z/OS



DFSMShsm Storage Administration Guide

SC35-0421-03

z/OS



DFSMShsm Storage Administration Guide

SC35-0421-03

Note
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 643.

Fourth Edition, September 2004

This edition applies to Version 1 Release 6 of z/OS® (5694-A01), Version 1 Release 6 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC35-0421-02.

IBM® welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcfs@us.ibm.com
World Wide Web: http://www.ibm.com/servers/eserver/zseries/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
- Title and order number of this document
- Page number or topic related to your comment

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

© Copyright International Business Machines Corporation 1984, 2004. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures xi

Tables xiii

About This Document xv
Major Divisions of This Document xv
Required product knowledge xvi
Referenced documents xvi
Accessing z/OS DFSMS documents on the Internet xvii
Using LookAt to look up message explanations xvii
Accessing z/OS licensed documents on the Internet xviii

Summary of Changes xix
Summary of Changes for SC35-0421-03 z/OS Version 1 Release 6 xix
New Information xix
Summary of Changes for SC35-0421-02 z/OS Version 1 Release 5 xix
New Information xix

Part 1. Introduction to Storage Management with DFSMShsm 1

Chapter 1. Welcome to DFSMShsm . . . 3
Purpose of Space Management 3
Purpose of Availability Management 4
What You Are Expected to Do 4
Tasks for Managing Storage with DFSMShsm . . 5
Overview of DFSMShsm Functions 5
DFSMShsm Functions for SMS-Managed Volumes 7
DFSMShsm Functions for Non-SMS-Managed Volumes 11
Devices Supported by DFSMShsm 12
How to Use The Example System 13
Multiple z/OS Images in the Example System . . 14
DASD Storage in the Example System 14
Tape Devices in the Example System 15
Other Assumptions in the Example System . . . 16

Part 2. How to Control Space Management 19

Chapter 2. Space Management of SMS-Managed Storage 21
Space Management Functions 21
Automatic Primary Space Management 21
Automatic Secondary Space Management . . . 21
Automatic Interval Migration 22
Automatic Recall 22
Command Space Management 22
Tasks for Automatic Space Management 22
Specifying the Storage to Be Managed and the Space Requirements 22
Specifying the DFSMShsm Hosts That Process Each Storage Group 25
Specifying How Groups of Data Sets Are Managed 25
Specifying Expiration Attributes 25
Specifying Partial Release 26
Specifying Migration Attributes 27
Specifying GDG Management Attributes 27
Using the Default Management Class 28
Management Class Examples 29
Specifying When Automatic Primary Space Management Can Process 31
Specifying When Automatic Primary Space Management Starts 31
Specifying When Automatic Secondary Space Management Can Process 31
Specifying When Automatic Secondary Space Management Starts 31
Specifying Automatic Interval Migration 32
Specifying How to Migrate Data Sets to Migration Level 2 32
Specifying How to Do Extent Reduction 34
Specifying How to Scratch Expired Data Sets . . . 34
Specifying the Maximum Automatic Primary Space Management Tasks 35
Specifying the Maximum Automatic Secondary Space Management Tasks 35
Specifying the Maximum Automatic Interval Space Management Tasks 36
Specifying Recall Tasks 36
Specifying Small Data Set Packing 36
Specifying the First Qualifier for Names of Migrated Data Sets 37
Specifying How to Do Migration Cleanup 38
Identifying Migration Level 2 Volumes to DFSMShsm 39
Specifying Duplexing for Migration Level 2 Tapes 39
Integrity Age for Space Management 40
Automatic Space Management Operation 41
Automatic Space Management 41
Starting Automatic Secondary Space Management 43
Starting Automatic Primary Space Management 49
Automatic Primary Space Management 50
Automatic Interval Migration 64
Automatic Recall 65
Additional Space Management during Migration and Recall When DFSMShsm Is the Data Mover . 70
Command Space Management 71
Deleting Migrated Data Sets 71
Migrating an Individual Data Set 73
Migrating an SMS-Managed Volume 74
Recalling SMS-Managed Data Sets 75
Recalling Direct Access Data Sets 75

© Copyright IBM Corp. 1984, 2004

Reconnecting Data Sets with Fast Subsequent Migration 76

Chapter 3. Space Management of Non-SMS-Managed Storage 79

Differences in What the Example System Shows . . 79
Subtasks for Space Management of Non-SMS-Managed Storage 79
Specifying the Age for Deleting List and Utility Data Sets 80
Specifying the Minimum Age for Migrating Data Sets 81
Specifying When Data Sets Become Eligible for Level 1 to Level 2 Migration 81
Specifying Recall Characteristics for the Type of Volume 81
Defining Primary Volumes to DFSMShsm . . . 82
Attributes of Primary Volumes 84
Defining Pools of Volumes 85
Pools in Systems Running under JES2 87
Pools in Systems Running under JES3 87
Automatic Space Management Processing for Non-SMS-Managed Storage 90
Automatic Secondary Space Management . . . 91
Automatic Primary Space Management 92
Automatic Interval Migration 101
Recall 101
Command Space Management 106
Command Volume Space Management 107
Command Volume Space Management (Primary) 107
Controlling Migration of Data Sets and Volumes 107
Recalling a Data Set to a Specific Volume . . . 108
Recalling an SMS-Managed Data Set to a Non-SMS-Managed Volume 108
Recalling an SMS-Managed BDAM Data Set to a Non-SMS-Managed Volume 108
Recalling an SMS-Managed Sequential Extended-Format Data Set to a Non-SMS-Managed Volume 108

Chapter 4. Other Space Management Considerations. 109

Common Recall Queue 109
Connecting to a Common Recall Queue List Structure 111
Populating the Common Recall Queue 111
Selecting Requests from the Common Recall Queue 112
Processing Recall Requests on the Common Recall Queue 113
Disconnecting from the Common Recall Queue 113
Processing of Common Recall Queue Requests during Shutdown 113
Deleting the Common Recall Queue Structure 114
Processing Failures in a Common Recall Queue Environment 114
Processing Priority of Recalls and Deletions . . . 116
Migration Volume Contention Priorities 117
Tape Migration Level 2 Volume Contention . . 117
Specifying the Desired Number of ML2 Partial Tapes 123
SDSP Data Set Contention 125
Multitasking Considerations for SDSP Data Sets 128
Preventing the Processing of Data Sets during Space Management 131
Deferred Generation Data Sets. 131
Security Considerations for Automatic Space Management 132
Space Management of Data Sets with Discrete Data Set Profiles 133
Password Protection of Data Sets on DASD . . 134
Access Method Services Considerations 135
GUARANTEED SPACE Attribute Considerations 135
Results of Recalling GUARANTEED SPACE VSAM Data Sets 136
Possible Problems 136
DFSMShsm Retrieval of Management Classes and Storage Groups. 138
Deletion of Dynamic I/O Device Considerations 138
Space Management Considerations for Sequential Extended-Format Data Sets. 139
Volume Space Management Processing 139
Automatic Primary Space Management Processing 139
Recall Processing 139

Chapter 5. Space Management Procedures 141

Interrupting and Restarting Automatic Primary or Secondary Space Management, or Interval Migration 141
When Automatic Primary or Secondary Space Management or Interval Migration Needs to be Restarted 141
Automatic Restart of Automatic Primary or Secondary Space Management. 142
Automatic Restart of Interval Migration . . . 143
Manual Restart of Automatic Primary or Secondary Space Management. 143
Defining a New Automatic Primary or Secondary Space Management Processing Window 144
When Automatic Primary or Secondary Space Management Is Running 144
When Automatic Primary or Secondary Space Management Is Not Running 145
Allowing Another Host to Take Over Secondary Space Management Functions 145

Part 3. How to Control Availability Management 147

Chapter 6. Availability Management of SMS-Managed Storage 151

What the Example System Shows 151
Automatic Incremental Backup 151
Automatic Full-Volume Dump. 151
Command Availability Management. 151
Tasks for Automatic Availability Management . . 152

Dump Tasks 152
Specifying Which Volumes to Dump and the Dump Classes to Use 152
Specifying When Automatic Dump Processing Starts 153
Specifying the DFSMSdss DASD I/O Buffering Technique to Use for Dump 154
Specifying Concurrent Copy for Volume Dumps 154
Specifying the Maximum Number of Dump Tasks 155
Specifying the Day on Which a Dump Occurs 155
Specifying the Characteristics of Dump Classes 156
Defining Dump Volumes to DFSMShsm . . . 162
Backup Tasks 163
Specifying Which Storage Group Has Automatic Backup Performed. 163
Specifying How to Manage Groups of Data Sets 164
Specifying Whether to Use Concurrent Copy 164
Specifying Frequency of Backup 165
Guaranteeing Frequency of Backup 165
Specifying the Maximum Number of Backup Versions to Keep 166
Specifying How Long to Keep Backup Versions 167
Specifying Who Can Back Up Data Sets by Command 168
Specifying Whether to Perform Automatic Backup 169
Example Management Classes. 169
Specifying the Start Time for Automatic Backup 172
Specifying Whether to Perform Volume Backup and the Kind of Device to Use. 173
Specifying the First Qualifier for Names of Backup Versions 173
Specifying the Maximum Number of Volume Backup Tasks in Each Host 174
Specifying the Data Set Backup by Command Environment 174
Specifying Whether to Back Up RACF Discrete Data Set Profiles 174
Specifying Whether DFSMShsm Performs Spill Processing 175
Specifying Whether to Back Up Only Changed Data Sets 175
Specifying How DFSMShsm Should Respond to a Data Set in Use 175
Specifying the Days on Which Automatic Backup Occurs 176
Defining Backup Volumes to DFSMShsm . . . 177
Specifying Duplexing for Backup Tapes . . . 178
Automatic Availability Management Operation . . 178
Dump Processing 178
Deleting Expired Dump Copies 179
Defining What to Do with Tapes Marked Unavailable during DUMP Processing 180
Dumping DFSMShsm-Managed Volumes . . . 180
Deleting and Uncataloging Excess Dump VTOC Copy Data Sets. 182
Automatic Backup Processing 182
Phase 1: Backing Up the Control Data Sets . . 183
Phase 2: Moving Backup Versions 184
Phase 3: Backing Up Migrated Data Sets . . . 185
Phase 4: Backing Up SMS-Managed Volumes from Storage Groups with the Automatic Backup Attribute 186
Selecting Daily Backup Volumes 191
Backup Volume Spill Process 200
Selecting Spill Backup Volumes 201
Initial Spill Backup Volume Selection (DASD and Tape) 202
Subsequent Spill Backup Volume Selection (DASD and Tape) 203
Cleanup Processing 204
Backup Volume Cleanup Process 205
Releasing a Backup Volume 206
Automatic Backup Example 206
Command Availability Management. 207
Dumping a Volume 207
Dumping the Volumes in One or More Storage Groups 208
Backing Up an Individual Data Set 208
Using the Data Set Backup by Command Capabilities 209
Backing Up All Data Sets on One or More Volumes 215
Backing Up All Data Sets in One or More Storage Groups. 215
Backing Up All Eligible Changed Data Sets on One or More Volumes 216
Backing Up All Eligible Changed Data Sets in One or More Storage Groups 216
Deleting Expired Backup Versions from the Backup Volumes 216
Deleting Backup Versions of Specific Data Sets 222
Command Recovery and Restore 224
Data Set Recovery and Restore 224
Restoring a Data Set from a DFSMShsm Dump Copy 235
Restore of VSAM Data Sets. 237
Managing Volume Backups with Fast Replication 238
Using the Copy Pool Construct for Fast Replication Setup 238
Using the Copy Pool Backup Storage Group Type for Fast Replication Setup 239
Using the Extend and Overflow Storage Group Types with Fast Replication 239
Preparing for the Fast Replication Backup . . . 240
Creating a Fast Replication Backup 240
Recovering a Fast Replication Backup 242
Withdrawing DFSMShsm Background Copies 244
Holding and Releasing the Fast Replication Functions. 244
Deleting Unneeded DFSMShsm Fast Replication Backup Versions 244
Obtaining DFSMShsm Fast Replication Backup Information 245
Dumping Fast Replication Target Volumes. . . 246
Preventing DFSMShsm Functions from Processing Target Volumes 247
Other Copy Pool Considerations 247
Volume Recovery and Restore 249
Specifying the Maximum Number of Data Set Recovery Tasks in Each Host 250

Full-Volume Restore with Update 251
Full-Volume Restore from a Dump Copy . . . 254
Volume Recovery from Incremental Backup Versions 256
Processing Priority of Volume Recoveries . . . 258

Chapter 7. Availability Management of Non-SMS-Managed Storage 261
Subtasks for Availability Management of Non-SMS-Managed Storage 261
Specifying How Frequently Data Sets Are Backed Up 261
Specifying How Many Versions to Keep for Each Backed Up Data Set 262
Specifying Which Volumes Are Backed Up Automatically 262
Specifying Which Volumes Are Dumped Automatically 263
Automatic Availability Management of Non-SMS-Managed Volumes 265
Backing Up Cataloged and Uncataloged Data Sets with the Same Name 265
Command Availability Management for Non-SMS-Managed Storage 267
Dumping a Volume 267
Changing the Backup Conditions for a Data Set 267
Backing Up an Individual Data Set 268
Inline Backup of an Individual Data Set . . . 268
Backing Up All Primary Volumes. 268
Backing Up All Changed Data Sets on Volumes 268
Deleting Expired Backup Versions of Non-SMS-Managed Data Sets 269
Deleting Backup Versions of Specific Data Sets 270
Recovering Specific Data Sets 271
Recovery of VSAM Data Sets 273
Data Set Restore from a DFSMShsm Dump Copy 274
Data Set Recovery of Retired Data Sets 276
Combining Full-Volume Restore with Volume Recovery 276
Volume Recovery from Incremental Backup Versions 276

Chapter 8. Aggregate Backup and Recovery Support (ABARS) 281
How Aggregate Backup Works 284
Specifying Data Sets to Be Backed Up 284
Backing Up Applications 286
ABACKUP Command 287
Example of Data Sets Associated with an Aggregate Group 287
Filtering for Generation Data Sets 288
Preparing for Aggregate Backup 289
SMS Requirements during Aggregate Backup 290
Creating Selection Data Sets 291
Creating an Instruction Data Set 297
Defining an Aggregate Group 298
Management Class Considerations 301
DFSMShsm Coexistence Considerations for Aggregate Groups and Management Classes . . 302
Protecting Aggregate Backup Tape Volumes . . 303
Checking Authorization during Aggregate Backup 304
Authorizing Comprehensive ABACKUP Command Authority 304
Authorizing Restricted ABACKUP Command Authority. 305
Using the Aggregate Backup Installation Exits 306
Ensuring That the Necessary Volumes Are Online. 306
Defining SETSYS Parameters for Aggregate Backup 306
Determining the Unit Type for the Control and Data Files 309
Minimizing Data Set Contention 309
Data Sets That DFSMShsm Dynamically Allocates during Aggregate Backup Processing . 310
Performing Aggregate Backup. 310
ABACKUP Command 311
Issuing the ABACKUP Command with the VERIFY Parameter 314
Issuing the ABACKUP Command with the EXECUTE Parameter 315
Aggregate Backup Output 316
Data Files 316
Control File 316
Instruction/Activity Log File 317
Activity Log. 317
ABACKUP CPU Time and Accounting Codes 319
DFSMShsm Coexistence Considerations for New ABACKUP Output Data Sets 320
Expiring ABR Records in the BCDS 320
ABACKUP Considerations 322
ABARS User Catalog and Catalog Alias ABACKUP Support 323
Aggregate Backup Assist 324
How Aggregate Backup Assist Works 324
What Aggregate Backup Assist Produces . . . 324
Special Considerations for ABARS 325
Authorization and Security Requirements . . . 325
Input Requirements 326
SMF Data 326
Aggregate Job Names 326
Storage Requirements 326
DD Statements Used by ABA 327
Sample JCL 327
Sample Output 328
How Aggregate Recovery Works 328
Recovered Aggregate Data Sets 330
Example Aggregate Recovery Site 330
Preparing for Aggregate Recovery 331
Authorization Checking during Aggregate Recovery 333
Authorizing Comprehensive ARECOVER Command Authority 333
Authorizing Restricted ARECOVER Command Authority. 334
Considering Storage Management Subsystem (SMS) Requirements 334
Determining Aggregate Recovery Site Constraints 336

