWN */
/* BELOW THIS COMMENT BEFORE USING THE CODE FRAGMENTS */
/*******************************************************************/
/*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*/
FILTLIST &OMVS_DATA_DSN INCLUDE
(OMVS*.**)
/*******************************************************************/
FILTLIST &HSM_NONSMS_DATA_DSN INCLUDE
(HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,HSM.MDB.**,
HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)
/*******************************************************************/
FILTLIST &HSM_CONTRDS_DATA_DSN INCLUDE
(HSM.%CDS.*,HSM.JRNL.*)
/*******************************************************************/
FILTLIST &HSM_LOGS_DATA_DSN INCLUDE
('HSM.EDITLOG',HSM.HSMLOG*,HSM.HSMPDO*)
/*******************************************************************/
FILTLIST &HSM_ACT_DATA_DSN INCLUDE
(HSMACT.H%.*.D*.T*)
/*******************************************************************/
FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
(HSM.%CDS.BACKUP.*,
```

```
HSM.JRNL.BACKUP.*)
/*********************************************************************/
FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
(ABARS.*.INSTRUCT)
/*********************************************************************/

FILTLIST &SMS_MANAGED_OMVS_DATA_HLQ INCLUDE
(OMVS*)
FILTLIST &SPECIAL_SC_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_SCNOSMS_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_STORCLAS INCLUDE
('SC54GRT','SC54STD','SC54LOW')
FILTLIST &SPECIAL_MC_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_MGMTCLAS INCLUDE
('MC54SPEC')
/*********************************************************************/
FILTLIST &VALID_DASD_UNIT INCLUDE
('3380','3390','SYSDA','SYSALLDA','')
FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
('STORE','CHANGE','CTRANS')
FILTLIST &VALID_TAPE_UNIT INCLUDE
('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')
/*********************************************************************/
/**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**/
SELECT /* MAIN LOGIC BLOCKS */
/*********************************************************************/
/* INCUDE ALL STANDARDS ENFORCEMENT LOGIC IN THIS BLOCK, BUT ONLY */
/* IN EITHER THE DATA CLASS OR STORAGE CLASS ACS ROUTINE */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD DATA SET NAMING STANDARD HERE IF USED */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD VOLUME FENCING STANDARD HERE IF USED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD DATA SET SIZE STANDARD HERE IF USED */
/*********************************************************************/
/*********************************************************************/
/* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK, INCLUDE BOTH DASD */
/* ALLOCATIONS FOR SMS AND NON-SMS */
/*********************************************************************/
WHEN ((&UNIT EQ &VALID_DASD_UNIT)
AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))
SELECT /* DASD DATA ALLOCATION */
/*********************************************************************/
/* EXCLUDE DFSMSHSM RECALLS AND RECOVERIES FROM ACS PROCESSING IF */
/* A STORAGE CLASS IS ALREADY ASSIGNED TO THE DATA SET */
/*********************************************************************/
WHEN ((&ACSENVIR = 'RECALL'OR &ACSENVIR ='RECOVER')
```

```
AND (&STORCLAS NE ''))
DO
SET &STORCLAS = &STORCLAS
EXIT
END /* DO */
/**************************************************************/
/* ALLOW SPECIAL USERS TO PLACE SELECTED DATA SETS OUTSIDE OF SMS */
/* BY USING STORCLAS=SCNOSMS AT ALLOCATION */
/**************************************************************/
WHEN ((&USER EQ &SPECIAL_SCNOSMS_USER)
AND (&STORCLAS EQ 'SCNOSMS'))
DO
SET &STORCLAS = ''
EXIT
END /* DO */
/**************************************************************/
/* ALLOW HSM NON-SMS SELECTED DATA SETS OUTSIDE OF SMS */
/**************************************************************/
WHEN (&DSN EQ &HSM_NONSMS_DATA_DSN)
DO
SET &STORCLAS = ''
EXIT
END /* DO */
/**************************************************************/
/* TAILOR AND ADD JCL EXCEPTIONS HERE IF THEY ARE USED */
/**************************************************************/
/**************************************************************/
/* TAILOR AND SET STORCLAS FOR ALLOCATION EXCEPTIONS */
/**************************************************************/
/* REPEAT THIS BLOCK FOR EACH SET OF SPECIAL USERS THAT ARE */
/* ALLOWED TO REQUEST SMS SERVICE THROUGH STORCLAS AT ALLOCATION */
/**************************************************************/
WHEN ((&USER EQ &SPECIAL_SC_USER)
AND (&STORCLAS EQ &SPECIAL_STORCLAS))
DO
SET &STORCLAS = &STORCLAS
EXIT
END /* DO */
/**************************************************************/
/* TAILOR AND ADD DATA SET LIST EXCEPTIONS HERE IF THEY ARE USED */
/* INCLUDE SEQUENTIAL STRIPING AND CONCURRENT COPY */
/**************************************************************/

/**************************************************************/
/* TAILOR AND ADD EITHER TEMP AND VIO OR TEMP AND NO VIO HERE IF */
/* TEMP DATA IS SMS-MANAGED */
/**************************************************************/

/**************************************************************/
/* TAILOR AND ADD TEMP SHORT DURATION HERE IF TEMP SHORT DURATION */
/* DATA AND IF IT IS SMS-MANAGED */
/**************************************************************/

/**************************************************************/
/* TAILOR AND ADD TSO LOGON AND LIST DATA SUBTYPES IF THEY HAVE */
```

```
/* STORAGE CLASSES DIFFERENT THAN OTHER TSO DATA AND THEY ARE */
/* SMS-MANAGED */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD OTHER TSO SUBTYPES HERE IF THEY ARE SMS-MANAGED */
/*********************************************************************/

/*********************************************************************/
/* TAILOR AND ADD TEST SUBTYPES HERE IF THEY ARE SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD BATCH PRODUCTION SUBTYPES HERE IF THEY ARE */
/* SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD ONLINE PRODUCTION SUBTYPES HERE IF THEY ARE */
/* SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD ANY OTHER DATA SUBTYPES HERE IF THEY ARE */
/* SMS-MANAGED */
/*********************************************************************/
WHEN (&DSN EQ &OMVS_DATA_DSN)
DO
SET &STORCLAS ='OPENMVS'
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_LOGS_DATA_DSN)
DO
SET &STORCLAS ='SC54STD'
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_ACT_DATA_DSN)
DO
SET &STORCLAS ='SC54LOW'
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_BKCONT1_DATA_DSN)
DO
SET &STORCLAS ='SC54LOW'
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_ABARS_DATA_DSN)
DO
SET &STORCLAS ='SC54LOW'
EXIT
END /* DO */
/*********************************************************************/
/* MAKE ALL OTHER DASD DATA SETS NON-MANAGED */
/*********************************************************************/
OTHERWISE /* NON-MANAGED DASD DATA */
DO
SET &STORCLAS = ''
EXIT
```

```
END /* DO */
END /* DASD ALLOCATION SELECT */
/*********************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK */
/*********************************************************************/
WHEN (&UNIT EQ &VALID_TAPE_UNIT)
SELECT /* TAPE ALLOCATIONS */
/*********************************************************************/
/* TAILOR AND ADD TAPE MOUNT MANAGEMENT SUBTYPES HERE IF THEY EXIST */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD SMS-MANAGED TAPE SUBTYPES HERE IF THEY EXIST */
/*********************************************************************/
/*********************************************************************/
/* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT */
/* OR SMS-MANAGED TAPE EXIST */
/*********************************************************************/
WHEN (&UNIT EQ &VALID_TAPE_UNIT)
SET &STORCLAS = &STORCLAS
END /* TAPE ALLOCATION SELECT */
/*********************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK, INCLUDE LOGIC FOR */
/* SUBTYPES FOR IAFC OR IMAGE PLUS HERE IF THEY EXIST */
/*********************************************************************/
WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
SELECT /* OPTICAL ALLOCATIONS */
WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
SET &STORCLAS = &STORCLAS
/*********************************************************************/
/* TAILOR AND ADD OPTICAL STORE FUNCTION HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD OPTICAL TRANSITION AND CHANGE FUNCTIONS HERE IF */
/* OPTICAL DATA EXISTS */
/*********************************************************************/
/*********************************************************************/
/* ADD OTHERWISE FOR OPTICAL DATA HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/
END /* OPTICAL ALLOCATION SELECT */
/*********************************************************************/
/* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR */
/* OPTICAL */
/*********************************************************************/
OTHERWISE /* NON-MANAGED DEVICES */
DO
SET &STORCLAS = ''
EXIT
END /* DO */
END /* SELECT */
END /* PROC */
```

Sample management class ACS routine

Example A-2 shows sample code that is required in the management class ACS routine, which assigns different management criteria to the DFSMShsm-owned data sets according to their use and importance.

Example A-2 Management class ACS routine for DFSMShsm data

```
PROC &MGMTCLAS
/*********************************************************************/
/* START OF MANAGEMENT CLASS PROC */
/*********************************************************************/
/* CHANGE HISTORY */
/* DATE PERSON DESCRIPTION OF CHANGE */
/*********************************************************************/
/* 95/05/06 DEF INITIAL IMPLEMENTATION FOR STAGE I */
/* DATA */
/* 95/02/05 DEF MINIMAL CONFIGURATION */
/*********************************************************************/
/* YOU NEED TO MOVE ALL THE FILTLISTS TO THE */
/* LOCATION BETWEEN THE TWO LINES OF SPECIAL CHARACTERS SHOWN */
/* BELOW THIS COMMENT BEFORE USING THE CODE FRAGMENTS */
/*********************************************************************/
/*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*/
FILTLIST &OMVS_DATA_DSN INCLUDE
(OMVS*.**)
/*********************************************************************/
FILTLIST &HSM_NONSMS_DATA_DSN INCLUDE
(HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,
HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)
/*********************************************************************/
FILTLIST &HSM_CONTRDS_DATA_DSN INCLUDE
(HSM.%CDS.*,HSM.JRNL.*)
/*********************************************************************/
FILTLIST &HSM_LOGS_DATA_DSN INCLUDE
('HSM.EDITLOG',HSM.HSMLOG*,HSM.HSMPDO*)
/*********************************************************************/
FILTLIST &HSM_ACT_DATA_DSN INCLUDE
(HSMACT.H%.*.D*.T*)
/*********************************************************************/
FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
(HSM.%CDS.BACKUP.*,
HSM.JRNL.BACKUP.*)
/*********************************************************************/
FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
(ABARS.*.INSTRUCT)
/*********************************************************************/
FILTLIST &SMS_MANAGED_OMVS_DATA_HLQ INCLUDE
(OMVS*)
FILTLIST &SPECIAL_SC_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_SCNOSMS_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_STORCLAS INCLUDE
('SC54GRT','SC54STD','SC54LOW')
```