Retrieving the Instruction Data Set 336
Creating an ABR Record in the BCDS 338
Issuing the DEFINE ARPOOL Command . . . 338
Defining SETSYS Parameters for Aggregate Recovery 341
Defining the Model Entity 344
Using the Aggregate Recovery Installation Exits 344
Processing Generation Data Sets 345
GDSs From Level 0 DASD in the Include List 345
Migrated GDSs in the Include List 346
GDSs in the ALLOCATE List 347
GDS Processing Order 347
Processing Migrated Data Sets in the INCLUDE List. 347
Processing VSAM Data Sets in the ALLOCATE List 348
Catalog Considerations during ARECOVER Processing 349
Performing Aggregate Recovery 350
ARECOVER Command 350
Issuing the ARECOVER Command with the PREPARE Parameter 357
Issuing the ARECOVER Command with the VERIFY Parameter 358
Issuing the ARECOVER Command with the EXECUTE Parameter 358
Recovering the ABARS Instruction Data Set . . 359
Backing Up and Recovering the ABARS Activity Log. 360
Conflict Avoidance and Conflict Resolution Processing Options 361
ARECOVER Data Set Name Conflict Resolution 361
Holding and Releasing ABARS Functions . . . 369
Displaying SMS Class Definitions during ABACKUP and ARECOVER 370
ABARS User Catalog and Catalog Alias ARECOVER Support 371
Data Sets That Are Dynamically Allocated during Aggregate Recovery Processing 372
Aggregate Recovery Output 372
Activity Log. 373
ARECOVER CPU Time and Accounting Codes 376
Using ISMF Panels to Perform Aggregate Backup 376
Using ISMF Panels to Perform Aggregate Recovery 382
Using the NetView File Transfer Program to Transmit Aggregate Backup Files 391
Using NetView File Transfer Program Panels 392
Issuing the ARECOVER Command When Using NetView FTP 396
ABARS Considerations for Sequential Data Striping 396
Aggregate Backup Processing 396
Aggregate Recovery Processing 397
Aggregate Roll-Off Processing. 398
Defining GDGs at the Recovery Site. 398
Ensuring Backup Copies of ARECOVERed SMS-Managed Data Sets Exist at the Recovery Site . 399

Chapter 9. Making Disaster Backup Copies of DFSMShsm-Owned Tape Volumes 401
Planning Disaster Recovery. 401
Disaster Backup Facilities 402
Making Copies of DFSMShsm-Owned Tape Volumes 402
Converting Partial Tapes to Full Tapes 407
Replacing an Original with an Alternate Tape Volume 407
Creating Duplex Tapes Off Site 408
Sending Alternate Tapes Off Site 408
Managing Alternate Volumes 408
Disaster Alternate Volumes. 409
Considerations When Using Disaster Alternate Volumes 411

Chapter 10. Other Considerations for Availability Management 413
Additional Considerations for Backup 413
Backup Volume Contention Resolution 413
Implications of Redefining the Backup Cycle 415
Inline Backup 415
Preventing Backup Processing When Using the HOLD BACKUP Command 419
GUARANTEED SPACE Attribute. 420
Data Set Backup with Concurrent Copy . . . 423
Additional Considerations for Dump 424
Dumping DFSMShsm-Owned Volumes. . . . 424
Dumping Control Data Sets 424
Dumping the Journal Data Set. 424
Preventing Dump Processing 425
Dumping All Allocated Space 425
Invoking Full-Volume Dump Compression . . 426
Bypassing Enqueue Processing between Full-Volume Dump and DFP Functions. . . . 426
Volume Recovery and Restore Considerations . . 426
Restoring Data Sets or Volumes Directly with DFSMSdss 427
Requesting a List of Dump Volumes for a Dump Copy 427
Determining the Dump Copy Data Set Name 427
Invoking the DFSMSdss RESTORE Function . . 427
Preventing Recovery and Restore Processing When Using the HOLD RECOVER Command . 428
Using the Date-Last-Referenced Field for Recovery 428
Deferred Generation Data Sets. 428
Security Considerations for Availability Management 429
Authorization Checking during Recovery . . . 429
Recovering or Restoring RACF-Indicated Data Sets 430
Maintaining Password Protection for Recovered Data Sets. 432
DFSMShsm Retrieval of Management Classes . . 433
Availability Management Considerations for Extended SAM and Extended and Compressed VSAM Data Sets 434

Availability Management Considerations for Sequential Data Striping 434
Backup Processing. 434
Recover and Restore Processing 435

Chapter 11. Availability Management Procedures 437
Backup and Recovery Procedures. 437
Interrupting and Restarting Backup Processing 437
Defining a New Automatic Backup Start Window 439
Backing Up and Recovering a Catalog 440
Full-Volume Dump Procedures 441
Interrupting and Restarting Automatic Dump Processing 442
Defining a New Automatic Dump Start Window 443

Part 4. DFSMShsm Support Activities. 445

Chapter 12. Maintaining DFSMShsm Control Data Sets 447
Tasks for Maintaining the Control Data Sets . . . 447
Specifying How DFSMShsm Backs Up the Control Data Sets 448
Allocating Backup Data Sets 448
Choosing the Data Mover to be Used 448
Specifying How Many Backup Versions to Keep 450
Specifying the Names of the Backup Data Sets 451
Specifying the Device Type for Backup of the Control Data Sets 451
Invoking the Control Data Set Backup Exit (ARCCBEXT) 453
Backing Up to Multiple DASD Data Sets 453
Backing Up to Multiple Tape Data Sets. 455
Multiple DFSMShsm-Host Considerations. . . . 456
Multicluster CDS Considerations 456
Nulling the Journal 457
Changing the Names of the Backup Data Sets . . 457
Changing BACKUPDEVICE from Tape to DASD 458
Changing BACKUPDEVICE from DASD to Tape 459
Using the QUERY Command to Get CDS Information 459
Backing Up the Control Data Sets Manually . . . 460
Possible Failures during Backup 461
Growth of Your Control Data Sets 461
Other Failures 462
Reasons For Recovering the Control Data Sets . . 462
I/O Errors 463
Recovering Control Data Sets Using the Enhanced CDS Recovery Function 463
Requirements for Using the Enhanced CDS Recovery Function. 463
Implementing Enhanced CDS Recovery using JCL. 464
Recovering Multicluster Non-Key-Range CDSs . . 469
CDS Recovery Procedure using CDSVERSIONBACKUP with UPDATEC 469
Updating Recovered Control Data Sets using UPDATEC 471
VSAM Considerations for the Control Data Sets 472
Determining If Control Data Sets Are Accurate . . 472
Repairing Errors in the Control Data Sets 473
Recovering from a Journal I/O Error 473
Moving the Journal 474
Steps to Ensure That the Control Data Sets Are Accurate 475
CDS Backup and Recovery Examples 475
Recovery Examples using the Enhanced CDS Recovery Function. 475

Chapter 13. Managing DFSMShsm Media. 483
Moving All Data Sets from One Level 0 Volume to Other Level 0 Volumes 483
Processing SMS-Managed Data Sets 484
Processing Non-SMS-Managed Data Sets 484
Examples of Moving Data Sets between Level 0 Volumes 485
Moving Some Data Sets from Migration Level 1 Volumes 488
Using the FREEVOL Command for Migration Level 1 Volumes 488
Using the MIGRATE Command for Migration Level 1 Volumes 489
Deleting Non-SMS-Managed Data Sets from Migration Volumes 490
Moving All the Data Sets from a Migration Level 1 Volume 490
Processing for SMS-Managed Data Sets. 491
Processing for Non-SMS-Managed Data Sets . . 491
Moving Data Sets from a DASD Migration Level 2 Volume to Other Migration Level 2 Volumes . . . 492
Removing Residual Dump and Backup VTOC Copy Data Sets from Migration Level 1 Volumes . 493
Preventing Further Migration to a DASD Migration Volume 493
Moving Data Sets from DASD Backup Volumes 493
Removing Volumes from DFSMShsm Management or Control 494
Removing Dump Copies and Dump VTOC Copy Data Sets from DFSMShsm. 494
JES3 Considerations 495
Consolidating Valid Data on Tape Volumes . . . 495
Defining the Parameters of the Recycle Process 496
Initiating Recycle Processing 497
PERCENTVALID Calculation 500
Selecting Tape Volumes during Recycling . . . 500
Making Cartridge-Type, Single-File-Format Tapes from Reel-Type or Multiple-File-Format Tapes 501
Limiting Data that is Written to a Cartridge-Type Device. 501
Specifying Capacity Utilization 502
Considerations for Using the 3592 Model J Tape Drive 502
Limiting the Number of ML2 and Backup Data Sets that Span Tape Volumes 504

How to Handle Inconsistencies between Tape Media Contents and OCDS TTOC Records . . . 505
Recovering a Damaged DFSMShsm Single-file Migration Level 2 Volume 506
Making Migration Level 2 Key Range Volumes Available 506
Recovering a Migrated Data Set 506
Recovering a Migration Volume 507
Procedure 1 507
Procedure 2 509
Recovering Backup Versions from a Damaged Migration Level 1 Volume 510
Returning Dump Tapes to Use. 511
Managing SDSP Data Sets 511
Reorganizing SDSPs 511
SDSP Backup and Recovery 512
Removing SDSPs 513
Protecting DFSMShsm-Owned Tape Volumes. . . 514
Converting to a New Tape Technology 514
Requesting DFSMShsm Service from Your Applications. 515

Chapter 14. Obtaining Information from DFSMShsm 517
Obtaining Information from the Control Data Sets 517
Obtaining Backup Volume Information 518
Obtaining Data Set Information 518
Obtaining Copy Pool Information 519
Obtaining Dump Class Information 520
Obtaining Dump Volume Information 520
Obtaining DFSMShsm Host Serialization Information 522
Obtaining Lists of Backup and Dump Parameters and Backup Volsers 522
Obtaining Information about Migration and DFSMShsm-Managed Volumes 522
Obtaining DFSMShsm-Managed Volume Information 523
Obtaining Tape Volume Information. 524
Obtaining User Authorization Information . . 526
Determining Which Tape Dump Volumes Are Available for Output during Dump Processing . 526
Obtaining Information about the DFSMShsm Address Space 526
Determining Which Tape Volumes Are Available for Output 529
Determining Which Tape Backup Volumes Are Available for Output 530
Determining Which Tape Migration Level 2 Volumes Are Available for Output 530
Obtaining Statistical Information about DFSMShsm Operations 530
Using the Problem Determination Aid to Trace DFSMShsm Processing 531
Specifying to DFSMShsm to Use the PDA . . . 532
Using the TRAP Command to Obtain Information about Conditions in the System When an Error Occurs 532
Using DEBUG to Determine the Results of DFSMShsm Processing without Moving Data Sets . 533

Chapter 15. DFSMShsm Tape Libraries 535
Automated Tape Library 535
Manual Tape Library 535

Chapter 16. Using DFSMShsm with DFSMSrmm 537
Retaining DFSMShsm Tape Volumes by Expiration Date 537
Retaining DFSMShsm Tape Volumes by DFSMSrmm Vital Record Processing. 538
Tape Security When Using DFSMShsm and DFSMSrmm 538
Recommendations 538

Chapter 17. Operating Procedures for DFSMShsm 539
Partitioning DFSMShsm Functions 539
Starting DFSMShsm 540
Stopping DFSMShsm 542
Shutdown Processing on the Common Recall Queue 543
Preventing DFSMShsm Functions from Running 543
Allowing Previously Prevented DFSMShsm Functions to Run 547
Canceling DFSMShsm Queued Requests 552
Printing DFSMShsm Log Information 553
Restarting DFSMShsm after an Abnormal End . . 562
Running Automatic Functions Concurrently . . . 563

Chapter 18. Purposes of DFSMShsm Commands 567
Command Function Changes for SMS 569
Long-Running Commands 570

Chapter 19. RACF Considerations . . 573
RACF Basics. 573
RACF Data Set Profiles 573
RACF Indication 574
Always-Call Support 575
RACF Protect-All Environment 575
RACF Environment for DFSMShsm 576
DFSMShsm and RACF Concepts 576
The RACF User ID for DFSMShsm 578
RACF Started Task Table 579
Always-Call Support for Migrated Data Sets and Backup Copies 579
Migration and Recall from DASD 580
Backup and Recovery from DASD 580
Aggregate Backup and Recovery Data Set Security 581
Backing Up RACF-Indicated Data Sets 581
Authorizing the Use of Concurrent Copy for Data Set Backup Commands 581
Small Data Set Packing Data Set Security 582
Uncataloging Migrated Data Sets. 582
Controlling Command Access with RACF FACILITY Class 582

Part 5. DFSMShsm Data Recovery Scenarios 583

Chapter 20. Data Recovery Scenarios 585
Case 1: Damaged CDS, Full Journal 585
Case 2: Damaged Journal, Undamaged CDS . . . 588
Case 3: Full Journal, Undamaged CDS 589
Case 4: Structurally Damaged CDS, Missing Journal Records 589
Case 5: Time Sequence Gaps in CDS Records, Journal Records Also Missing 589
Case 6: Overwritten Migration Tape 590
Case 7: Overwritten Backup Tape. 594
Case 8: Damaged ML1 Volumes 598
Case 9: Reestablish Access to Previously Deleted Migrated Data Sets (No Backup Exists, ML2 Only) . 604
Case 10: Restoring Stacked Output Tapes with a Down-Level MVS Image 606
Case 11: Correcting Errors within the Common Recall Queue 607
Case 12: Recovering a Deleted ML1 Data Set without a Backup 608

Appendix A. Consolidated Commands, Storage Group Definitions, and Management Classes for the Example System 611
ADDVOL Command 611
DEFINE Command 614
SETSYS Command 615
Storage Groups. 617
Management Classes 618

Appendix B. Preventive Methods for Avoiding Data Loss. 621
Steps for Preventing a Full Journal (Damaged CDS) 621
Preventing a Full Journal (Undamaged CDS) . . . 621
Preventing Structural Damage to CDS and Missing Journal Records 622
Preventing Time Sequence Gaps in CDS Records (Journal Records Also Missing) 622
Preventing Overwritten Migration Tape or Overwritten Backup Tape 623
Prepare for Recovery of a Head Crash on an ML1 Volume 623
Reestablishing Access to Previously Deleted, Migrated Data Sets (No Backup Exists, ML2 Only) . 623
Restoring Stacked Output Tapes with a Down-Level MVS Image 624

Appendix C. Linux-z/OS DFSMShsm Dump or Restore HOW-TO. 625
How DFSMShsm Supports Linux. 625
Requirements 625
Hardware Environment 625
DASD Background 626
Linux Requirements 626
Linux Disk Utilities 626
Authorization Requirements 627
Setting Up Your Dump Environment for Linux Volumes 627
Performing Only Dumps 628
When Do You Want Your Dumps Performed? 628
How Long Do You Want to Keep Your Dumps? 628
How Many Dumps Are Stacked on Each Dump Tape? 628
How Many Tape Drives Do You Have?. 629
Recovering Your Data 629
Recovering a Volume with DFSMShsm 629
Recovering a Volume with DFSMSdss 630
Recovering a Partition with DFSMShsm . . . 630
Recovering a Partition with DFSMSdss 631
Example Dump Environment 632
How to Define Your Dump Cycle 633
How to Define Your Dump Classes 633
How to ADDVOL Your Volumes 634
How to Start Automatic Dumping 634
Recovering from a Dump 634
Submitting DFSMShsm Commands Via JCL and FTP 636
The DFSMShsm LIST Command 637
Naming Conventions. 638
Data Set and Partition Naming Conventions . . 638
Volser Naming Conventions 638
References 639

Appendix D. Accessibility 641
Using assistive technologies 641
Keyboard navigation of the user interface 641
z/OS information 641

Notices 643
Programming interface information 644
Trademarks 644

Glossary 645

Index 661

Figures

1. Data Set Flow in DFSMShsm 7
2. Example System for SMS-Managed Volumes 13
3. Example System for Non-SMS-Managed Volumes 14
4. Comprehensive Example System 17
5. Storage Group Attributes for Space Management 24
6. DFSMShsm Commands for Space Management with SMS 40
7. SMS-Managed Volumes before start of Automatic Space Management 42
8. SMS-Managed Volumes after Migration Cleanup. 45
9. SMS-Managed Volumes after Level 1 to Level 2 Migration 49
10. SMS-Managed Volumes after DFSMShsm-Managed Volume Management Pass 1 55
11. Component Name Limitations when Recalling Migrated VSAM Data Sets 59
12. SMS-Managed Volumes after DFSMShsm-Managed Volume Management Pass 2 63
13. Multiple-Processing Units in a JES3 and DFSMShsm Environment 90
14. Example System for Non-SMS-Managed Volumes before Automatic Space Management . 91
15. Example System for Non-SMS-Managed Volumes after Primary Volume Management Pass 1 94
16. Component Name Limitations When Recalling Migrated VSAM Data Sets 97
17. Example System for Non-SMS-Managed Volumes after Primary Volume Management Pass 2 100
18. Example System for Non-SMS-Managed Volumes after Undirected Recall of Data Sets . 106
19. The Common Recall Queue 110
20. SDSP Contention Environment. 126
21. Part 1—Ideal Multitasking Migration 129
22. Part 2—Recall Processing Has a Higher Priority than Migration 130
23. Part 3—All SDSP Data Sets Are Full 130
24. Storage Group Dump Attributes for Availability Management. 153
25. DFSMShsm Automatic Dump Commands for SMS-Managed Storage 162
26. Storage Group Attributes for Backup 164
27. Backup Attributes for STANDMC 169
28. Backup Attributes for GDSMC 170
29. Backup Attributes for LARGEMC 171
30. Backup Attributes for NOMIGMC 172
31. DFSMShsm Automatic Backup Commands for SMS-Managed Storage 177
32. Aggregate Backup and Recovery Support 282
33. Aggregate Backup Information Flow 286
34. Example of an Instruction Data Set 298
35. Aggregate Group ISMF Panel 1 300
36. Aggregate Group ISMF Panel 2 301
37. Aggregate Backup Invocation 311
38. Example of an Aggregate Backup Activity Log 319
39. Example of DISPLAY ABARSVERSIONS Parameter Output 322
40. Aggregate Recovery Information Flow 329
41. Example of a Retrieved Instruction Data Set 337
42. Expected Results for Level 0 Data Sets in the INCLUDE List (Processed by DFDSS) . . . 346
43. Expected Results for Migrated Data Sets in the INCLUDE List 347
44. Aggregate Recovery Invocation 350
45. Conflict Resolution Data Set 363
46. Example of an Aggregate Recovery Activity Log 374
47. Aggregate Group Application Selection Panel 377
48. Aggregate Group List Panel. 378
49. Aggregate Group Backup Panel 378
50. Aggregate Group Application Selection Panel 380
51. Aggregate Group Backup Panel 381
52. Aggregate Group Application Selection Panel 383
53. Aggregate Recovery Panel DFQDCAG5 384
54. Aggregate Recovery Panel DFQDCAG6 386
55. Aggregate Recovery Panel DFQDCAG7 387
56. Aggregate Recovery Panel DFQDCAG8 388
57. Aggregate Recovery Panel DFQDCAG9 389
58. Aggregate Recovery Panel DFQDCAGA 390
59. Aggregate Recovery Panel DFQDCAGB 391
60. NetView FTP Main Task Selection Panel 392
61. Main Transfer Parameters Panel 393
62. Main Data Set Parameters Panel 394
63. Additional Transfer Parameters Panel 394
64. Additional Parameters for the Sending Data Set Panel 395
65. Example 2: Additional Parameters for the Receiving Data Set Panel. 395
66. Sample Batch Job Used to Transmit Aggregate Backup Files 396
67. Example of Inline Backup Job Stream 418
68. Example of Messages Resulting from Inline Backup Job Stream. 419
69. Example System for MIGRATE CONVERT 486
70. SMS Volume Status 487
71. DFSMShsm Log and Edit Log Overview 554
72. Example of a List from the ARCPRLOG Program 560
73. Example of a List from the ARCPEDIT Program 561
74. Example of a Summary List from the ARCPEDIT Program 562
75. Consolidated Storage Group Attributes 618
76. Consolidated Management Classes 619