```
FILTLIST &SPECIAL_MC_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_MGMTCLAS INCLUDE
('MC54SPEC')
/*******************************************************************/
FILTLIST &VALID_DASD_UNIT INCLUDE
('3380','3390','SYSDA','SYSALLDA','')
FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
('STORE','CHANGE','CTRANS')
FILTLIST &VALID_TAPE_UNIT INCLUDE
('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')
/*******************************************************************/
/**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**/
SELECT /* MAIN LOGIC BLOCKS */
/*******************************************************************/
/* INCLUDE AN EMPTY STANDARDS ENFORCEMENT BLOCK HERE FOR */
/* COMPATIBILITY WITH THE OTHER ACS ROUTINES */
/*******************************************************************/


/*******************************************************************/
/* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK */
/*******************************************************************/
WHEN ((&UNIT EQ &VALID_DASD_UNIT)
AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))
SELECT /* DASD DATA ALLOCATION */
/*******************************************************************/
/* EXCLUDE DFSMSHSM RECALLS AND RECOVERIES FROM ACS PROCESSING IF */
/* A MANAGEMENT CLASS IS ALREADY ASSIGNED TO THE DATA SET */
/*******************************************************************/
WHEN ((&ACSENVIR = 'RECALL'OR &ACSENVIR ='RECOVER')
AND (&MGMTCLAS NE ''))
DO
SET &MGMTCLAS = &MGMTCLAS
EXIT
END /* DO */
/*******************************************************************/
/* TAILOR AND ADD JCL EXCEPTIONS HERE IF THEY ARE USED */
/*******************************************************************/


/*******************************************************************/
/* TAILOR AND SET MGMTCLAS FOR ALLOCATION EXCEPTIONS */
/*******************************************************************/
/* REPEAT THIS BLOCK FOR EACH SET OF SPECIAL USERS THAT ARE */
/* ALLOWED TO REQUEST SMS SERVICE THROUGH MGMTCLAS AT ALLOCATION */
/*******************************************************************/
WHEN ((&USER EQ &SPECIAL_MC_USER)
AND (&MGMTCLAS EQ &SPECIAL_MGMTCLAS))
DO
SET &MGMTCLAS = &MGMTCLAS
EXIT
END /* DO */
/*******************************************************************/
/* TAILOR AND ADD DATA SET LIST EXCEPTIONS HERE IF THEY ARE USED */
```

```
/* INCLUDE CONCURRENT COPY */
/**********************************************************************/


/**********************************************************************/
/* SYSTEM TEMPORARY DATA SETS DO NOT HAVE MANAGEMENT CLASSES */
/**********************************************************************/
/**********************************************************************/
/* TAILOR AND ADD TEMP SHORT DURATION HERE IF IT EXISTS */
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD TSO LOGON AND LIST SUBTYPES IF THEY HAVE */
/* MANAGEMENT CLASSES DIFFERENT THAN OTHER TSO DATA */
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD OTHER TSO SUBTYPES HERE IF THEY EXIST */
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD TEST SUBTYPES HERE IF THEY EXIST
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD BATCH PRODUCTION SUBTYPES HERE IF THEY EXIST */
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD ONLINE PRODUCTION SUBTYPES HERE IF THEY EXIST */
/**********************************************************************/
/**********************************************************************/
/* TAILOR AND SET MGMTCLAS FOR OTHER SUBTYPES */
/**********************************************************************/
/* REPEAT THIS BLOCK FOR EACH OTHER SUBTYPE THAT HAS A DIFFERENT */
/* MANAGEMENT CLASS */
/**********************************************************************/
WHEN (&DSN EQ &HSM_BKCONT1_DATA_DSN)
DO
SET &MGMTCLAS =’MC54NMIG’
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_ACT_DATA_DSN)
DO
SET &MGMTCLAS =’MC54WORK’
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_LOGS_DATA_DSN)
DO
SET &MGMTCLAS =’MC54SPEC’
```

```
EXIT
END /* DO */
WHEN (&DSN EQ &HSM_ABARS_DATA_DSN)
DO
SET &MGMTCLAS ='MC54PRIM'
EXIT
END /* DO */
/**********************************************************************/
/* TAILOR AND ADD OTHER SUBTYPES HERE IF THEY EXIST */
/**********************************************************************/
/**********************************************************************/
/* TAILOR AND SET MGMTCLAS FOR ALL OTHER SMS-MANAGED SUBTYPES */
/**********************************************************************/
OTHERWISE /* UNEXPECTED DASD DATA */
DO
SET &MGMTCLAS = 'MC54NMIG'
EXIT
END /* DO */
END /* DASD ALLOCATION SELECT */
/**********************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK */
/**********************************************************************/
WHEN (&UNIT EQ &VALID_TAPE_UNIT)
SELECT /* TAPE ALLOCATIONS */
/**********************************************************************/
/* TAILOR AND ADD TAPE MOUNT MANAGEMENT SUBTYPES HERE IF THEY EXIST */
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD SMS-MANAGED TAPE SUBTYPES HERE IF THEY EXIST */
/**********************************************************************/


/**********************************************************************/
/* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT */
/* OR SMS-MANAGED TAPE EXIST */
/**********************************************************************/
WHEN (&UNIT EQ &VALID_TAPE_UNIT)
SET &MGMTCLAS = &MGMTCLAS
END /* TAPE ALLOCATION SELECT */
/**********************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK */
/**********************************************************************/
WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
SELECT /* OPTICAL ALLOCATIONS */
WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
SET &MGMTCLAS = &MGMTCLAS
/**********************************************************************/
/* TAILOR AND ADD OPTICAL STORE FUNCTION HERE IF OPTICAL DATA EXISTS */
/**********************************************************************/


/**********************************************************************/
/* TAILOR AND ADD OPTICAL TRANSITION AND CHANGE FUNCTIONS HERE IF */
```

```
/* OPTICAL DATA EXISTS */
/*******************************************************************/
/*******************************************************************/
/* ADD OTHERWISE FOR OPTICAL DATA HERE IF OPTICAL DATA EXISTS */
/*******************************************************************/


END /* OPTICAL ALLOCATION SELECT */
/*******************************************************************/
/* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR */
/* OPTICAL */
/*******************************************************************/
OTHERWISE /* UNEXPECTED DEVICES */
DO
SET &MGMTCLAS = 'MC54NMIG'
EXIT
END /* DO */
END /* SELECT */
END /* PROC */
```

Sample storage group ACS routine

Example A-3 shows sample code that we used in our storage group ACS routine. You might need to create another storage group and update your storage group ACS if you want to separate the DFSMShsm data sets from other data.

Example A-3 Storage group ACS routine for DFSMShsm data

```
PROC &STORGRP
/*******************************************************************/
/* START OF STORAGE GROUP PROC */
/*******************************************************************/
/* CHANGE HISTORY */
/* DATE PERSON DESCRIPTION OF CHANGE */
/*******************************************************************/
/* 95/05/06 DEF INITIAL IMPLEMENTATION FOR STAGE I */
/* DATA */
/* 95/02/05 DEF MINIMAL CONFIGURATION */
/*******************************************************************/
/* YOU NEED TO MOVE ALL THE FILTLISTS TO THE */
/* LOCATION BETWEEN THE TWO LINES OF SPECIAL CHARACTERS SHOWN */
/* BELOW THIS COMMENT BEFORE USING THE CODE FRAGMENTS */
/*******************************************************************/
/*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*/
FILTLIST &OMVS_DATA_DSN INCLUDE
(OMVS*.**)
/*******************************************************************/
FILTLIST &HSM_NONSMS_DATA_DSN INCLUDE
(HSM.SMALLDS.**,HSM.BACK.**,HSM.HMIG.**,
HSM.VCAT.**,HSM.VTOC.**,HSM.DUMPVTOC.**)
/*******************************************************************/
FILTLIST &HSM_CONTRDS_DATA_DSN INCLUDE
(HSM.%CDS.*,HSM.JRNL.*)
/*******************************************************************/
```

```
FILTLIST &HSM_LOGS_DATA_DSN INCLUDE
('HSM.EDITLOG',HSM.HSMLOG*,HSM.HSMPDO*)
/*******************************************************************/
FILTLIST &HSM_ACT_DATA_DSN INCLUDE
(HSMACT.H%.*.D*.T*)
/*******************************************************************/
FILTLIST &HSM_BKCONT1_DATA_DSN INCLUDE
(HSM.%CDS.BACKUP.*,
HSM.JRNL.BACKUP.*)
/*******************************************************************/
FILTLIST &HSM_ABARS_DATA_DSN INCLUDE
(ABARS.*.INSTRUCT)
/*******************************************************************/
FILTLIST &SMS_MANAGED_OMVS_DATA_HLQ INCLUDE
(OMVS*)
FILTLIST &SPECIAL_SC_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_SCNOSMS_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_STORCLAS INCLUDE
('SC54GRT','SC54STD','SC54LOW')
FILTLIST &SPECIAL_MC_USER INCLUDE
('MHLRES3','MHLRES4','MHLRES5')
FILTLIST &SPECIAL_MGMTCLAS INCLUDE
('MC54SPEC')
/*******************************************************************/
FILTLIST &VALID_DASD_UNIT INCLUDE
('3380','3390','SYSDA','SYSALLDA','')
FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
('STORE','CHANGE','CTRANS')
FILTLIST &VALID_TAPE_UNIT INCLUDE
('3490','3480','3590',TAPE*,'T3480','T3490','T3590','AFF=')
/*******************************************************************/
/**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**/
SELECT /* MAIN LOGIC BLOCKS */
/*******************************************************************/
/* INCLUDE AN EMPTY STANDARDS ENFORCEMENT BLOCK HERE FOR */
/* COMPATIBILITY WITH THE OTHER ACS ROUTINES */
/*******************************************************************/


/*******************************************************************/
/* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK */
/*******************************************************************/
WHEN ((&UNIT EQ &VALID_DASD_UNIT)
AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))
SELECT /* DASD DATA ALLOCATION */
/*******************************************************************/
/* TAILOR AND ADD GUARANTEED SPACE LOGIC HERE IF GUARANTEED SPACE IS */
/* USED */
/*******************************************************************/
WHEN ((&USER EQ &SPECIAL_SC_USER)
AND (&STORCLAS EQ 'SC54GRT') )
DO
SET &STORGRP EQ 'SG54HSM'
```

```
EXIT
END
/*********************************************************************/
/* TAILOR AND ADD DATA SET LIST EXCEPTIONS HERE IF THEY ARE USED */
/* INCLUDE SEQUENTIAL STRIPING AND CONCURRENT COPY */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD EITHER TEMP AND VIO OR TEMP AND NO VIO HERE IF */
/* TEMPORARY DATA IS SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD TEMP SHORT DURATION HERE IF THERE IS TEMP SHORT */
/* DURATION DATA AND IF IT IS SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD TSO LOGON AND LIST SUBTYPE IF THEY HAVE STORAGE */
/* GROUPS DIFFERENT THAN OTHER TSO SUBTYPES AND THEY ARE SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD OTHER TSO SUBTYPES HERE IF THEY ARE SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD TEST SUBTYPES HERE IF THEY ARE SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD BATCH PRODUCTION SUBTYPES HERE IF THEY ARE */
/* SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND ADD ONLINE PRODUCTION SUBTYPES HERE IF THEY ARE */
/* SMS-MANAGED */
/*********************************************************************/
/* TAILOR AND ADD OTHER SUBTYPES HERE IF THEY ARE SMS-MANAGED */
/*********************************************************************/
/*********************************************************************/
/* TAILOR AND SET STORGRP FOR ALL OTHER SMS-MANAGED SUBTYPES */
/*********************************************************************/
WHEN (&DSN EQ &OMVS_DATA_DSN)
DO
SET &STORGRP =’OPENMVS’
END /* DO */
OTHERWISE /* UNEXPECTED DASD DATA */
DO
SET &STORGRP = ’SG54HSM’
EXIT
END /* DO */
END /* DASD ALLOCATION SELECT */
/*********************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK */
/*********************************************************************/
WHEN (&UNIT EQ &VALID_TAPE_UNIT)
SELECT /* TAPE ALLOCATIONS */
/*********************************************************************/
/* TAILOR AND ADD TAPE MOUNT MANAGEMENT SUBTYPES HERE IF THEY EXIST */
/*********************************************************************/
```