© Copyright IBM Corp. 1984, 2004

Tables

1. AUTO MIGRATE Attribute Values 23
2. Data Set Characteristics for Automatic Space Management Example 42
3. Standard Interval Migration Processing Compared to TMM Processing 65
4. Recall Destination Summary 66
5. Commands for Deleting Migrated Data Sets 72
6. Data Set Characteristics for Automatic Space Management Example (Non-SMS-Managed Volumes) 91
7. Current Backup Version of Data Sets 96
8. Volume Contention in a Single DFSMShsm-Host Environment 120
9. Volume Contention in a Multiple DFSMShsm-Host Environment 122
10. SDSP Contention in Single and Multiple DFSMShsm-Host Environments 126
11. RACF Authority Required for Space Management Functions 133
12. Restarting Automatic Primary or Secondary Space Management 141
13. Generations of Example Dumps 161
14. Data Set Characteristics for Availability Management Example 206
15. Determining Data Set Sizes and Their Task Targets 212
16. Backup Version Data Set Selection for Recovery 257
17. Backup Version Data Set Selection for Recovery 277
18. Data Sets at the Example Aggregate Backup Site 288
19. Aggregate Backup Checklist 289
20. Acceptable Generic Job Names 326
21. DD Statements Used by ABA 327
22. Aggregate Recovery Data Set Processing 330
23. Aggregate Recovery Checklist 332
24. Volume Contention in Single and Multiple DFSMShsm Host Environments 413
25. Resource Access Authority Required for DFSMShsm Functions 429
26. Recovering RACF-Protected Data Sets Backed Up under DFHSM before Version 2 Release 2.0 432
27. SETSYS Parameters Not Applicable to SMS-Managed Data Sets 570
28. DFSMShsm Commands Not Used for SMS Storage Management 570
29. RACF Authorization Required for DFSMShsm Functions 577

© Copyright IBM Corp. 1984, 2004

About This Document

This document describes the tasks that must be performed to use DFSMShsm™, a component of DFSMS that manages your storage in a hierarchical manner. It contains examples of the commands needed to perform the tasks, descriptions of the processing that DFSMShsm performs, and examples of the results of that processing.

This document is intended for use by people responsible for managing storage for a computer center. It is designed to be tutorial in nature. Information presented in one section is often based on knowledge obtained in a preceding section.

For information about the accessibility features of z/OS™, for users who have a physical disability, see Appendix D, "Accessibility," on page 641.

Major Divisions of This Document

This document is divided into five parts, which are divided into chapters. The divisions of this document are:

- **Part 1, "Introduction to Storage Management with DFSMShsm," on page 1** provides an overview of DFSMShsm.
  - **Chapter 1, "Welcome to DFSMShsm," on page 3,** which describes the functions of DFSMShsm and explains how to use this document.
- **Part 2, "How to Control Space Management," on page 19** introduces the tasks necessary to control space management functions:
  - **Chapter 2, "Space Management of SMS-Managed Storage," on page 21,** which describes the tasks to be performed and the results of processing for space management of data sets managed by the storage management subsystem (SMS).
  - **Chapter 3, "Space Management of Non-SMS-Managed Storage," on page 79,** which describes the tasks to be performed and the results of processing for space management of data sets not managed by SMS.
  - **Chapter 4, "Other Space Management Considerations," on page 109,** which describes aspects of space management that deserve consideration but are peripheral to the discussions in the two preceding sections.
  - **Chapter 5, "Space Management Procedures," on page 141,** which describes procedures for starting and stopping space management under varying conditions.
- **Part 3, "How to Control Availability Management," on page 147,** introduces the tasks necessary to control availability management functions:
  - **Chapter 6, "Availability Management of SMS-Managed Storage," on page 151,** which describes the tasks to be performed and the results of processing for availability management of data sets managed by SMS.
  - **Chapter 7, "Availability Management of Non-SMS-Managed Storage," on page 261,** which describes the tasks to be performed and the results of processing for availability management of data sets not managed by SMS.
  - **Chapter 8, "Aggregate Backup and Recovery Support (ABARS)," on page 281,** which describes the command-driven functions that back up and recover a user-defined group (aggregate group) of data sets.

© Copyright IBM Corp. 1984, 2004

- **Chapter 9, "Making Disaster Backup Copies of DFSMShsm-Owned Tape Volumes," on page 401,** which describes how to make backup copies of DFSMShsm-owned, cartridge-type, single-file-format tape volumes and how to recover those volumes.
- **Chapter 10, "Other Considerations for Availability Management," on page 413,** which describes aspects of availability management that deserve consideration but are peripheral to the discussions in the preceding sections.
- **Chapter 11, "Availability Management Procedures," on page 437,** which describes procedures for starting and stopping availability management under varying conditions.

- **Part 4, "DFSMShsm Support Activities," on page 445,** which introduces the activities necessary to maintain DFSMShsm-owned data sets and volumes.
  - **Chapter 12, "Maintaining DFSMShsm Control Data Sets," on page 447,** which describes the activities necessary to ensure that you can recover from damaged or lost DFSMShsm control data sets.
  - **Chapter 13, "Managing DFSMShsm Media," on page 483,** which describes the activities necessary to manage data stored on DFSMShsm-owned volumes.
  - **Chapter 14, "Obtaining Information from DFSMShsm," on page 517,** which describes the information that you can obtain from DFSMShsm about its operations and the data sets and volumes that it is managing.
  - **Chapter 15, "DFSMShsm Tape Libraries," on page 535,** which briefly explains DFSMShsm automated and manual tape libraries.
  - **Chapter 16, "Using DFSMShsm with DFSMSrmm," on page 537,** which explains the interaction between DFSMShsm and DFSMSrmm when managing DFSMShsm tape volumes.
  - **Chapter 17, "Operating Procedures for DFSMShsm," on page 539,** which describes procedures to be performed by the computing system operator.
  - **Chapter 18, "Purposes of DFSMShsm Commands," on page 567,** which briefly describes the purpose of various DFSMShsm commands.
  - **Chapter 19, "RACF Considerations," on page 573,** which discusses pertinent RACF® information. RACF is a component of the Security Server for z/OS. This information includes the use of discrete and generic profiles with DFSMShsm, and the use of FACILITY class profiles with regard to command control access in DFSMShsm.
- **Part 5, "DFSMShsm Data Recovery Scenarios," on page 583,** which describes real-life examples of data loss and recovery situations.

Required product knowledge

Readers of this publication should have a systems background, especially concerning TSO/E commands. The system programmer and storage administrator must understand the basic concepts of DFSMS before they read this document.

Referenced documents

The following publications are referenced in this document:

Publication Title	Order Number
Tivoli NetView File Transfer Program publications	SK3T-4270-07
z/OS DFSMS Access Method Services for Catalogs	SC26-7394
z/OS DFSMS: Implementing System-Managed Storage	SC26-7407

Publication Title	Order Number
z/OS DFSMS Installation Exits	SC26-7396
z/OS DFSMSdfp Storage Administration Reference	SC26-7402
z/OS DFSMSdss Storage Administration Guide	SC35-0423
z/OS DFSMSdss Storage Administration Reference	SC35-0424
z/OS DFSMShsm Implementation and Customization Guide	SC35-0418
z/OS DFSMShsm Managing Your Own Data	SC35-0420
z/OS DFSMShsm Storage Administration Reference	SC35-0422
z/OS DFSMSrmm Implementation and Customization Guide	SC26-7405
z/OS MVS System Messages, Vol 1 (ABA-AOM)	SA22-7631
z/OS Security Server RACF Command Language Reference	SA22-7687
z/OS Security Server RACF Security Administrator's Guide	SA22-7683
z/OS DFSORT Application Programming Guide	SC26-7523

Accessing z/OS DFSMS documents on the Internet

In addition to making softcopy documents available on CD-ROM, IBM provides access to unlicensed z/OS softcopy documents on the Internet. To view, search, and print z/OS documents, go to the z/OS Internet Library: http://www.ibm.com/eserver/zseries/zos/bkserv/

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM®, and VSE:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services running OMVS).
- Your Windows® workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Windows DOS command line.
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Accessing z/OS licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

`http://www.ibm.com/servers/resourcelink`

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code. [1]

To obtain your IBM Resource Link user ID and password, log on to:

`http://www.ibm.com/servers/resourcelink`

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

1. z/OS.e customers received a Memo to Licensees, (GI10-0684) that includes this key code.

Summary of Changes

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You may notice changes in the style and structure of some content in this document--for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Summary of Changes for SC35-0421-03 z/OS Version 1 Release 6

This document contains information previously presented in *z/OS Version 1 Release 5 DFSMShsm Storage Administration Guide* (SH35-0421-02).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- **Multiple tasks for DFSMShsm secondary space management**: This enhancement allows secondary space management functions to process in multiple tasks.
- **Fast Replication Dump**: DFSMShsm has extended the full volume dump function to dump the fast replication backup copies to tape. Recovery can be performed from the backup versions on the DASD target volumes or from the dump tapes. You can restore fast replication backup at the volume level from dumps or DASD copies, or at the copy pool level from DASD copies. Managed recovery at the data set level is not provided.
- **Support for new media types**: DFSMShsm supports new media types for the 3592 Model J tape drive, as follows:
  - MEDIA7 tapes are supported for all functions of DFSMShsm
  - MEDIA6 and MEDIA8 tapes are supported only for the DFSMShsm ABARS function.

z/OS DFSMShsm Data Recovery Scenarios is no longer available as a separate document. The information from that book is added to this book as Chapter 20, "Data Recovery Scenarios," on page 585.

Summary of Changes for SC35-0421-02 z/OS Version 1 Release 5

This document contains information previously presented in *z/OS Version 1 Release 3 DFSMShsm Storage Administration Guide* (SH35-0421-01).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

© Copyright IBM Corp. 1984, 2004

- **DFSMShsm RACF FACILITY class**: DFSMShsm now uses the RACF FACILITY class to control access to storage administrator commands and user commands. This allows a security administrator to identify which storage administrators can perform which tasks. It also allows control of which actions users can perform.
- **Fast Replication**: DFSMS has enhanced its use of volume-level fast replication to create backup versions for sets of storage groups. A set of storage groups is defined through a new SMS construct termed 'copy pool'. The new DFSMShsm FRBACKUP command creates a fast replication backup version for each volume in every storage group 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 new DFSMShsm FRRECOV command. This function enables the backup and recovery of a large set of volumes to occur within a small time frame. The term fast replication refers to the FlashCopy® function supported by IBM TotalStorage™ Enterprise Storage Server™ (ESS) disk and the SnapShot function supported by IBM RAMAC® Virtual Array (RVA) disk.
- **DFSMShsm Migration Installation Exits**: You can now invoke the ARCMDEXT and ARCMMEXT installation exits for a command migration request for a data set.
- **3590 Model H1*x***: The 3590 Model H1*x* software support provides customers with the following benefits:
  - A significant improvement in storage capacity through 384-track recording on the Model H, up from 256-track recording on the Model E1*x*. This reduces the amount of floor space needed for tape storage devices. Installations are thus able to cut the costs associated with tape storage and management.
  - Customers can now record 60 to 90 GB of compressed data on standard-length media and 30 GB of uncompressed data. They can record 120 to 180 GB of compressed data on extended-length media and 60 GB of uncompressed data.
  - Supports both standard- and extended-length 3590 media that are currently in use by Model B1*x* and Model E1*x* drives. This reduces storage costs by allowing installations to reuse media that they already own.
  - Emulates 3490E storage devices. This allows customers to add 3590 Model H1*x* devices to third-party libraries.
  - Permits the rewriting of a tape cartridge from the beginning, regardless of the format—128-track or 256-track—that had previously been written on the cartridge. The 3590 Model H1*x* is downwardly read compatible with tapes written by 3590 Model B1*x* and 3590 Model E1*x*. This reduces storage costs by allowing installations to reuse media that they already own.
  - Reports any previously recorded volume serial number in any interchange-and-overwrite scenario among any of the 3590 Models: B, E, or H. This allows installations to keep volume traceability records.
- **3592 Model J1A**: The IBM TotalStorage Tape System 3592 consists of the IBM TotalStorage Enterprise Tape Drive 3592 Model J1A drive and the IBM TotalStorage Enterprise Tape Controller 3592 Model J70 controller. The 3592 Model J tape drive uses the newly introduced EFMT1 recording technology to write to the newly introduced MEDIA5 tape cartridges.

  The 3592 Model J tape drives offer these features.
  - The 3592 Model J optionally provides performance scaling which uses only the first 60 GB of physical tape. The default is to use the full 300 GB of physical tape.
  - The 3592 Model J reads and writes EFMT1 (enterprise format 1) recording technology.

- The 3592 Model J uses the Enterprise Tape Cartridge (MEDIA5) physical media providing 300 GB of uncompressed capacity and 900 GB (assuming a 3:1 compression ratio) of compressed capacity depending on the type of data written.
- The 3592 Model J emulates 3590 Model B1A tape drives or 3490E tape drives.
- The 3592 Model J coexists with 3490E and 3590 devices in the 3494 automated tape library.

The 3592 Model J coexists with 3490, 3490E, and 3590 devices in a manual tape library. The 3592 Model J tape drive is always in emulation mode, either as a 3590-1 device or as a 3490E device. The 3592 Model J cannot use the MEDIA3 and MEDIA4 cartridges used in 3590 tape drives , nor the MEDIA1 and MEDIA2 cartridges used in 3490 tape drives.

Part 1. Introduction to Storage Management with DFSMShsm

This introductory part of the Storage Administration Guide describes the functions of the DFSMShsm program and highlights the following information:

- The tasks that you perform to make DFSMShsm do useful work
- An overview of how DFSMShsm moves data
- How to use this document. You are introduced to the example system, which is referred to throughout the rest of this document.

Organization of This Part

This part contains one chapter. This chapter gives a brief introduction to DFSMShsm and an overview of its functions.

© Copyright IBM Corp. 1984, 2004

Chapter 1. Welcome to DFSMShsm

DFSMShsm is a functional component of the DFSMS/MVS family, which provides facilities for managing your storage devices. DFSMShsm is a DASD storage management and productivity tool for managing low-activity and inactive data. 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 cooperates with the other products in the DFSMSdfp family to provide efficient and effective storage management.

DFSMSdfp provides a Storage Management Subsystem (SMS) that allows storage administrators to control the use of storage. The Storage Management Subsystem provides storage groups, storage classes, management classes, and data classes that control the allocation parameters and management attributes of data sets. DFSMShsm performs space management and availability management of each data set as directed by the management class attributes of that data set. In addition, the storage group controls the allocation of the data set when DFSMShsm returns the data set to level 0 (L0) storage.