```
/*********************************************************************/
/* TAILOR AND ADD SMS-MANAGED TAPE SUBTYPES HERE IF THEY EXIST */
/*********************************************************************/


/*********************************************************************/
/* ADD OTHERWISE FOR TAPE DATA HERE IF EITHER TAPE MOUNT MANAGEMENT */
/* OR SMS-MANAGED TAPE EXIST */
/*********************************************************************/



END /* TAPE ALLOCATION SELECT */
/*********************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK */
/*********************************************************************/
WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
SELECT /* OPTICAL ALLOCATIONS */
WHEN (&ACSENVIR EQ &VALID_OPTICAL_ACSENVIR)
SET &STORGRP = &STORGRP
/*********************************************************************/
/* TAILOR AND ADD OPTICAL STORE FUNCTION HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/


/*********************************************************************/
/* TAILOR AND ADD OPTICAL TRANSITION FUNCTION HERE IF OPTICAL DATA */
/* EXISTS */
/*********************************************************************/


/*********************************************************************/
/* ADD OTHERWISE FOR OPTICAL DATA HERE IF OPTICAL DATA EXISTS */
/*********************************************************************/


END /* OPTICAL ALLOCATION SELECT */
/*********************************************************************/
/* THIS OTHERWISE IS FOR ALLOCATIONS OTHER THAN DASD, TAPE, OR */
/* OPTICAL */
/*********************************************************************/


OTHERWISE /* UNEXPECTED DEVICES */
DO
SET &STORGRP = 'SG54HSM'
EXIT
END /* DO */
END /* SELECT */
END /* PROC */
```

Sample data class ACS routine

Example A-4 shows sample code that we used in our data class ACS routine.

Example A-4 Data class ACS routine for DFSMShsm data

```
PROC &DATACLAS
/**********************************************************************/
/* START OF DATA CLASS PROC */
/**********************************************************************/
/* CHANGE HISTORY */
/* DATE PERSON DESCRIPTION OF CHANGE */
/**********************************************************************/
/* 95/05/06 DEF INITIAL IMPLEMENTATION FOR STAGE I */
/* DATA */
/* 95/02/05 DEF MINIMAL CONFIGURATION */
/**********************************************************************/
/*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*/

FILTLIST &STANDARD_NAME_DSN INCLUDE
(%%%T%%%.**,*.T*.**,*.P*.**)
FILTLIST &VALID_DASD_UNIT INCLUDE
('3380','3390','SYSDA','SYSALLDA','')
FILTLIST &VALID_OPTICAL_ACSENVIR INCLUDE
('STORE','CHANGE','CTRANS')
FILTLIST &SPECIAL_NONSMS_USER INCLUDE
('FREEMAN')
FILTLIST &SPECIAL_SC_USER INCLUDE
('FREEMAN')
FILTLIST &SPECIAL_DATACLAS INCLUDE
('SCSPECN','SCSPECF')
FILTLIST &CRITICAL_PERFORMANCE_DSN INCLUDE
(FREEMAN.P*.NEEDFAST)
FILTLIST &TSO_DATA_HLQ INCLUDE
(%%%T%%%)
FILTLIST &TEST_DATA_DSN INCLUDE
(T*.**)
FILTLIST &BATCH_PROD_DATA_DSN INCLUDE
(P*.**)
FILTLIST &VALID_TAPE_UNIT INCLUDE
('3490','3480','3590',TAPE*,'T3480','T3490',
'T3590','AFF=')
FILTLIST &TAPE_MOUNT_MANAGEMENT_DSN INCLUDE
('')
FILTLIST &SMS_MANAGED_TAPE_DSN INCLUDE
('')
FILTLIST &OPTICAL_DSN INCLUDE
('')
/**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**/

SELECT /* ALLOCATION TYPE */

/**********************************************************************/
/* INCLUDE ALL STANDARDS ENFORCEMENT LOGIC IN THIS BLOCK IN */
/* ONE AND ONLY ONE OF EITHER THE DATA CLASS OR STORAGE ACS */
```

```
/* ROUTINE, LEAVE BLOCK IN OTHER ACS ROUTINES FOR CONSISTENCY */
/**********************************************************************/
/**********************************************************************/
/* START STANDARDS ENFORCEMENT LOGIC BLOCK */
/**********************************************************************/


/**********************************************************************/
/* END STANDARDS ENFORCEMENT LOGIC BLOCK */
/**********************************************************************/

/**********************************************************************/
/* INCLUDE ALL DASD ALLOCATIONS IN THIS BLOCK, INCLUDE */
/* BOTH DASD ALLOCATIONS TO SMS AND TO NON SMS */
/**********************************************************************/

WHEN ((&UNIT EQ &VALID_DASD_UNIT)
AND (&ACSENVIR NE &VALID_OPTICAL_ACSENVIR))
SELECT /* DASD DATA ALLOCATION */

/**********************************************************************/
/* DO NOT MANAGE DATA FOR SPECIAL USERS THAT HAVE BEEN */
/* GIVEN AUTHORITY TO USE DATACLAS=SCNONSMS ON THEIR JCL */
/* THIS ALLOWS SELECTIVE OVERRIDE OF THE DATA CLASS */
/* ACS LOGIC DURING AN EMERGENCY SITUATION SO DATA THAT */
/* WOULD NORMALLY BE MANAGED BY SMS CAN BE ALLOCATED OUTSIDE OF SMS */
/**********************************************************************/
WHEN (&DATACLAS NE '')
DO
SET &DATACLAS = &DATACLAS
EXIT
END /* DO */

/**********************************************************************/
/* FOR ANY OTHER DATA TYPES NOT IDENTIFIED FOR */
/* SMS MANAGEMENT, PLACE THEM IN NON SMS STORAGE */
/**********************************************************************/

OTHERWISE /* NONMANAGED DASD DATA */
DO
SET &DATACLAS = ''
EXIT
END /* DO */
END /* DASD ALLOCATION SELECT */

/**********************************************************************/
/* INCLUDE ALL TAPE ALLOCATIONS IN THIS BLOCK, INCLUDE */
/* BOTH TAPE ALLOCATIONS TO SMS AND TO NON SMS */
/* PRIOR TO SMS MANAGED TAPE OR TAPE MOUNT MANAGEMENT */
/* ALL TAPE ALLOCATION WOULD BE TO NON SMS */
/**********************************************************************/
/**********************************************************************/
/* START TAPE ALLOCATION LOGIC BLOCK */
/**********************************************************************/
```

```
/*********************************************************************/
/* END TAPE ALLOCATION LOGIC BLOCK */
/*********************************************************************/
/*********************************************************************/
/* INCLUDE ALL OPTICAL ALLOCATIONS IN THIS BLOCK, INCLUDE */
/* LOGIC FROM PRODUCTS LIKE IAFC OR IMAGE PLUS HERE */
/*********************************************************************/
/*********************************************************************/
/* START OPTICAL ALLOCATION LOGIC BLOCK */
/*********************************************************************/


/*********************************************************************/
/* END OPTICAL ALLOCATION LOGIC BLOCK */
/*********************************************************************/

OTHERWISE /* NONMANAGED DEVICES */
DO
SET &DATACLAS =''
EXIT
END /* DO */
END /* SELECT */
END /* PROC */
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Exploiting the IBM Health Checker for z/OS Infrastructure*, REDP-4590
- *IBM zEnterprise BC12 Technical Guide*, SG24-8138
- *IBM zEnterprise EC12 Technical Guide,* SG24-8049
- *Reduce Storage Occupancy and Increase Operations Efficiency with IBM zEnterprise Data Compression*, SG24-8259
- *zEDC Compression: DFSMShsm Sample Implementation*, REDP-5158
- *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268
- *VSAM Demystified*, SG24-6105
- *z/OS V1.13 Technical Update,* SG24-7961
- *z/OS V1R6 DFSMS Technical Guide*, SG24-6651
- *z/OS V1R7 DFSMS Technical Update*, SG24-7225
- *z/OS Version 1 Release 7 Implementation*, SG24-6755
- *z/OS V1R8 DFSMS Technical Update*, SG24-7435

You can search for, view, download or order these documents and other Redbooks, Redpapers, web docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *DFSMS Advanced Services*, SC26-7400
- *DFSMS Access Method Services for Catalogs*, SC26-7394
- *DFSMS Installation Exits*, SC26-7396
- *DFSMS Macro Instructions for Data Sets*, SC26-7408
- *DFSMS Managing Catalogs*, SC26-7409
- *DFSMS Using the New Functions*, SC26-7473
- *DFSMS Using Data Sets*, SC26-7410
- *DFSMS Utilities*, SC26-7414

- *DFSMSdss Storage Administration Guide*, SC26-0423
- *DFSMSdss Storage Administration Reference*, SC35-0424
- *DFSMShsm Diagnosis*, GC52-1083
- *z/OS DFSMShsm Implementation and Customization Guide*, SC23-6869
- *DFSMShsm Managing Your Own Data*, SC35-0420
- *DFSMShsm Storage Administration Guide*, SC26-0421
- *DFSMShsm Storage Administration Reference*, SC26-0422
- *DFSMSrmm Application Programming Interface,* SC26-7403
- *DFSMSrmm Guide and Reference,* SC26-7404
- *OS/390 V2R10.0 DFSMSdfp Storage Administration Reference,* SC26-7331
- *DFSMSrmm Implementation and Customization Guide,* SC26-7405
- *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC26-0426
- *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*, SC35-0427
- *MVS Initialization and Tuning Guide,* SA22-7591
- *MVS Initialization and Tuning Reference,* SA22-7592
- *MVS System Commands,* SA22-7627
- *MVS System Management Facilities (SMF)*, SA22-7630
- *Partitioned Data Set Extended Usage Guide*, SG24-6106
- *z/OS Security Server RACF System Programmer's Guide*, SA22-7681
- *z/OS DFSMS Introduction*, SC26-7397
- *z/OS DFSMS Storage Administration Reference (for DFSMShsm, DFSMSdss, DFSMSdfp)*, SC26-7402
- *z/OS Hardware Configuration Definition User's Guide*, SC33-7988
- *z/OS and z/VM Hardware Configuration Manager User's Guide*, SC33-7989
- *z/OS MVS System Messages Vol 1 (ABA-AOM),* SA22-7631
- *z/OS MVS System Messages Vol 2 (ARC-ASA),* SA22-7632
- *z/OS MVS System Messages Vol 3 (ASB-BPX),* SA22-7633
- *z/OS MVS System Messages Vol 4 (CBD-DMO),* SA22-7634
- *z/OS MVS System Messages Vol 5 (EDG-GFS),* SA22-7635
- *z/OS MVS System Messages Vol 6 (GOS-IEA),* SA22-7636
- *z/OS MVS System Messages Vol 7 (IEB-IEE),* SA22-7637
- *z/OS MVS System Messages Vol 8 (IEF-IGD),* SA22-7638
- *z/OS MVS System Messages Vol 9 (IGF-IWM),* SA22-7639
- *z/OS MVS System Messages Vol 10 (IXC-IZP),* SA22-7640
- *z/OS MVS Planning: Operations*, SA22-7601
- *z/OS and z/OS.e Planning for Installation*, GA22-7504