Purpose of Space Management

Space management is the function of DFSMShsm that allows you to keep DASD space available for users in order to meet the service-level objectives for your system. The purpose of space management is to manage your DASD storage efficiently. To do this, space management automatically and periodically performs functions that (1) move low-activity data sets from user-accessible volumes to DFSMShsm volumes and (2) reduce the space occupied by data on both the user-accessible volumes and the DFSMShsm volumes. The DFSMShsm space management functions are:

- Automatic primary space management of DFSMShsm-managed volumes, which includes:
  - Deletion of temporary data sets
  - Deletion of expired data sets
  - Release of unused, over-allocated space
  - Migration to DFSMShsm-owned migration volumes
- Automatic secondary space management of DFSMShsm-owned volumes, which includes:
  - Migration level cleanup, including deletion of expired migrated data sets and some migration control data set (MCDS) records
  - Moving migration copies from migration level 1 (ML1) to migration level 2 (ML2) volumes
- Automatic interval migration, initiated when a DFSMShsm-managed volume exceeds a specified threshold
- Automatic recall of user data sets back to DASD volumes
- Space management by command
- Space-saving functions, which include:
  - Small data set packing
  - Partitioned data set compression
  - Data compaction
  - Data set reblocking

© Copyright IBM Corp. 1984, 2004

Purpose of Availability Management

Availability management ensures that a recent copy of your DASD data set exists. Availability management ensures that lost or damaged data sets can be retrieved at the most current possible level. To do this, availability management automatically and periodically performs functions that (1) copy all the data sets on DASD volumes to tape volumes and that (2) copy the changed data sets on DASD volumes either to other DASD volumes or to tape volumes. By copying the data sets, DFSMShsm minimizes the space that is occupied by the data set on the backup volume. Availability management comprises the following functions:

- Aggregate backup and recovery
- Automatic physical full-volume dump
- Automatic incremental backup
- Automatic control data set backup
- Command dump and backup
- Command recovery
- Disaster backup
- Expiration of backup versions
- Fast replication backup and recovery

What You Are Expected to Do

Making DFSMShsm perform useful work in your system requires four sets of tasks:

- Installing DFSMShsm and its required data sets in the system
- Specifying to DFSMShsm the automatic and command activities that it must perform
- Monitoring and tuning the operations of DFSMShsm
- Partitioning the DFSMShsm activities between systems (z/OS images) that share storage which must be managed. Establishing one or more DFSMShsm address spaces (hosts) for each of those z/OS images

The first of these sets of tasks is described in the *z/OS DFSMShsm Implementation and Customization Guide*. DFSMShsm provides a starter set of DFSMShsm commands used at each DFSMShsm startup in SYS1.SAMPLIB to help you get started using DFSMShsm.

This *Storage Administration Guide* describes the second, third, and fourth sets of tasks. That is, it describes how to use DFSMShsm to manage your storage, how to monitor and tune DFSMShsm operations, and how to partition DFSMShsm activities between all of the DFSMShsm hosts. Because this document is designed as a textbook, you can learn about DFSMShsm tasks by reading this document from beginning to end. However, because this document is organized by tasks, once you understand which tasks you want to perform, you can go either to specific parts or chapters to learn all of the necessary activities for that specific task.

Related reading:

- For additional information about the starter set, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about whether and how you need to partition DFSMShsm activities, see Chapter 17, "Operating Procedures for DFSMShsm," on page 539.

Tasks for Managing Storage with DFSMShsm

The following set of tasks for using DFSMShsm to manage storage has been defined:

- Setting overall DFSMShsm system parameters. This task consists of defining those parameters that DFSMShsm uses, no matter which of its functions it is to perform.
- Performing space management. This task consists of:
  - Defining the parameters that DFSMShsm uses for automatic space management
  - Issuing commands that cause DFSMShsm to perform specific space management functions
- Performing availability management. This task consists of:
  - Defining the parameters that DFSMShsm uses for automatic availability management
  - Issuing commands that cause DFSMShsm to perform specific availability management functions
- Maintaining the control data sets. This task consists of backing up and maintaining correct contents for the data sets that DFSMShsm uses to keep track of the data sets that it is managing.
- Managing DFSMShsm media. This task consists of eliminating invalid data sets from DFSMShsm volumes and in using DFSMShsm to move data sets when new kinds of volumes are introduced into your system.
- Determining what occurs in DFSMShsm. This task consists of using commands that cause DFSMShsm to issue reports of what has occurred under specific conditions and of reading and interpreting the reports.
- Running DFSMShsm in a simulation mode. This task consists of using DFSMShsm as though you were managing storage, but having DFSMShsm only issue a report instead of actually moving data sets. The purpose of the task is to analyze the results that would be obtained by using DFSMShsm commands in the way you planned to use them.
- Operating DFSMShsm. This task consists of directing the operations that start, stop, and interrupt DFSMShsm.

Overview of DFSMShsm Functions

Figure 1 on page 7 shows the generalized flow of data in DFSMShsm. The user's data is stored on level 0 volumes, either SMS-managed or non-SMS-managed.

Note: Only level 0 DASD volumes can be SMS managed. Migration DASD and backup DASD should never be managed by SMS. Migration, backup, and dump tapes can be SMS-managed in tape libraries.

Your objective in using DFSMShsm is to make the most efficient use of all your DASD storage, primarily by making better use of the level 0 volumes. Better use of the level 0 volumes results when:

- You specify thresholds for DFSMShsm-managed volumes, and you can assure the users that space will be available for extending old data sets or allocating new data sets (space management).
- You can assure the users that a copy of their data sets will be available if the data sets should accidentally be lost from the level 0 volumes (availability

management). Such assurance that you can provide backup copies may encourage users not to unnecessarily maintain their own backup copies on level 0 volumes.

DFSMShsm allows you to perform both availability management and space management automatically, periodically, and by issuing specific commands when manual operations are necessary or desirable. The dump and backup copies are made to tape (for dump) or to DASD or tape (for backup).

Rule: Very large data sets (in excess of 64K tracks, where K=1024) cannot be migrated or backed up to DASD; they must be migrated or backed up to tape.

DFSMShsm can then restore the volumes and recover the data sets when they are needed to replace destroyed or damaged data sets.

Note: Although DFSMShsm performs the same functions of copying, moving, and deleting data sets for both SMS-managed and non-SMS-managed volumes, the kinds and levels of control are different. Therefore, these DFSMShsm functions are described separately for SMS-managed volumes and for non-SMS-managed volumes.

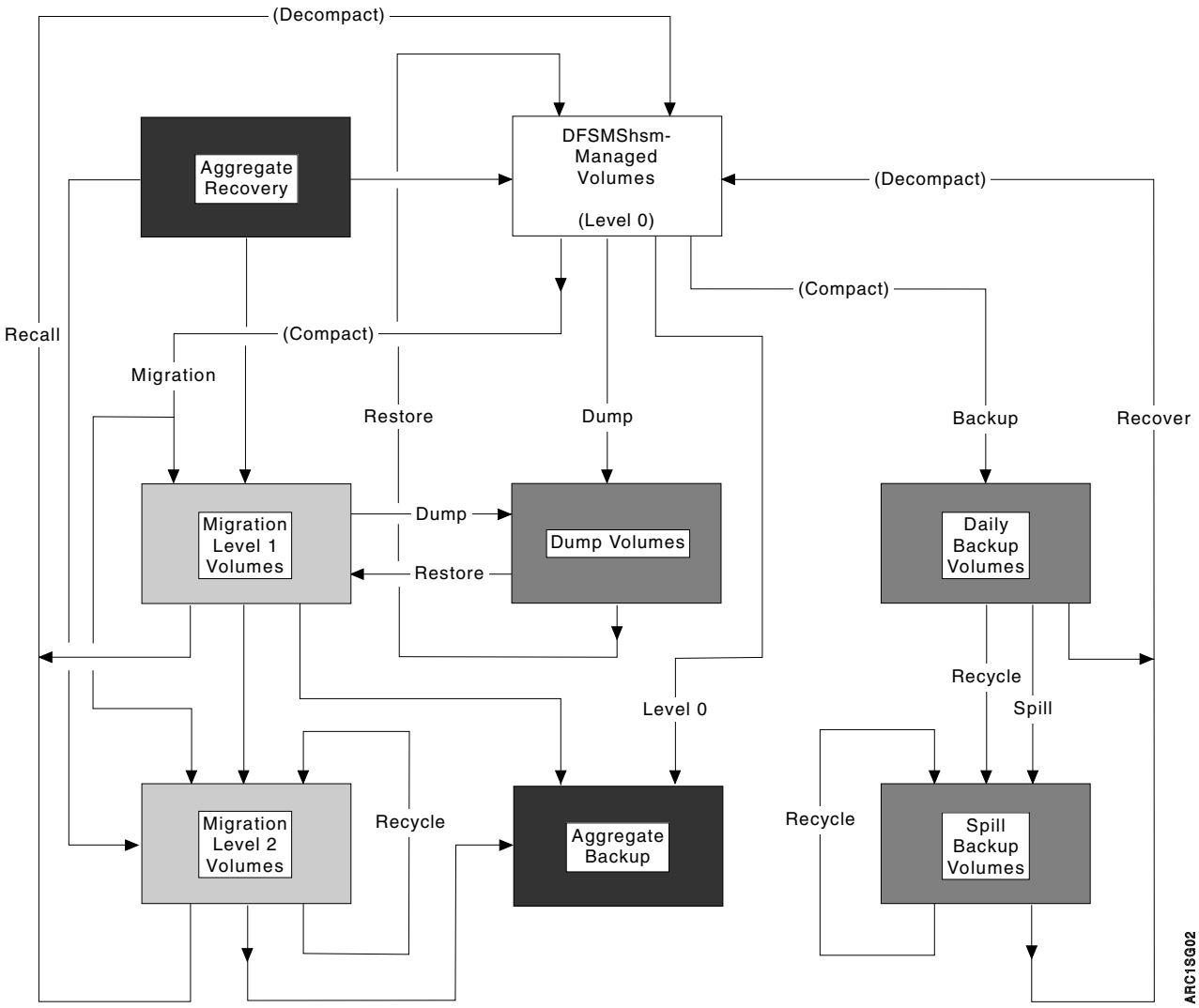


Figure 1. Data Set Flow in DFSMShsm

DFSMShsm Functions for SMS-Managed Volumes

For SMS-managed level 0 volumes, the attributes specified in the storage groups and the management classes govern how DFSMShsm performs automatic space management and incremental backup. You can override many of the attributes in the storage groups and management classes when the functions are performed by issuing specific commands. See Figure 1 when reading the following descriptions.

Note: If you have converted a non-SMS-managed L0 volume to an SMS-managed volume, you should:

- Add the volume to a storage group having the desired SMS attributes.
- DELVOL the volume to remove non-SMS attributes. Unexpected failures, for instance in autobackup processing, might occur if the volume is not DELVOLed.

Related reading: For additional information about conversion, see *z/OS DFSMS: Implementing System-Managed Storage*.

Use of Storage Groups and Management Classes

By managing storage with SMS, you establish the following:

- Storage classes to define levels of service provided to data sets
- Storage groups to govern the volumes to which data sets are allocated and subject to management by DFSMShsm
- Storage groups to define which volumes are used as target volumes of the fast replication backup versions
- Management classes to define how DFSMShsm manages the data sets
- Data classes to govern values for initially allocating the data sets
- Automatic class selection (ACS) routines that automatically select storage classes, management classes, data classes, and storage groups to govern the data sets

DFSMShsm is concerned mainly with DASD storage groups and with management classes. DASD storage groups allow you to pool volumes for the purpose of defining how and whether:

- Automatic dumps are performed on the volumes and to what dump classes
- Automatic backups are performed on the volumes
- Automatic volume space management is performed on the volumes and at what level of occupancy

Management classes allow you to group data sets logically and specify how they are managed by space management and availability management.

As users allocate data sets, the automatic class selection routine associates them with the appropriate storage class, storage group, management class, and data class (data and management classes are optional). If a data set is not SMS managed, SMS does not associate any class or storage group with the data set.

The storage group and the management class are related only by the needs of particular data sets. That is, SMS can allocate data sets that are associated with different management classes to the volumes assigned to a single storage group. Thus, in using DFSMShsm to manage SMS-managed storage, you must think in terms of managing data sets rather than in terms of managing volumes. (An exception to the preceding statement is the automatic dump function, which operates on a volume basis.)

Dump and Restore Functions

Because automatic dump is a volume function, the attributes that govern it are taken from the storage group. As Figure 1 on page 7 shows, DFSMShsm copies the data sets from level 0 volumes to dump volumes. Dump can be performed only on complete volumes.

Restore, the reverse of dump, moves dumped data sets from dump volumes to level 0 volumes. There are conditions under which a data set, in contrast to a volume, can be restored.

Related reading:

- For additional information about restoring a non-VSAM data set, see “Restoring a Non-VSAM Data Set from a Dump Copy” on page 231.
- For additional information about restoring a VSAM data set, see “Restoring a VSAM Data Set from a Dump Copy” on page 232.

Backup and Recovery Functions

The backup function is a data-set-level function when DFSMShsm is processing SMS-managed volumes. That is, the management class attributes define how the

data set is treated for creation and retention of backup versions. Thus, on the same volume you can have data sets with different:
- Permissions for automatic backup
- Conditions for command backup
- Minimum frequencies of backup
- Numbers of backup versions
- Retention periods for backup versions

For backup, DFSMShsm makes copies (versions) of changed data sets residing on level 0 volumes onto 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 have been 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.

Backup can occur to either DASD or tape, whichever you specify. If backup is performed to DASD, your DASD volumes can become filled with the current backup versions and earlier backup versions that have not been discarded. When DASD backup volumes become full, DFSMShsm transfers the old (non-current but valid) backup versions to spill backup volumes. 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 have been superseded by the more current versions that have been made. When tape volumes have many invalid versions, they are selected by the recycle process, which moves valid versions to spill backup volumes.

Recovery, the reverse of backup, returns 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 for complete volumes.

Aggregate Backup and Recovery Functions

The aggregate backup and aggregate recovery functions provide you with the capability to back up and recover a user-defined group of data sets. The user-defined group of data sets may be those belonging to an application or any combination of data sets that you want treated as a separate entity. With the aggregate backup and aggregate recovery functions, you can:
- Define the components of an aggregate
- Back up data sets by aggregate
- Recover data sets by aggregate
- Duplicate your aggregates at a remote site
- Resume business at a remote location if necessary

When the NetView® File Transfer Program is installed on any two nodes, it can be used to transfer copies of aggregate backup files by running a batch job or using NetView File Transfer Program input panels. The NetView File Transfer Program is referred to as NetView FTP in this book.

Related reading: For additional information about aggregate backup and recovery support (ABARS), see Chapter 8, “Aggregate Backup and Recovery Support (ABARS),” on page 281.

Fast Replication Function

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

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

Related reading: For additional information about the fast replication function, see "Managing Volume Backups with Fast Replication" on page 238.

Space Management Function

Space management consists of removing appropriate data sets from the level 0 volumes and reducing the space occupied on the level 0 volumes by the data sets that remain there. In removing the data sets, you can have DFSMShsm either delete them entirely or move them to other DASD or tape volumes. If you move the data sets to other DASD or tape volumes (ML1 or ML2), they can be automatically deleted from the migration volumes when they expire.

Rule: Very large data sets (in excess of 64K tracks, where K=1024) cannot be migrated to DASD; they must be migrated to migration level 2 tape.

The space management functions of DFSMShsm for SMS-managed volumes are performed on a data set level basis. That is, for each data set on a volume to be managed, DFSMShsm uses the management class attributes of the management class with which the data set is associated to determine:

- Whether to release unused overallocated space
- Whether to delete the data set based on:
  - The amount of time since the data set was created
  - The amount of time since the data set was last used
  - An explicit date
- The minimum time to leave the data set unopened on the level 0 volume before moving it
- Whether a data set on a level 0 volume is eligible to move to an ML1 volume or directly to an ML2 volume
- Whether or not to move the data sets on ML1 volumes to ML2 volumes
- For data sets on ML1 volumes, the minimum time to leave the data set unopened on both the level 0 volume and the ML1 volume before moving it to an ML2 volume
- A capability to migrate older generation data groups (GDGs)
- Whether or not to delete or migrate rolled-off generations if the data set was a member of a GDG

DFSMShsm uses the attributes from the storage group to determine which level 0 volumes to process for space management, how full the volume must be before data sets are migrated from the volume, and how much free space should be on the volume after space management has finished.

As space management processes each volume, it first tries to make space on the volume by deleting expired and temporary data sets and by releasing over-allocated space in data sets that cannot be deleted. During extent reduction, DFSMShsm releases any unused space in the data set, both internal and on the end, and places the data set on the same volume in as few extents as possible. If deleting data sets and releasing space does not make enough space available, DFSMShsm moves eligible data sets to a migration volume (usually an ML1 volume) as specified by the management class.

Rule: All ML1 volumes must be DASD. If ML1 volumes become sufficiently full, DFSMShsm moves migrated data sets from ML1 volumes to ML2 volumes as specified by the management class attributes for each data set.

ML2 volumes can be either DASD or tape. You can issue a DFSMShsm command to move data sets from DASD ML2 volumes to tape ML2 volumes. As data sets on tape ML2 volumes become invalid, you can recycle the valid data sets to other tape ML2 volumes.

Recall, the reverse of migration, moves data sets from ML1 or ML2 volumes to level 0 volumes. With the recall function, DFSMShsm returns data sets to storage groups as specified by SMS, whether the data set migrated from a volume in a storage group or not. This allows data sets that migrated before they were SMS-managed to be put under SMS management during recall.

DFSMS Sysplex Name Support for DFSMShsm Automatic Functions

Because DFSMShsm supports the use of DFSMS sysplex names for the MIGRATE SYSTEM/SYSPLEX NAME, BACKUP SYSTEM/SYSPLEX NAME, and DUMP SYSTEM/SYSPLEX NAME storage group system name attributes, it handles each storage group for automatic space management, data availability management, and automatic dump processing as follows. How the storage groups are handled is referred to as the "system affinity" or "host affinity."