- *IBM Health Checker for z/OS User's Guide,* SA22-7994
- *DFSMS Software Support for IBM System Storage TS1130 and TS1120 tape drives (3592)*, SC26-7514

Online resources

These websites are also relevant as further information sources:

- Capture and archive the contents of PDA files when they are swapped:

  http://www.ibm.com/support/docview.wss?uid=isg3T1012687
- The CF sizer:

  http://www.ibm.com/systems/support/z

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

ABARS activity log 408
activity log 40
administration
  converting level 0 DASD volumes 334
  converting level 1 DASD volumes 334
  expire old backup versions 328
  EXPIREBV command output 328
  FREEVOL command 334
  reorganizing CDSs 343
aggregate backup and recovery support
  ISMF panels 281
  overview 15
  QUERY ABARS command 385
  query ABARS parameters 385
  RACF protection 91
ARCCATGP group 92
ARCTOOLS utilities 398
AUDIT command 335, 418
AUTH command 37, 88
authorized user 37
automatic class selection routines
  processing order 4
automatic secondary space management 12
availability management
  dump processing 265
  functions 243
  management class attributes 248
  overview 13

B

backup
  inline 14
backup control data set
  AUDIT BCDS command 336, 422
  CDS reorganization 343
  expire old backup versions 328
  LIST command 374
  LIST DUMPCLASS command 379
  listing migration volumes 381
  threshold value 41

C

command backup
  selecting target device 46
command data set backup 46
commands
  AUDIT 335–336, 418, 422
  AUTH 37, 88
  EXPIREBV 328
  FREEVOL 334
  HSENDCMD 328
  LIST DUMPCLASS 379
  LIST MIGRATIONVOLUME 381
  QUERY 382
  QUERY ABARS 385
  QUERY MIGRATIONLEVEL2 390
  QUERY SPACE 387
  QUERY STARTUP 383
  QUERY STATISTICS 384
  QUERY TRAPS command 384
  QUERY WAITING 390
  REPORT 390
common service area 41
concurrent copy 367
  RACF protection 92
control data sets
  backup 350
  backup procedures 367
  CDS reorganization 343
  creating backup copies 351
  extended addressability in a sysplex 208
  LIST command 374
  multicluster CDSs 364
  multicluster performance 365
  removing key range 97, 357
  RLS access mode 96, 356
  serialization technique 97, 357

D

data set compaction 47
  optimizing 48
data set serialization 38
DCOLLECT function 398
DFSMS health check 428
DFSMSdfp 2
DFSMSdss 2
DFSMShsm
  activity log 40
  activity log messages 40
  authorized user 37
  automatic primary space management 11
  automatic secondary space management 12
  base SETSYS parameters 37
  data set serialization 38
  DDNAMES in start procedure 36
  directing monitor messages 40
  exit activation 39
  functions for non-SMS-managed volumes 15
  functions for SMS-managed volumes 4
  installation exits 39
  interval space management 12
  monitoring DFSMShsm 29, 373
  overview 2
  protecting tapes 46, 85
  reblocking data sets 38
  secondary host promotion 205

selecting scratch pool 43
SMF records 39
SMS interaction 3
space management 10
sysplex considerations 201
system dump data set 40
tape reuse 41
trace log data sets 403
DFSMShsm monitoring
ARCTOOLS utilities 398
DCOLLECT 398
DCOLLECT function 398
function summary report 391
information sources 373
LIST command syntax 374
LIST DUMPCLASS command 379
listing dump class attributes 379
listing migration volumes 381
QUERY ABARS command 385
QUERY command 382
query daily statistics 384
query DFSMShsm statistics 384
QUERY MIGRATIONLEVEL2 command 390
query ML2 volumes 390
QUERY SPACE command 387
QUERY STARTUP command 383
QUERY STATISTICS command 384
QUERY TRAPS command 384
QUERY WAITING command 390
REPORT command 390
DFSMSopt 2
DFSMS-related health checks 428
DFSMSrmm 2
disaster recovery
duplex tape 44
dump process
listing dump class attributes 379

E

esoteric tape unit names 43
expire old backup versions 328
EXPIREBV command 328

F

Fast replication 15

H

hardware compaction 41
Health Checker 428
HSEND command 328

I

ICHRIN03 module 82
IDRC 41
IGDSMSxx 103, 363
implementation
query daily statistics 384
inline backup 14
installation 428
installation exits 39
interval space management 12

J

journal data set
manual backup 355
SETSYS JOURNAL 39
threshold value 41

L

log data sets
activity log data sets 408
SETSYS ACTLOGTYPE command 408
switching trace log data sets 403
trace log data sets 403