- If the storage group system name is a system name, the storage group is processed by that specific system.
- If the storage group system name is blank, the storage group can be processed by any system.
- If the storage group system name is a sysplex name, the storage group is processed by a subset of systems.

Each automatic function (automatic space management, data availability management, or automatic dump processing) is processed independently. Therefore, the various systems used by one automatic function will not affect the other automatic functions.

DFSMShsm Functions for Non-SMS-Managed Volumes

For non-SMS-managed volumes, DFSMShsm uses volume parameters to determine how each data set on the volume is managed.

Dump and Restore Functions

The dump function for non-SMS-managed volumes is governed by parameters of the DFSMShsm commands, but it is performed the same way as for SMS-managed volumes. However, in addition to performing the automatic dump function for level 0 volumes, DFSMShsm also performs automatic dump for ML1 volumes.

Rule: Because the ML1 volumes are not managed by SMS—even in an environment in which all level 0 volumes are SMS-managed—you must use non-SMS-managed volume parameters to perform automatic dump for ML1 volumes.

A data set dumped from a non-SMS-managed volume can be restored to an SMS-managed volume if the data set does not already exist, if SMS is active, and if the data set is indicated by the ACS routines to be SMS-managed.

Backup and Recovery Functions

The volume-basis operation of DFSMShsm for non-SMS-managed volumes causes differences in how backup is performed. For non-SMS-managed volumes:

- If automatic backup is in effect for the volume, all changed data sets that DFSMShsm can manage are backed up.
- All data sets processed by a particular DFSMShsm host have the same backup frequency unless the frequency has been modified for that data set by a DFSMShsm command.
- All data sets processed by a particular DFSMShsm host have the same number of backup versions, unless the number of backup versions has been modified for that data set by a DFSMShsm command.

In addition, you can specify the number of days to retain backup versions of non-SMS-managed data sets only by command parameters for the entire system complex.

A data set backed up from a non-SMS-managed volume can be recovered to an SMS-managed volume if the data set does not already exist, if SMS is active, and if the data set is indicated by the ACS routine to be SMS-managed.

Space Management Function

Space management consists of removing appropriate data sets from the level 0 volumes and reducing the space occupied on the level 0 volumes by the data sets that remain there. In removing the data sets, you can have DFSMShsm either delete them entirely or move them to other DASD or tape volumes. If you move the data sets to other DASD or tape volumes (ML1 or ML2), they can be automatically deleted from the migration volumes when they expire.

Differences in the way that DFSMShsm performs space management for non-SMS-managed data sets include:

- The deletion age and management techniques are the same for all data sets on a volume managed by the deletion (delete by age) technique.
- The deletion age and management techniques are the same for all data sets on a volume managed by the retirement (delete if backed up) technique.
- The migration age and management techniques are the same for all data sets on a volume managed by the migration technique.
- The time for the data set to remain unopened before migration from ML1 to ML2 volumes is the same for all non-SMS-managed data sets in the system complex.

A data set 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.

Devices Supported by DFSMShsm

DFSMShsm supports devices other than those specifically recommended in the examples in this document. For example, ML2 volumes can be either tape or DASD. Daily backup and spill backup volumes can also be either tape or DASD.

DFSMShsm can be used on the following device types:

- All models of 3390, 9345, and 3380 DASD

- IBM 3494 and 3495 Tape Library Dataservers
- All models of 3590-1, 3490, 3480, 3480XF (with and without cartridge loaders), and 3592 (including Model J tape drives)
- Cache storage controls for DFSMShsm control data sets:
  - IBM 2105
  - IBM 3990
  - IBM 3880 Model 23
  - IBM 3880 Model 13 (only indexes)

Tip: For improved performance, the DFSMShsm control data sets should be on cache storage controllers.

How to Use The Example System

Figure 2 and Figure 3 on page 14 show two aspects of the example system used to explain DFSMShsm. The figures are separated into SMS-related and non-SMS-related examples to simplify the discussions. Figure 4 on page 17 shows the unified system. The following major features affect the way in which the DFSMShsm system commands are structured.

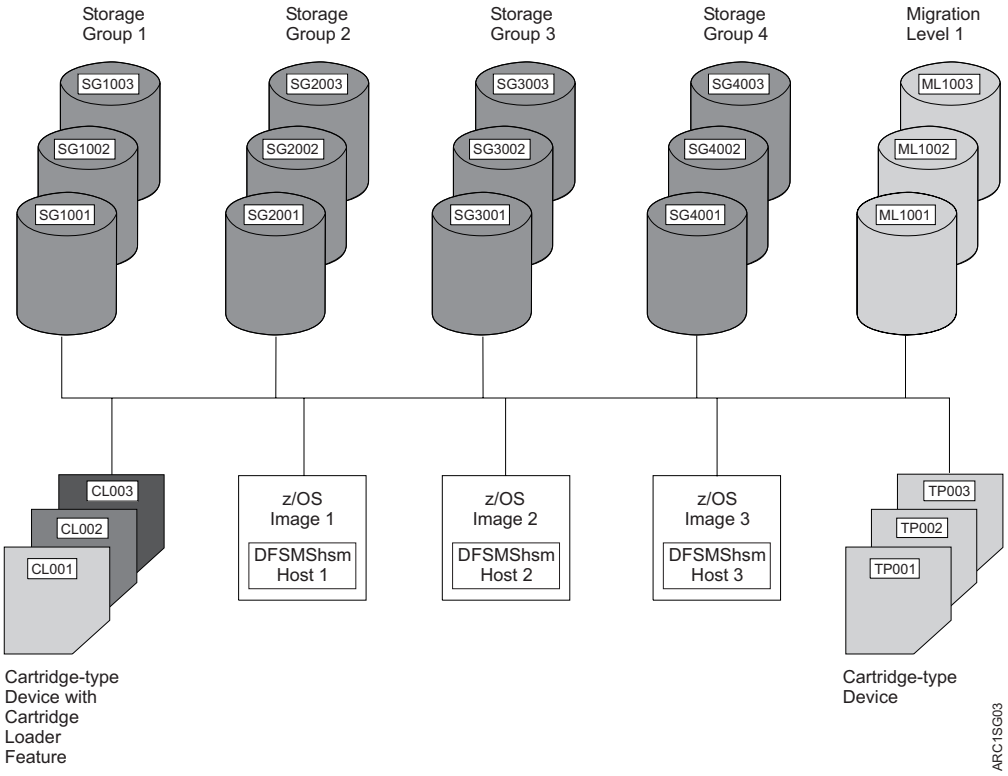


Figure 2. Example System for SMS-Managed Volumes

Note: Cartridge-type devices may be in a manual tape library (MTL) or in an automated tape library (ATL).

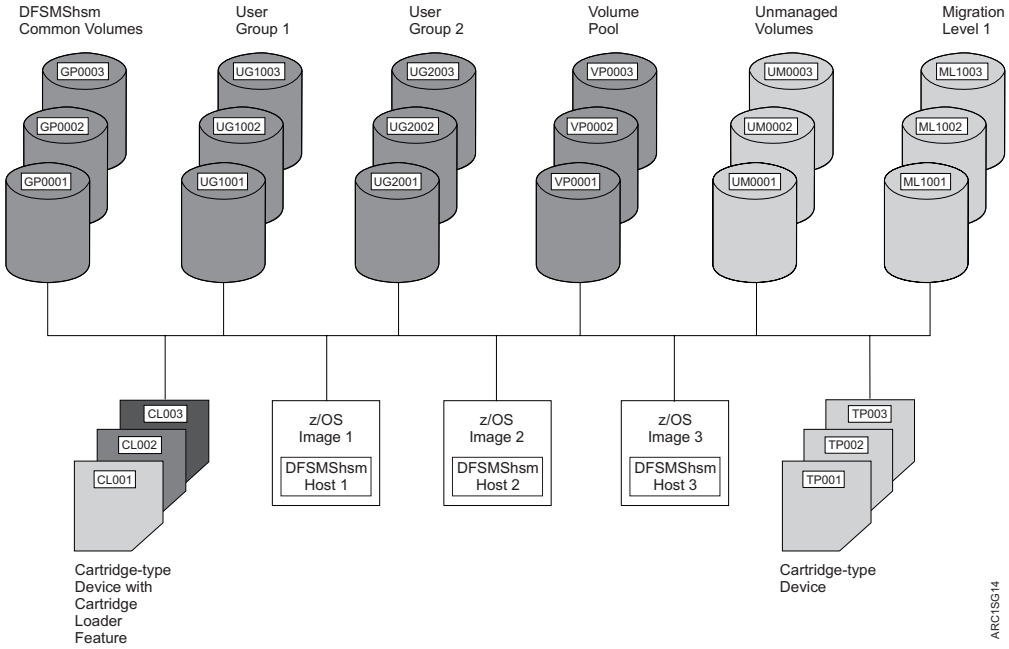


Figure 3. Example System for Non-SMS-Managed Volumes

Related reading: For additional information about how to direct DFSMShsm functions to a system-managed library, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Multiple z/OS Images in the Example System

Using multiple z/OS images allows the processing of a single DFSMShsm host on one z/OS image or multiple DFSMShsm hosts on more than one z/OS image. In the example system, there is one DFSMShsm host per z/OS image. DFSMShsm is shown running under JES2. However, when the use of a single z/OS image or of JES3 affects the selection of DFSMShsm commands or the results attained with those commands, the differences that result from using JES3 instead of JES2 are explained.

Note: The z/OS images, tapes, and ML1 volumes shown in Figure 2 on page 13 and Figure 3 are duplicated in both figures.

DASD Storage in the Example System

The example system shows only a limited number of DASD units in each classification. You can, of course, have any number of units in any of the classifications (up to the maximum channel and addressing capabilities of your system). All of the DASD is shared among all of the z/OS images. Where other sharing arrangements cause differences in the use of commands or in the results obtained, the differences are explained.

Storage Groups 1 through 4

The example system uses DFSMSdfp to manage its storage. Figure 2 shows those elements of the system that are essential for using the SMS capability of DFSMSdfp. Storage groups 1 through 4 are the storage that is actually under control of the SMS sysplex name support at this time. The data sets on these volumes are the ones referred to as SMS-managed. Expectations are that some of the DASD volumes shown in both this figure and in Figure 3 will be brought

under SMS control as time goes on. This expectation applies only to the level 0 volumes, not to the DFSMShsm-owned volumes.

User Groups 1 and 2

User groups 1 and 2 (Figure 3) are non-SMS-managed level 0 volumes that DFSMShsm manages and that are reserved for particular groups of data sets. Data sets with particular high-level qualifiers in the data set names are directed to these volumes. Groups of volumes to which data sets are directed by their high-level qualifiers are known as *data set pools*.

Volume Pool

The volume pool (Figure 3) consists of non-SMS-managed volumes that are differentiated on some basis other than that of data set names. Data sets that are assigned to these volumes return to these volumes if they are migrated and recalled.

DFSMShsm Common Volumes

The DFSMShsm common volumes (Figure 3) are the DFSMShsm-managed volumes that contain non-SMS-managed data sets that do not belong on the volumes in either the user groups or the volume pools.

Unmanaged Volumes

The unmanaged volumes (Figure 3) are level 0 volumes that are not managed by either SMS or DFSMShsm. DFSMShsm operations can be performed on these volumes, but only by issuing commands. Automatic DFSMShsm operations are not performed on these volumes.

Migration Level 1

The ML1 volumes (Figure 2 and Figure 3) are DFSMShsm-owned volumes on which migrated data sets are stored. ML1 volumes are always DASD. As with the z/OS images, the ML1 volumes shown in both figures are the same volumes.

Tape Devices in the Example System

In the example system, there are two groups of tape devices, again duplicated in both figures. These devices are used for dump volumes, ML2 volumes, and backup volumes. ML2 and backup volumes can also use DASD.

Figures 2, 3, and 4 show two groups of cartridge-type devices: those with cartridge loaders, and those without. The cartridge-type devices without cartridge loaders are used (in the example) mainly as input devices for recall and recovery because recall or recovery is always from a specific tape. The cartridge-type devices with cartridge loaders are used for output to backup, dump, and ML2 tapes because output can use scratch tapes (tapes that no longer contain valid data, and have been returned to the global scratch pool) and take advantage of the cartridge loader's ability to change tapes automatically.

Note: Cartridge-type devices may be in an MTL or an ATL.

Related reading: For additional information about manual tape libraries (MTLs) and automated tape libraries (ATLs), see the *z/OS DFSMShsm Implementation and Customization Guide*.

Other Assumptions in the Example System

In order to understand how to use the example system, other assumptions must be made concerning your knowledge and concerning security systems for the example system.

Your Knowledge

There is an assumption that you know about MVS/ESA™, DFSMSdfp, TSO, and RACF, a component of the Security Server for z/OS.

Under this assumption, if you are told to perform an action related to these products, you know how to perform that action without a discussion being necessary.

Security Systems

So that you can be shown what DFSMShsm does in relation to all types of data set protection, it is assumed that the example system provides the following:

- RACF or equivalent security system
- Expiration date protection for tapes

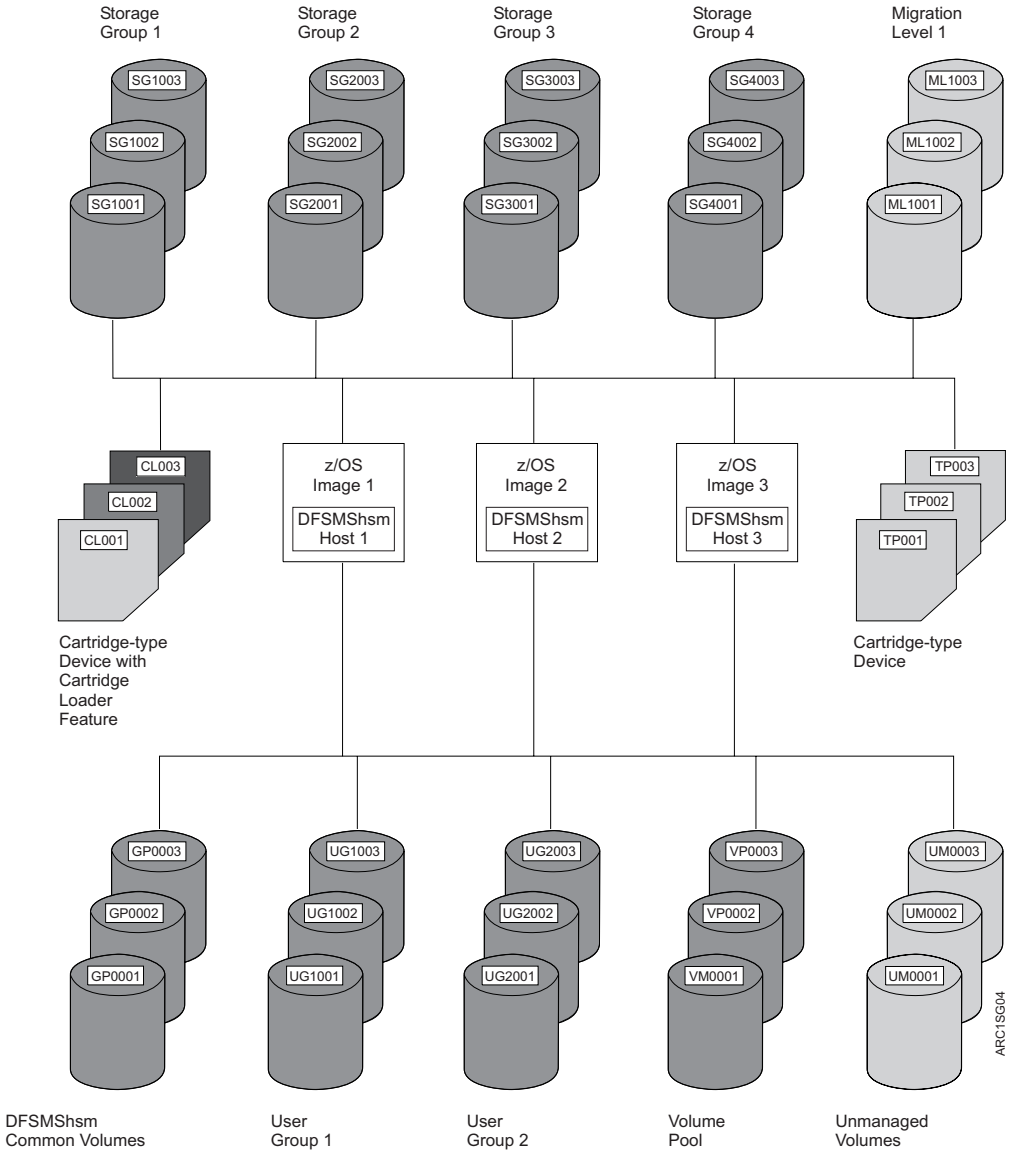


Figure 4. Comprehensive Example System

Now you can use the example system to see how to manage low-activity and inactive data sets by using DFSMShsm.