M

management class
backup attributes 248
messages
ARC0734I 40
migration considerations 431
migration control data set
CDS reorganization 343
LIST command 374
listing migration volumes 381
threshold value 41
monitoring DFSMShsm 29, 373
multicluster control data sets
overview 364
performance considerations 365
multiple host environment
defining primary host 185, 440
multitasking 201
volume serialization 200

O

offline control data set
CDS reorganization 343
LIST command 374
threshold value 41
operation
changing DFSMShsm control parameters 325
operations
cancel queued requests 324
restart after abnormal end 327

P

problem determination
activity log data sets 408
archiving PDA trace data 403
AUDIT command 335, 418
auditing 335, 418
IPCS considerations 40
monitoring DFSMShsm 29, 373
QUERY TRAPS command 384

QUERY WAITING command 390
SVC dumps 404
switching trace log data sets 403
trace log data sets 403

Q

QUERY command 382

R

RACF
ABARS protection 91
ADDUSER command 80
ARCCATGP group 92
catalog operations for migrated data sets 92
copying discrete data set profiles 47
ICHRIN03 82
LISTDSD command 84
protecting activity logs 84
protecting backup data set 93
protecting concurrent copy 92
protecting DFSMShsm resources 84
protecting DFSMShsm tapes 85
protecting migration data set 93
protection for ABARS 91
RDEFINE command 81
RLIST command 81
RLS facility classes profiles 363
SETSYS RACFIND 45
started procedure table 81
tape security 46
TAPEVOL resource class 85
verifying DFSMShsm generic data set profile 84
record level sharing
altering IGDSNSxx member 103, 363
altering SMS configuration 100
cache and lock structures 98, 358
DFSMShsm procedure 103, 364
implementation 97, 357
implementation checklist 104, 364
RACF Facility class profiles 363
RACF facility class profiles 363
sample SHCDS allocation JCL 98, 358
serialization 96, 356
SHCDS activation 103, 363
SMSVSAM address space 103, 363
storage class update 102, 362
RECYCLE
converting to new DASD Volumes 334
number of tasks 41
SETSYS ML2RECYCLEPERCENT command 331
SETSYS RECYCLEPERCENT command 331
setting a threshold 331
tape copy considerations 333
tape drives 41
Redbooks website 467
Contact us xv
RLS
See record level sharing

S

secondary host promotion 205
configuring automatic backup 207
primary host 205
secondary host 205
SSM host 205
take back function 207
security
providing security to DFSMShsm tapes 46
SETSYS parameter
BACKUPPREFIX 93
SETSYS parameters
ACCEPTPSCBUSERID 47
ACTLOGMSGLVL 40
ACTLOGTYPE 40, 84, 408
CDSVERSIONBACKUP 350, 355
COMPACT 47
COMPACTPERCENT 48
CONVERSION 38
CSALIMITS 41
DFHSMDATASETSERIALIZATION 38
DSBACKUP 46
DUPLEX 44
ERASEONSCRATCH 46
EXITOFF 39
EXITON 39
INPUTTAPEALLOCATION 42
JOURNAL 39
MAXRECYCLETASKS 41
MIGRATEPREFIX 93
ML2RECYCLEPERCENT 331
MONITOR 40
MOUNTWAITTIME 44
NOACCEPTPSCBUSERID 47
NOCONVERSION 38
NOCSALIMITS 41
NOERASEONSCRATCH 46
NOPROFILEBACKUP 47
NORACFIND 45
NOSMF 39
NOSYS1DUMP 40
NOTAPEHARDWARECOMPACT 41
OBJECTNAMES 48
OUTPUTTAPEALLOCATION 42
PARTIALTAPE 41
PROFILEBACKUP 47
RACFIND 45
RECYCLEPERCENT 331
RECYCLETAPEALLOCATION 42
SELECTVOLUME 43
SMF 39
SOURCENAMES 48
spanning data sets 42
SYS1DUMP 40
SYSOUT 39
TAPEDELETION 43
TAPEHARDWARECOMPACT 41
TAPEINPUTPROMPT 44
TAPESECURITY 46, 85
TAPESPANSIZE 42

TAPEUTILIZATION 45
USERDATASETSERIALIZATION 38
USERUNITTABLE 43
SHCDS
definition 98, 358
SMF records 39
SMS health check function from SDSF 429
SMSVSAM address space 103, 363
space management
automatic primary space management 11
automatic secondary space management 12
interval space management 12
migration 11
overview 10
recall 11
recycle 11
spanning data sets 42
storage class
RLS changes 101, 361
storage group
backup attributes 250
storage management system
updating configuration for RLS 100
sysplex 429
sysplex considerations
CDS extended addressability 208
coexistence 205
overview 201
single GRSplex serialization 202
system dump data set 40
system managed storage
ACS routines 4
aggregate group 4
data class 3
management class 4
overview 3
sample data class ACS routine 463
sample storage class ACS routine 450
sample storage group ACS routine 459
storage class 3
storage group 4

T

tape management
duplex tape 44
erase-on-scratch 46
esoteric unit names 43
global scratch pool 43
LIBRARYBACKUP subparameter 45
LIBRARYMIGRATION subparameter 45
mount wait time 44
operator prompts 44
RACF protecting tapes 85
specific scratch pool 43
tape pools 43
tape reuse 41
tape security 46
tape utilization 45
trace log data sets 403

V

virtual concurrent copy 368
VSAM RLS 96, 356

Redbooks

IBM z/OS DFSMShsm Primer

SG24-5272-03

ISBN 0738440841

IBM

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

SG24-5272-03

ISBN 0738440841

Printed in U.S.A.