Part 2. How to Control Space Management

Space management is the DFSMShsm program function that you use to ensure that your customers have space available on DASD volumes to allocate new data sets or to extend old ones. You can make the space available by:

- Deleting data sets that have outlived their usefulness
- Removing unused allocated space from data sets
- Moving low-activity data sets from level 0 volumes to other DASD or tape volumes
- Returning the moved data sets to the level 0 devices when the data sets are needed

Functions of Space Management

Space management makes it possible for you to:

- Automatically delete, remove unused space from, and move data sets on DFSMShsm-managed volumes to provide a specified amount of free space on each volume. The space management functions begin at a specified time of day and only on a specific day of a cycle. This operation is called *automatic primary space management*. The process of removing unused space from the data sets is called *space reduction*. Moving the data sets (as opposed to deleting or removing unused space from them) is called *migration*.
- Reconnect unchanged data sets that are recalled from ML2 tapes to their original ML2 tapes with the fast subsequent migration function.
- Ensure, on an hourly basis, that a specified amount of space is available on DFSMShsm-managed volumes. If DFSMShsm finds any volume without the specified amount of space, it schedules migration from all such volumes (known as *interval migration*) following the current or a subsequent hourly space check.
- Automatically clean up migration volumes and the migration control data set (MCDS). This function and the following one are known as *automatic secondary space management*.
- Automatically, or by command, migrate data sets from level 1 migration volumes to level 2 migration volumes.
- Automatically recall needed migrated data sets to level 0 volumes.
- Recall migrated data sets to level 0 volumes by command.
- Delete eligible data sets on a non-SMS-managed volume by command.
- Delete eligible non-SMS migrated data sets on a migration volume by command.
- Migrate individual data sets or eligible data sets on a volume by command.

Because DFSMShsm space management for SMS-managed volumes is controlled at a different level than non-SMS-managed volumes, the descriptions of managing the two different groups of volumes are separated.

© Copyright IBM Corp. 1984, 2004

Organization of This Part

This part is divided into the following chapters.

Task	See Page
Chapter 2, “Space Management of SMS-Managed Storage”	21
Chapter 3, “Space Management of Non-SMS-Managed Storage”	79
Chapter 4, “Other Space Management Considerations”	109
Chapter 5, “Space Management Procedures”	141

Chapter 2. Space Management of SMS-Managed Storage

DFSMShsm manages SMS-managed storage by data sets. That is, the specification for how to treat each data set on a volume is contained in the management class associated with that data set. The storage management subsystem determines to which volume the data set is recalled.

Related reading: To view the consolidation of all of the DFSMShsm commands, storage group attributes, and management class attributes that show the automatic storage management of the example system, see Appendix A, "Consolidated Commands, Storage Group Definitions, and Management Classes for the Example System," on page 611.

Space Management Functions

This section describes the subtasks necessary to do space management for SMS-managed storage.

Automatic Primary Space Management

During automatic primary space management, all DFSMShsm hosts delete temporary data sets and expired data sets from the DFSMShsm-managed volumes they are processing. This is done under the control of the management class associated with each data set on the volume or the expiration date contained in the data set's volume table of contents (VTOC) entry. Under control of the data set management classes, all DFSMShsm hosts release unused allocated space in physical sequential, partitioned, and extended format virtual storage access method (VSAM) data sets. During data set and volume processing, the fast subsequent migration reconnects eligible data sets to the ML2 tape from which they were most recently recalled. Under the extent reduction function, all DFSMShsm hosts also reduce the number of extents of physical sequential, partitioned, and direct access data sets that have exceeded a 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. This continues until the SMS-managed volumes have the specified amount of free space. However, if deletion of expired data sets, fast subsequent migration, and space reduction of the remaining data sets achieves the specified free space, no data sets are moved. Each data set migrates in compacted form. Data sets that are expected to be smaller than 110 KB (where KB equals 1024 bytes) after they are compacted migrate to the small data set packing (SDSP) data sets on ML1001 and ML1002.

Using the fast subsequent migration function, certain data sets that are recalled from ML2 tapes can be reconnected to the ML2 tapes from which they were most recently recalled. Data sets that are eligible for reconnection undergo fast subsequent migration.

Automatic Secondary Space Management

Automatic secondary space management schedules TAPECOPY commands for migration tape copy needed (TCN) records, deletes expired data sets from the *migration volumes*, deletes obsolete MCDs, VSRs, and DSRs during migration cleanup, and moves data sets (under control of the management class) from ML1 to ML2 volumes.

© Copyright IBM Corp. 1984, 2004

Automatic Interval Migration

Each hour throughout the day, DFSMShsm host 2 performs interval migration, as needed, for all storage groups. 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 DFSMShsm-managed volume reaches the low threshold or no more data sets are eligible.

During interval migration processing, DFSMShsm performs the following functions:

- Issues a message for unmatched data sets
- Deletes temporary data sets
- Deletes expired data sets
- Excludes data sets that are candidates to be backed up from migration to tape
- Migrates data sets that are candidates for migration

Related reading: For additional information about managing storage and space requirements, see "Specifying the Storage to Be Managed and the Space Requirements."

Automatic Recall

If a data set that has migrated is needed for processing, DFSMShsm recalls the data set from the migration volume on which it resides to an SMS-managed volume. The storage management subsystem selects the SMS-managed volume to which the data set is returned. Migrated extended format VSAM data sets can be recalled only as extended format VSAM data sets. Migrated extended format sequential access method (SAM) data sets can be recalled as extended format or nonextended format data sets.

Command Space Management

DFSMShsm provides you with the ability, by issuing individual commands, to perform space management functions.

Tasks for Automatic Space Management

The tasks for controlling automatic space management of SMS-managed storage require that you add DFSMShsm commands to the ARCCMD*xx* member and that you specify attributes in the SMS storage groups and management classes. It is assumed that you have already defined the storage groups and management classes. The descriptions within this document are of attributes that you need to add to the previously defined storage groups and management classes. Similarly, the descriptions of DFSMShsm commands relate to commands to be added to the ARCCMD*xx* member of SYS1.PARMLIB.

Specifying the Storage to Be Managed and the Space Requirements

DFSMShsm must be told what storage to manage and how much space to make free on the volumes. This specification comes from the storage group.

You can use the AUTO MIGRATE attribute to specify whether DFSMShsm performs automatic volume space management on the volumes that are assigned to the storage group. The AUTO MIGRATE attribute specifies the type of space management that DFSMShsm uses on the volumes that are assigned to the storage group. Table 1 shows the acceptable values of the AUTO MIGRATE attribute and the processing that is performed based on those values.

Table 1. AUTO MIGRATE Attribute Values

AUTO MIGRATE Value	Automatic Primary Space Management Performed?	Automatic Interval Migration Performed?
Y	Yes	Yes (note 1)
I	Yes	Yes
P	Yes	No
N	No	No (note 2)

Notes:

1. Depends on the value of SETSYS INTERVALMIGRATION | NOINTERVALMIGRATION.
2. Not performed, even if you have specified SETSYS INTERVALMIGRATION.

The AUTOMIGRATE=Y value indicates that you want the storage group managed by primary space management, and, on occasion, by interval migration. You can control when interval migration is allowed using the SETSYS INTERVALMIGRATION or NOINTERVALMIGRATION parameters.

Example: You might want interval migration performed from 0900 to 1700 during the day, but not in the evening hours when batch processing is heaviest. During primary space management, volumes above the LOW threshold are processed. During the day, it might be possible that a few volumes exceed their HIGH threshold, and in this case, interval migration is performed.

The AUTOMIGRATE=I value indicates that you want the storage group managed by interval migration. Primary space management can also be used, but it plays a minor role overall. This value is not affected by the current setting of the SETSYS INTERVALMIGRATION or NOINTERVALMIGRATION parameters that are specified for storage groups that are specified with AUTOMIGRATE=Y. This value also has a different trigger for interval migration. When a volume exceeds the mid-point between HIGH and LOW thresholds, interval migration is performed.

Example: If HIGH=70 percent and LOW=20 percent, then volumes that exceed 45 percent are processed.

This value is ideal for use with Tape Mount Management (TMM), which redirects tape data sets to this storage group, and then lets DFSMShsm move them to tape hourly, as needed.

The AUTOMIGRATE=P value indicates that you want the storage group managed only by primary space management. This is useful when interval migration is considered too much of a drain on resources or is disruptive in other ways. This value ignores the SETSYS INTERVALMIGRATION setting that applies to the storage groups that are specified with AUTOMIGRATE=Y, and it avoids the hourly space checks that are performed for each volume.

The AUTOMIGRATE=N value indicates that you want no automatic space management to be performed. Data sets and volumes can be migrated using commands instead. You can use this method for volumes that are managed by other products, or when you do not want space management, such as for application database or system volumes that must remain DASD-resident for other reasons.

Example: In Figure 5, primary space management manages storage groups SG1 and SG2 (although interval migration is also done on these groups). Interval management manages storage group SG3. Storage group SG4 is not managed in either manner, but has thresholds to control new allocations.

```
NAME ===> SG1
        MIGRATE SYSTEM/SYSPLEX NAME ===>
        AUTO MIGRATE ===> Y
        ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80

        NAME ===> SG2
        MIGRATE SYSTEM/SYSPLEX NAME ===>
        AUTO MIGRATE ===> Y
        ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80

        NAME ===> SG3
        MIGRATE SYSTEM/SYSPLEX NAME ===>
        AUTO MIGRATE ===> I
        ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 20  LOW ===> 0

        NAME ===> SG4
        MIGRATE SYSTEM/SYSPLEX NAME ===>
        AUTO MIGRATE ===> N
        ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80
```

Figure 5. Storage Group Attributes for Space Management

The ALLOCATION/MIGRATE THRESHOLD attribute determines two controls for DFSMShsm:

- How much free space is available on the volumes when space management is finished (LOW)
- How full the volumes must be to cause interval migration to be performed on them (HIGH)

"Pass 1 of Automatic Primary space Management" on page 50
(non-data-movement function in primary space management) and extent reduction are performed for all volumes in the storage group (AUTO MIGRATE = Y or I). However, "Pass 2 of Automatic Primary Space Management" on page 55
(migration) is performed only for those volumes that are above the LOW threshold after pass 1 and extent reduction have completed. The HIGH threshold only determines eligibility for interval migration, and is not used for primary space management. SMS uses the HIGH threshold attribute to balance space allocations. DFSMShsm uses the HIGH threshold attribute to determine the need for interval migration. The LOW threshold attribute specifies the maximum level of occupied space after automatic volume space management processing has finished.

In Figure 5, storage groups SG1, SG2, and SG4 have the following threshold attributes:

```
HIGH===> 95 LOW===> 80
```

Storage group SG3 has the following threshold attributes:

```
HIGH===> 20 LOW===> 0
```

Note: A low threshold of 0 is permitted only with AM = I.

Related reading:

- For additional information about storage groups and their functions in SMS, see the *z/OS DFSMSdfp Storage Administration Reference*.
- For additional information about primary space management, see "Automatic Primary Space Management" on page 50.

Specifying the DFSMShsm Hosts That Process Each Storage Group

When DFSMShsm processes an SMS-managed volume for space management, it records the time at which the volume was processed. If another DFSMShsm host attempts to process the same volume for space management, it checks the time at which the volume was last processed. If the volume was processed within the last 14 hours, DFSMShsm does not process the volume again.

Because of the time-recording process for SMS-managed volumes, DFSMShsm does not require you to control which DFSMShsm host processes which volumes, and you should not do such manual load balancing. However, should you find it necessary to limit processing of certain volumes to one z/OS image, you can do so with the MIGRATE SYSTEM/SYSPLEX NAME attribute in the storage group.

Example: If only z/OS image 1 were to process storage group SG1 for space management, you could specify:

```
Storage group 1 (SG1)
  MIGRATE SYSTEM/SYSPLEX NAME ===> PRC1
```

The system name is the system-wide identifier for the z/OS image, not the identifier assigned with the HOSTID parameter of the DFSMShsm startup procedure.

Specifying How Groups of Data Sets Are Managed

During space management, DFSMShsm processes individual SMS-managed data sets to comply with the attributes specified in the management classes with which the data sets are associated. Management class attribute groups let you specify:

- Expiration attributes
  - EXPIRE AFTER DAYS NON-USAGE
  - EXPIRE AFTER DATE/DAYS
  - RETENTION LIMIT
- Partial release attribute
- Migration attributes
  - COMMAND OR AUTO MIGRATE
  - PRIMARY DAYS NON-USAGE
  - LEVEL 1 DAYS NON-USAGE
- GDG management attributes
  - # GDG ELEMENTS ON PRIMARY
  - ROLLED-OFF GDS ACTION

Specifying Expiration Attributes

The expiration attributes, EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS, govern how long a data set associated with the management class is kept if no expiration date is listed in the data set VTOC entry. (The expiration date

is in the data set VTOC entry if it was explicitly specified at the time the data set was created or if the data class specified an expiration date.)

EXPIRE AFTER DAYS NON-USAGE specifies how many days must elapse from the last access of the data set until the data set becomes eligible for expiration. EXPIRE AFTER DATE/DAYS specifies an absolute date or period after the data set has been created for a data set to become eligible for expiration. The default for both attributes is NOLIMIT.

Both attributes can be used together. When both attributes have a value other than NOLIMIT, both attributes must be satisfied before the data set is expired. When both attributes are NOLIMIT, the data set does not expire. When one of the attributes is NOLIMIT, that attribute is not used to expire the data set; the expiration of the data set is based on the attribute that is not NOLIMIT.

It is possible to have a management class request both automatic backup processing—by using the "AUTO BACKUP" setting of Y— and expiration processing—by using the "EXPIRE AFTER...." as numeric values. In this case, DFSMShsm determines if a backup copy has been made before expiring the data set. The data set will be deleted when any of the following conditions are true:

- The data set change flag is off in the VTOC entry *and* the last backup date in the catalog is nonzero.

  Note: DFSMShsm does not verify the existence of the backup version.
- The data set change flag is off in the VTOC entry *and* the last backup date in the catalog is zero *and* the management class does not request automatic backup for this data set.
- The data set change flag is on in the VTOC entry *and* the management class does not request automatic backup for this data set.

Rule: "ADMIN OR USER COMMAND BACKUP" must not be "NONE" and "AUTO BACKUP" must be Y to request automatic backup.

Note: This restriction does not affect expiration date processing as specified by SETSYS EXPIREDDATASETS(SCRATCH). When a data set with a specific expiration date expires, it is scratched regardless of any management class attributes when this SETSYS option is chosen.

Specifying Partial Release

The PARTIAL RELEASE attribute specifies whether DFSMShsm space management processing releases unused allocated space in physical sequential, partitioned, and extended format VSAM data sets. The default for PARTIAL RELEASE is NO. The other options are YES and CONDITIONAL. CONDITIONAL means to release the space only if there is a nonzero secondary allocation quantity.

Notes:

1. The CONDITIONAL option applies only to physical sequential and partitioned data sets. If specified for an extended format VSAM data set, it will be processed as if YES were specified.
2. The PARTIAL RELEASE attribute does not apply to normal migration and recall processing.

Related reading: For additional information about migration and recall space allocation, see "Additional Space Management during Migration and Recall When DFSMShsm Is the Data Mover" on page 70.

Specifying Migration Attributes

The migration attributes allow you to specify:

- Whether migration is permitted for the data sets
- When a data set is eligible to migrate from level 0
- When a data set is eligible to migrate from level 1
- To what level a data set is eligible to migrate

The COMMAND OR AUTO MIGRATE attribute specifies whether to migrate the data sets automatically, by command, or not at all. If you specify BOTH, both automatic and command migration can occur. If you specify COMMAND, the data sets cannot migrate automatically, but a migrate command for individual data sets can cause the data sets to migrate, regardless of the value specified on the PRIMARY DAYS NON-USAGE attribute. If you specify NONE, the data sets cannot migrate. The default is BOTH.

The PRIMARY DAYS NON-USAGE attribute specifies the minimum number of days a data set must remain unused on an SMS-managed volume before it becomes eligible to migrate. Eligibility to migrate does not imply that the data set will migrate the next time that space management occurs. If space management achieves the necessary free space without migrating all the eligible data sets on the volume, space management processing stops when the necessary free space is available. The default for this attribute is two days.

The LEVEL 1 DAYS NON-USAGE attribute serves two purposes:

- It specifies the minimum number of days that a data set must be unused before it becomes eligible for migration from level 1 to level 2. This number is the total days not used; it is not days that the data set has actually resided on an ML1 volume. All data sets that are eligible migrate at each level 1 to level 2 migration.
- It indicates what level to migrate to from either level 0 or level 1. A value of 1 to 9999 for this attribute specifies that the data set may migrate first from level 0 to level 1, then from level 1 to the specified ML2 device, either DASD or tape. A value of 0 for this attribute specifies that the data set migrates directly from level 0 to ML2 tape. (If the data set has already migrated to ML1 before the attribute was set to 0, it migrates to the specified ML2 device, either DASD or tape.) A value of NOLIMIT specifies that the data set migrates to ML1 and is not to migrate to ML2.

If the value of LEVEL 1 DAYS NON-USAGE has been met and the data set still resides on level 0, it will migrate directly to ML2.

The default for this attribute is 60.

Specifying GDG Management Attributes

The generation data group (GDG) management attributes let you specify migration eligibility for generation data sets based on relative generation number and what to do with rolled-off generation data sets (GDSs).

In establishing the order for migrating data sets from a DFSMShsm-managed volume, DFSMShsm can give high priority to generation data sets (members of generation data groups). That is, DFSMShsm can choose to migrate generation data sets before it migrates non-generation data sets. Generation data sets that are chosen in preference to non-generation data sets are said to receive early migration.

You may want to give high priority to generation data sets if old generations are used infrequently. To specify that old generations have high priority for migration and to specify how many generations have standard priority, you can specify the # GDG ELEMENTS ON PRIMARY attribute. When you specify this attribute, all non-rolled-off generations of each GDG that exceed the number you specify are given priority for early migration. If you specify 0 for this attribute, all generations of the GDGs are given priority for early migration.

The ROLLED-OFF GDS ACTION attribute specifies what to do with rolled-off GDSs. The values for this attribute are MIGRATE and EXPIRE. If you specify MIGRATE, DFSMShsm assigns the highest migration priority to rolled-off GDSs. If you specify EXPIRE, DFSMShsm deletes any rolled-off GDS that has met the expiration date specified in its data set VTOC entry or that has no expiration date specified in the data set VTOC entry. If a NOSCRATCH GDS rolls off while on a level 0 volume and is migrated, the GDS is expired during migration cleanup.

The GDG management attributes have no defaults. If you do not specify values for these attributes, the other migration and expiration attributes in the management class govern migration and expiration of generation data sets. If a GDS does not meet any of the criteria specified by the GDG management attributes, the other migration and expiration attributes in the management class govern migration and expiration of the data set.

Using the Default Management Class

SMS allows you to define a management class and to specify that this management class is the default management class. Thus, any SMS data set that is not assigned to a management class by the automatic class selection routine uses the attributes specified in the default management class.

If you do not specify a default management class, DFSMShsm generates its own default by using the default values for all of the management class attributes. Thus, in the absence of a specified default management class, DFSMShsm uses the following values for any SMS-managed data set for which a management class is not assigned.

Management Class Attribute	Default Value
EXPIRE AFTER DAYS NON-USAGE	NOLIMIT
EXPIRE AFTER DATE/DAYS	NOLIMIT
PARTIAL RELEASE	N
PRIMARY DAYS NON-USAGE	2
LEVEL 1 DAYS NON-USAGE	60
# GDG ELEMENTS ON PRIMARY	no default
ROLLED-OFF GDS ACTION	no default
BACKUP FREQUENCY	1
NUMBER OF BACKUP VERS (Data Set Exists)	2
NUMBER OF BACKUP VERS (Data Set Deleted)	1
RETAIN DAYS ONLY BACKUP VER (Data Set Deleted)	60
RETAIN DAYS EXTRA BACKUP VERS	30
ADMIN OR USER COMMAND BACKUP	BOTH
AUTO BACKUP	Y

Management Class Attribute	Default Value
BACKUP COPY TECHNIQUE	S

Note: DFSMShsm creates a management class called HSMDEFMC.

Management Class Examples

Assume that you want to use the following management classes: standard, GDS, large, and no migration.

Standard Management Class Attributes

For data sets using the standard management class (STANDMC), the following actions occur:

- The data sets expire if they are not used in 120 days and 360 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 10 days migrate to ML1.
- Data sets that are not used for 60 days migrate to ML2.

These are the management class attributes for STANDMC:

```
NAME ===> STANDMC
               EXPIRE AFTER DAYS NON-USAGE ===> 120
               EXPIRE AFTER DATE/DAYS      ===> 360
               PARTIAL RELEASE             ===> Y
               PRIMARY DAYS NON-USAGE      ===> 10
               LEVEL 1 DAYS NON-USAGE      ===> 60
               COMMAND OR AUTO MIGRATE     ===> BOTH
```

GDS Management Class Attributes

Data sets in generation data groups use the GDS management class (GDSMC) for the following results:

- Data sets expire if they are not used in seven days and 90 days have passed since they were created.
- Unused allocated space is not released from any data sets.
- Data sets that are not used in five days migrate to ML1.
- Data sets never migrate to migration level 2.
- Non-rolled-off data sets that exceed the last two generations receive priority for command and automatic migration.
- Rolled-off generation data sets are expired.

These are the management class attributes for GDSMC:

```
NAME ===> GDSMC
             EXPIRE AFTER DAYS NON-USAGE ===> 7
             EXPIRE AFTER DATE/DAYS      ===> 90
             PARTIAL RELEASE             ===> N
             PRIMARY DAYS NON-USAGE      ===> 5
             LEVEL 1 DAYS NON-USAGE      ===> NOLIMIT
             COMMAND OR AUTO MIGRATE     ===> BOTH
             # GDG ELEMENTS ON PRIMARY   ===> 2
             ROLLED-OFF GDS ACTION       ===> EXPIRE
```

Large Management Class Attributes

In the example system, large data sets are not to migrate to ML1 DASD. Instead, those data sets should migrate directly to ML2 tape. Data sets that use this management class (LARGEMC) receive the following processing:

- The data sets expire if they are not used in 90 days, and 180 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 15 days migrate to ML2 tape.

These are the management class attributes for LARGEMC:

```
NAME ===> LARGEMC
             EXPIRE AFTER DAYS NON-USAGE ===> 90
             EXPIRE AFTER DATE/DAYS      ===> 180
             PARTIAL RELEASE             ===> Y
             PRIMARY DAYS NON-USAGE      ===> 15
             LEVEL 1 DAYS NON-USAGE      ===> 0
             COMMAND OR AUTO MIGRATE     ===> BOTH
```

No-Migration Management Class Attributes

For data sets that have value as long as they are used but lose value quickly if they are not used, the no-migration (NOMIGMC) management class has been defined. Data sets that use this management class receive the following processing:

- The data sets never migrate.
- The data sets expire if they are not used in 20 days.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.

These are the attributes for NOMIGMC:

```
NAME ===> NOMIGMC
             EXPIRE AFTER DAYS NON-USAGE ===> 20
             EXPIRE AFTER DATE/DAYS      ===> NOLIMIT
             PARTIAL RELEASE             ===> Y
             PRIMARY DAYS NON-USAGE      ===>
             LEVEL 1 DAYS NON-USAGE      ===> NOLIMIT
             COMMAND OR AUTO MIGRATE     ===> NONE
```

The EXPIRE AFTER DAYS NON-USAGE attribute has been set to 20 days and the EXPIRE AFTER DATE/DAYS attribute has been set to NOLIMIT. The data sets will expire after 20 days of nonusage. If both attributes are set to NOLIMIT, the data sets will never expire.

Specifying When Automatic Primary Space Management Can Process

The PRIMARYSPMGMTCYCLE parameter of the DEFINE command defines the cycle of automatic primary space management.

Example: To define an automatic primary space management cycle, use the following command:

```
DEFINE PRIMARYSPMGMTCYCLE(YNNNYNN) -
CYCLESTARTDATE(1998/09/12)
```

Note: Although this example shows automatic primary space management running two days a week (Saturday and Wednesday). You might want to run it more often.

Specifying When Automatic Primary Space Management Starts

The PRIMARYSPMGMTSTART parameter of the SETSYS command controls the processing window of automatic primary space management.

Example: To define a processing window of 2:00 a.m. to 4:00 a.m., use the following command:

```
SETSYS PRIMARYSPMGMTSTART(0200 0400)
```

Note: Backup of the control data sets can interfere with starting automatic primary space management.

Related reading: For additional information about backing up control data sets, see "Phase 1: Backing Up the Control Data Sets" on page 183.

Specifying When Automatic Secondary Space Management Can Process

The SECONDARYSPMGMTCYCLE parameter of the DEFINE command defines the cycle of automatic secondary space management.

Example: To define the days for secondary space management to run and the start date, use the following command:

```
DEFINE SECONDARYSPMGMTCYCLE(YNNNYNN) -
CYCLESTARTDATE(1998/09/12)
```

Specifying When Automatic Secondary Space Management Starts

The SECONDARYSPMGMTSTART parameter of the SETSYS command controls the processing window of automatic secondary space management.

You can run automatic secondary space management any time of the day except when automatic primary space management is running (if SDSPs are used).

Note: You must specify SECONDARYSPMGMTSTART if you want it to run; it will not run by default.

Example: To define a processing window of 12:30 a.m. to 2:00 a.m., use the following command:

```
SETSYS SECONDARYSPMGMTSTART(0030 0200)
```

Although secondary space management ending time is 2:00 a.m., any data set deletion management work elements (MWEs) that secondary space management built up to 2:00 a.m. would still continue to be processed. If a large number of data sets needed deletion, MWE processing could continue for a significant time after secondary space management stops building data set deletion MWEs.

Notes:

1. Data set deletions generated by SSM are processed after any user recalls or user generated deletions.
2. Backup of the CDSs can interfere with the starting of automatic secondary space management. See "Phase 1: Backing Up the Control Data Sets" on page 183 for discussion of this consideration.

Specifying Automatic Interval Migration

The INTERVALMIGRATION parameter of the SETSYS command controls whether an DFSMShsm host does automatic interval migration. Figure 6 on page 40 shows that two different SETSYS commands are available to add to the ARCCMD*xx* member, one for DFSMShsm hosts 1 and 3, and one for DFSMShsm host 2.

For DFSMShsm hosts 1 and 3, the command is:

```
SETSYS NOINTERVALMIGRATION
```

For DFSMShsm host 2, the command is:

```
SETSYS INTERVALMIGRATION
```

In Figure 6 on page 40, only one DFSMShsm host performs automatic interval migration because one DFSMShsm host is usually sufficient. If multiple DFSMShsm hosts are requested to do interval processing, only the first one to process a given volume performs the function.

DFSMShsm host 2 performs hourly interval migration, if needed, on volumes in storage groups 1, 2, and 3 because they have AM=Y or AM=I. Storage group 4 does not have interval migration performed because it has AM=N.

DFSMShsm hosts 1 and 3 perform hourly interval migration on volumes in storage group 3 because it has AM=I, which tells DFSMShsm to ignore NOINTERVALMIGRATION for these volumes.

Specifying How to Migrate Data Sets to Migration Level 2

You can specify that DFSMShsm use only tape volumes for ML2 volumes. Parameters of the SETSYS command provide the specification for how to migrate data sets to ML2.

If you are using the duplex tape option for your ML2 tapes, see "The Duplex Tape Option" on page 402 for information specific to the duplexing environment.

Example: you can add the following SETSYS command to the ARCCMD*xx* member in each DFSMShsm host:

```
SETSYS TAPEMIGRATION(ML2TAPE(TAPE(ACL)) RECONNECT(ALL))
```

The TAPEMIGRATION parameter of the SETSYS command specifies how migration to tape occurs. The ML2TAPE subparameter specifies that migration occurs from ML1 to ML2 tape. The TAPE(ACL) option of the ML2TAPE subparameter limits the kind of tape unit that is selected to the kind of unit that is specified. This example specifies that only units that are identified by the esoteric name of ACL can be used as output tape units for ML2 volumes.

The RECONNECT parameter of the SETSYS TAPEMIGRATION command controls whether DFSMShsm attempts to reconnect a data set that is recalled from a single ML2 tape to that migration copy when the data set is next migrated.

The ALL keyword of the fast subsequent migration RECONNECT subparameter specifies that you want all eligible data sets to reconnect to their ML2 tapes. You want this to occur even if the data sets would otherwise be migrated to ML1.

Note: Only data sets that are directed to ML1 by the FORCML1 parameter (on the ARCHMIG macro) are excluded from reconnection.

Alternative subparameters that have meaning to SMS-managed data sets are NONE and NONE(ROUTETOTAPE).

The NONE subparameter specifies that ML2 volumes are DASD. NONE(ROUTETOTAPE(*unittype*)) specifies that although ML2 volumes are DASD, any management class with a value of 0 in the LEVEL 1 DAYS NON-USAGE attribute causes data sets to migrate directly to tape of the kind specified by *unittype*. The *unittype* subparameter restricts the kind of unit for the ML2 tape unit.

If you specify TAPEMIGRATION(NONE) or TAPEMIGRATION(NONE(ROUTETOTAPE(*unittype*))), you can define key ranges to be stored on different DASD ML2 volumes. The MIGRATIONLEVEL2 parameter of the DEFINE command specifies the key ranges for DASD ML2 volumes.

Guideline: If you do not intend to use tape, do not specify TAPEMIGRATION. If you do, an offline control data set is required.

With SMS, both DASD and tape can be used together by specifying the NONE or (NONE(ROUTETOTAPE) subparameters of the TAPEMIGRATION parameter and specifying LEVEL 1 DAYS NON-USAGE = 0 in some management classes. Under those specifications, level 1 to level 2 migration is to DASD but migration from DFSMShsm-managed volumes is to tape for data sets in the appropriate management classes.

When migrating SMS volumes and both PRIMARY DAYS NON-USAGE and LEVEL 1 DAYS NON-USAGE have been satisfied in an ML2 tape environment, the SMS data sets are moved directly to ML2 tape. In an ML2 DASD environment, the target is ML1 DASD unless LEVEL 1 DAYS NON-USAGE is 0.

Related reading:

- For additional information about the RECONNECT keyword, see "Specifying Fast Subsequent Migration" on page 76.

- For additional information about the DEFINE command, see the *z/OS DFSMShsm Storage Administration Reference*.

Specifying How to Do Extent Reduction

The MAXEXTENTS parameter of the SETSYS command specifies how many extents a non-VSAM single-volume data set can have before it is reduced to fewer extents.

Example: You can add the following command to the ARCCMD*xx* member in each DFSMShsm host:

```
SETSYS MAXEXTENTS(11)
```

Notes:

1. The daily statistics report output shows the number of extent reductions done each day. You can use this information to tune the SETSYS MAXEXTENTS parameter.
2. DFSMShsm does not perform extent reduction for extended format data sets.
3. Using a low MAXEXTENTS value might cause extent reduction to occur more frequently than you want. If you specify a MAXEXTENTS value, try using a value of 5 or more.

Specifying How to Scratch Expired Data Sets

During automatic primary space management, DFSMShsm scratches expired data sets that have explicit expiration dates. The EXPIREDDATASETS parameter of the SETSYS command specifies whether to scratch this type of data set. The SCRATCH subparameter specifies that during space management, DFSMShsm scratches any data set that has passed the expiration date indicated in the data set's VTOC entry.

Example: You can add the following command to the ARCCMD*xx* member in each DFSMShsm host:

```
SETSYS EXPIREDDATASETS(SCRATCH)
```

The expiration attributes in the management class control the expiration of non-VSAM data sets that do not have an expiration date in their data set VTOC entry or VSAM data sets that do not have an expiration date in the base cluster catalog entry. The management class does not control data sets with an expiration date in the data set VTOC or catalog entry. These data sets require the EXPIREDDATASETS(SCRATCH) parameter of the SETSYS command.

Note: For multivolume non-VSAM data sets, DFSMShsm looks at the VTOC of the first volume.

An alternative subparameter for the EXPIREDDATASETS parameter is NOSCRATCH. NOSCRATCH is the default value and prevents DFSMShsm from scratching data sets that have passed the expiration date indicated in the data set's VTOC entry during space management processing.

Specifying the Maximum Automatic Primary Space Management Tasks

DFSMShsm allows up to 15 primary space management tasks to run concurrently in each DFSMShsm host during automatic primary space management. Use the MAXMIGRATIONTASKS parameter of the SETSYS command to control the maximum number of automatic primary space management tasks that you want to run concurrently. If you do not specify this parameter on any SETSYS command, the default value for the maximum number of concurrent primary space management tasks is 2.

Example: You can add the following command to the ARCCMD*xx* member in each DFSMShsm hosts to handle automatic primary space management and automatic interval space management:

```
SETSYS MAXMIGRATIONTASKS(5)
```

If migration is to tape, each concurrent primary space management task that migrates a data set directly to tape requires a tape unit. If you are using the duplex tape option, two tape units per task are required. Tape units are allocated when a task encounters the first data set that is being migrated to tape, and the task does not release the tape units until it has completed processing all volumes. If migration is to SDSP data sets, each concurrent primary space management task being performed needs an SDSP data set. An SDSP data set is allocated when a task encounters the first data set that is being migrated to an SDSP, and the task does not release the SDSP until the entire volume has been processed unless RECALL or ABACKUP needs this SDSP data set.

Specifying the Maximum Automatic Secondary Space Management Tasks

DFSMShsm allows up to 30 secondary space management tasks to run concurrently within a single DFSMShsm host during automatic secondary space management. Use the MAXSSMTASKS parameter of the SETSYS command to control the maximum number of automatic secondary space management tasks that you want to run concurrently. Up to 15 tasks are available for cleanup functions and up to 15 tasks are available for tape movement functions. Secondary space management cleanup consists of the following functions:

- Expiration of migrated data sets
- Deletion of MCDS records that are no longer needed
- Deletion of SDSP records that are no longer needed.

Tape movement functions consists of moving ML1 data to ML2 tape, also referred to as ML1-to-ML2 data movement.

This SETSYS MAXSSMTASKS parameter is dynamic. If you change the number of tape movement tasks or cleanup tasks while secondary space management is active, the new number of tasks are immediately effective.

If you do not specify this parameter on any SETSYS command, the default value for the maximum number of concurrent cleanup tasks is 2 and the default value for the maximum number of concurrent tape movement tasks is 1. To aid in the processing of the deletions initiated by secondary space management, DFSMShsm has increased the default value for the SETSYS MAXRECALLTASKS from 5 to 15.

Example: You can add the following command to the ARCCMD*xx* member in each DFSMShsm hosts to handle automatic secondary space management:

```
SETSYS MAXSSMTASKS(CLEANUP(2) TAPEMOVEMENT(1))
```

Each task that you specify for tape movement requires a tape unit. If you are using the duplex tape option, two tape units per task are required.

Specifying the Maximum Automatic Interval Space Management Tasks

DFSMShsm allows up to 15 interval migration tasks to run concurrently in each DFSMShsm host during automatic interval space management. Use the MAXINTERVALTASKS parameter of the SETSYS command to control the maximum number of automatic interval migration tasks that you want to run concurrently. If you do not specify MAXINTERVALTASKS on any SETSYS command, it defaults to the value of MAXMIGRATIONTASKS.

If migration is to tape, a tape unit is required for each concurrent interval migration task that migrates a data set directly to tape. If using the duplex tape option, two tape units per task are required. Tape units are allocated when a task encounters the first data set being migrated to tape, and the task does not release the tape units until it has completed processing all volumes. If migration is to SDSP data sets, an SDSP data set is needed for each concurrent interval migration task being performed. An SDSP data set is allocated when a task encounters the first data set being migrated to an SDSP, and the task does not release the SDSP until the entire volume has been processed unless RECALL or ABACKUP needs this SDSP data set.

Specifying Recall Tasks

DFSMShsm can perform multitasking operations on a maximum of 15 tasks that recall data sets from DASD or tape and that delete migrated data sets. These are called *recall tasks*.

The MAXRECALLTASKS parameter of the SETSYS command controls the number of recall tasks that can be processed at the same time. The TAPEMAXRECALLTASKS parameter specifies how many of these tasks can be recalling data sets on ML2 tape volumes. The number of tape recall tasks must be less than or equal to the maximum number of recall tasks. If the number of active tape recalls is less than or equal to the number specified by the TAPEMAXRECALLTASKS parameter, DFSMShsm performs multitask DASD recall operations up to the number specified by the MAXRECALLTASKS parameter.

Example: In the following example, if you specified 15 recall tasks and 6 tape recall tasks and if only 3 tape recall tasks were running, DFSMShsm would start as many as 12 recall tasks from DASD. You can add this command to the ARCCMD*xx* member in each DFSMShsm host.

```
SETSYS MAXRECALLTASKS(15) TAPEMAXRECALLTASKS(6)
```

Specifying Small Data Set Packing

The SMALLDATASETPACKING parameter of the SETSYS command controls the sending of data sets to the SDSP data sets.

In specifying small data set packing, you specify the actual physical size of the data sets to be considered as candidates for migration to an SDSP data set. You can specify sizes as large as 800 KB. A candidate data set is eligible for migration to an SDSP data set if its estimated size, after compaction, is less than or equal to 400 KB or to the value you specify, whichever is smaller. The compaction size is estimated using an internal default of 50%. If the data set is not compacted, it is a candidate for migration to an SDSP if its actual size is less than or equal to 400 KB or to the value you specify, whichever is smaller.

Small data set packing causes data sets that are smaller than the size specified to be written as records in special VSAM data sets that you have allocated on the ML1 volumes. Because the minimum allocation unit is a track, writing small data sets as records in a larger data set results in saving space on the ML1 volumes.

The number of SDSP data sets defined must be at least equal to the maximum number of concurrent volume migration tasks that could be executing in your complex.

Example: You can add the following command, the command to be added to the ARCCMD*xx* member in each DFSMShsm host:

```
SETSYS SMALLDATASETPACKING(KB(110))
```

Recommendation: Have recall and ABARS use additional SDSP data sets in case some data sets become full during migration.

Related reading: For additional information about small data set packing, see "Managing SDSP Data Sets" on page 511.

Specifying the First Qualifier for Names of Migrated Data Sets

When DFSMShsm migrates a data set, it leaves the original data set name in the catalog. For use in its own records, DFSMShsm generates its own name for the data set. The form of the data set name that DFSMShsm generates is:

prefix.HMIG.T*ssmmhh.user1.user2.Xyddd*

The MIGRATEPREFIX parameter of the SETSYS command specifies the *prefix* (high-level qualifier) of the generated name. If you do not specify a migrate prefix, DFSMShsm uses the UID that you specified in the startup procedure.

Example: You can add the following command to the ARCCMD*xx* member in each DFSMShsm host:

```
SETSYS MIGRATEPREFIX(HSM)
```

The migrate prefix is used in the ACS filters that prevent DFSMShsm-owned data sets from being allocated on SMS-managed volumes. In addition, the use of the prefix allows you to use RACF generic profile protection for all your migration copies.

Related reading:

- For additional information about data set names, see "Names of Migrated Data Sets" on page 57.

- For additional information about the migrate prefix, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Specifying How to Do Migration Cleanup

During automatic secondary space management, DFSMShsm cleans up the migration control data set (MCDS) and the migration volumes by:

- Deleting expired migrated data sets
- Cleaning up SDSP data sets
- Deleting certain data sets that were not scratched during recall, deletion, or GDS roll-off
- Deleting old daily statistics and volume statistics records
- Deleting old MCDS records

You can control the age of the MCDS records and statistics records to be deleted, the number of cleanup tasks that run concurrently, and the days on which migration cleanup is done.

Data sets that are eligible for ML2 reconnection with the fast subsequent migration function may have their MCD records retained for a longer period of time than for nonreconnectable data sets. This increases the likelihood of a reconnection when the data set next migrates. However, keeping the MCD records longer (while increasing the likelihood of reconnection) also causes growth in the size of the MCDS.

The MIGRATIONCLEANUPDAYS parameter of the SETSYS command follows the form:

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

Replace *recalldays* with a decimal number between 1 and 999 that specifies the number of days that DFSMShsm keeps MCDS data set records for recalled data sets that are not compacted, or are compacted and meet or exceed the current value of SETSYS COMPACTPERCENT. For data sets that are compacted but do not meet the current value of SETSYS COMPACTPERCENT, the MCDS data set records are retained for 90 days.

Replace *statdays* with a decimal number from 1 to 999 that specifies the number of days DFSMShsm keeps the daily and volume statistics records. The REPORT command uses these records, which the IDCAMS DCOLLECT function also collects.

Replace *reconnectdays* with the number of days that you want to add to the predicted remigration date to control retention of the MCD record for reconnectable data sets. The predicted remigration date is based on the number of days that the data set was unreferenced prior to the date of its last migration. MCD records for reconnectable data sets are kept until they meet both the *recalldays* and the *reconnectdays* criteria.

You can specify different ages for MCDS records and statistics records.

Example: You can specify the following command to the ARCCMD*xx* member in each DFSMShsm host:

```
SETSYS MIGRATIONCLEANUPDAYS(25 10 3)
```

In this example, the length of time to keep MCDS records is a period of 25 days. This is because 20 days is the longest period of residence on a specified DFSMShsm-managed volume. Therefore, any data set that is recalled has a chance to migrate again before its MCDS record is scratched. Daily statistics and volume statistics records are kept for a period of 10 days. MCDS records for data sets that are marked as eligible for reconnection are kept 3 days beyond their predicted remigration date. The *reconnectdays* value can be adjusted up or down, if the data set is expected to reside on level 0 for a shorter or longer period prior to its next migration as compared to its most recent migration.

Note: The MIGRATIONCLEANUPDAYS(*recalldays*) parameter of the SETSYS command is one of the criteria that are used for deleting the MCD record of a recalled data set. However, if the data set was compacted during migration and is not eligible for compaction again, the MCD record is kept for at least 90 days. The 90 day retention occurs regardless of the value specified by *recalldays*. Also the MCD record specifies a count of the number of times that a data set by this name has migrated. This historic information is lost when the MCD record is deleted. If you plan to use this field, use this information to influence the value that you use for *recalldays*.

Related reading: For additional information about fast subsequent migration and the *reconnectdays* parameter, see "Reconnecting Data Sets with Fast Subsequent Migration" on page 76.

Identifying Migration Level 2 Volumes to DFSMShsm

If you use DASD ML2 volumes or tape volumes that are set aside only for DFSMShsm use, you must identify the volumes to DFSMShsm. The ADDVOL command does this. An example command that could be issued in at least one DFSMShsm host if you were using DASD ML2 volumes is:

```
ADDVOL ML2001 UNIT(3390) MIGRATION(MIGRATIONLEVEL2)
```

Specifying Duplexing for Migration Level 2 Tapes

The duplex tape option creates two identical tapes in parallel, with one tape designated as the original and one the alternate. To create duplex migration level 2 tapes, specify

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

If duplexing is not desired for your migration level 2 tapes, specify

```
SETSYS DUPLEX(MIGRATION(N))
```

The DFSMShsm default is no duplexing.

Figure 6 shows the total commands that you can added to the ARCCMD*xx* members for space management.

```
/* SPECIFYING WHEN TO START AUTOMATIC */
/* PRIMARY SPACE MANAGEMENT IN        */
/* PROCESSING UNIT 1                  */
SETSYS PRIMARYSPMGMTSTART(0200 0400)
DEFINE PRIMARYSPMGMTCYCLE(YNNNYNN -
  CYCLESTARTDATE(1998/09/12))

/* SPECIFYING WHEN TO START AUTOMATIC */
/* SECONDARY SPACE MANAGEMENT IN      */
/* PROCESSING UNIT 2                  */
SETSYS SECONDARYSPMGMTSTART(0030 0200)
DEFINE SECONDARYSPMGMTCYCLE(YNNNYNN -
  CYCLESTARTDATE(1998/09/12))

/* SPECIFYING AUTOMATIC INTERVAL MIGRATION */
/* IN PROCESSING UNIT 2                    */
SETSYS INTERVALMIGRATION

/* SPECIFYING NOAUTOMATIC INTERVAL MIGRATION */
/* IN PROCESSING UNITS 1 AND 3               */
SETSYS NOINTERVALMIGRATION

/* SPECIFYING LEVEL 2 TAPE MIGRATION */
SETSYS TAPEMIGRATION(ML2TAPE(TAPE(ACL))RECONNECT(ALL))

/* SPECIFYING THE MAXIMUM NUMBER OF AUTOMATIC */
/* VOLUME MIGRATION TASKS IN                  */
/* PROCESSING UNITS 1 AND 2                   */
SETSYS MAXMIGRATIONTASKS(5)

/* SPECIFYING THE MAXIMUM NUMBER OF CONCURRENT  */
/* AUTOMATIC SECONDARY SPACE MANAGEMENT TASKS   */
SETSYS MAXSSMTASKS(CLEANUP(2) TAPEMOVEMENT(1))

/* SPECIFYING EXTENT REDUCTION */
SETSYS MAXEXTENTS(6)

/* SPECIFYING SCRATCHING EXPIRED DATA SETS */
SETSYS EXPIREDDATASETS(SCRATCH)

/* SPECIFYING RECALL TASKS */
SETSYS MAXRECALLTASKS(15) -
  TAPEMAXRECALLTASKS(6)

/* SPECIFYING SMALL DATA SET PACKING */
SETSYS SMALLDATASETPACKING(KB(110))

/* SPECIFYING MIGRATED DATA SET NAME PREFIX */
SETSYS MIGRATEPREFIX(HSM)

/* SPECIFYING THE DAYS TO KEEP STATISTICS */
/* RECORDS AND MCD RECORDS                */
/* PROCESSING UNIT 2 ONLY                 */
SETSYS MIGRATIONCLEANUPDAYS(25 10 3)
```

Figure 6. DFSMShsm Commands for Space Management with SMS

Integrity Age for Space Management

When DFSMShsm does space management on a volume, it maintains data set integrity by processing only data sets that meet a certain *integrity age*. This integrity age specifies the minimum time that must elapse since the data set was last used before the data set can be processed by space management. The integrity age is affected by the DFSMShsm host environment and by the method of serialization.

Note: A multiple DFSMShsm-host environment is determined by DFSMShsm at startup time by examining whether the index component of the migration

control data set resides on a DASD volume that has been SYSGEN'd as SHARED or SHAREDUP. If this is the case, DFSMShsm performs what is called *multiple host serialization*.

Integrity age is established as follows:

- If serialization is DFHSMDATASETSERIALIZATION and the processing environment is a single DFSMShsm-host environment, the integrity age is one day.
- If serialization is DFHSMDATASETSERIALIZATION and the processing environment is a multiple DFSMShsm-host environment, the integrity age is two days.
- If serialization is USERDATASETSERIALIZATION, the integrity age is zero days.
- For temporary SMS-managed data sets in a JES3 environment, the integrity age is two days.

Automatic Space Management Operation

It has been previously pointed out how the commands to control DFSMShsm are structured. In the text that follows, examples are used to describe what DFSMShsm does in the process of automatic space management. Automatic primary and secondary space management, automatic interval migration, and automatic recall are shown.

Automatic Space Management

Figure 7 on page 42 shows the start of automatic space management. The data set names above the names of each of the groups of DASD devices and below the groups of tape devices are the names of data sets that are stored on the devices. Only the names of data sets that DFSMShsm processes during the course of this example are shown.

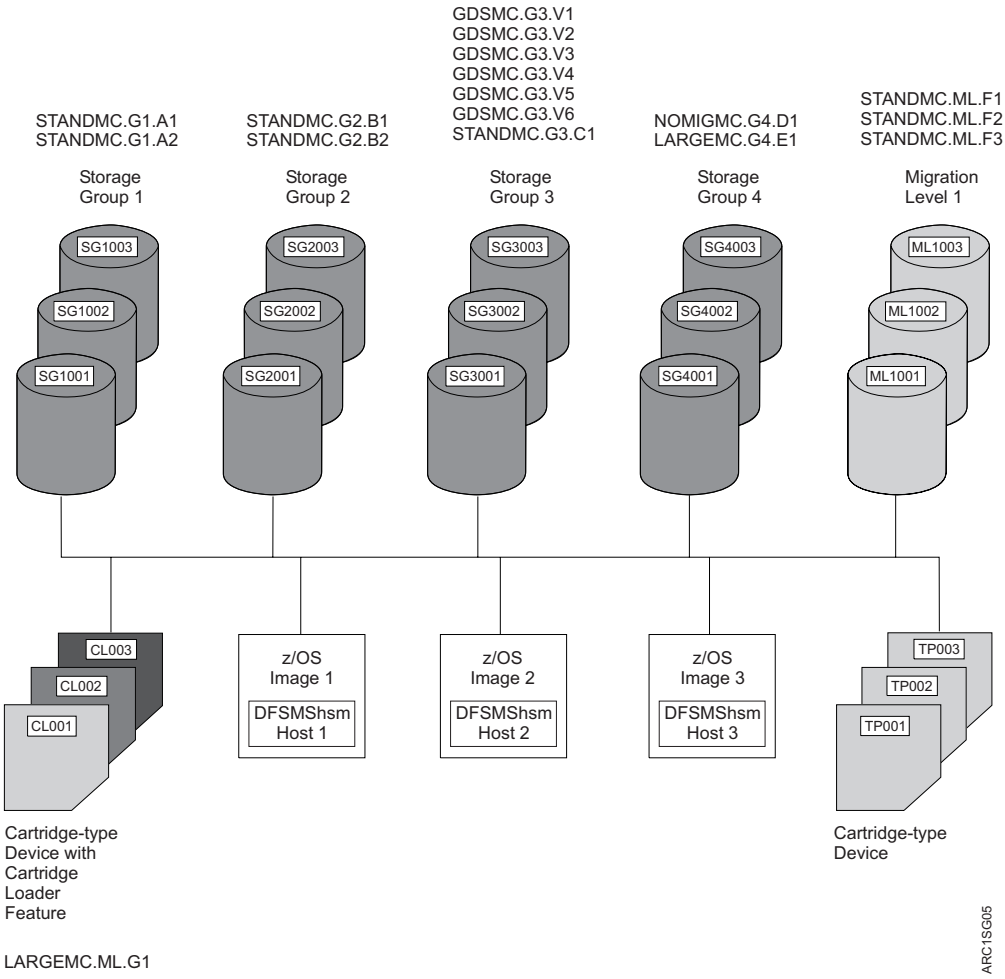


Figure 7. SMS-Managed Volumes before start of Automatic Space Management

Note: The names of the management classes have been used as high-level qualifiers for the data set names. This was done to make it easier for you to associate the data sets with the management classes. This is merely a teaching aid. You are not required to use the management class name as any part of the name of the data set.

Table 2 shows the characteristics of each of the data sets used in Figure 7. The names of the data sets are not precisely as the system might generate them. They have been shortened and modified for convenience in using them for examples.

Table 2. Data Set Characteristics for Automatic Space Management Example

Data Set Name	Days Since Created	Last Usage (Days)	Size (KB)	GDG Status
GDSMC.G3.V6	1	1	217	Not rolled off
GDSMC.G3.V5	2	2	215	Not rolled off
GDSMC.G3.V4	8	5	210	Not rolled off
GDSMC.G3.V3	10	5	207	Not rolled off
GDSMC.G3.V2	11	5	205	Not rolled off
GDSMC.G3.V1	15	5	200	Rolled off
LARGEMC.G4.E1	50	16	1200	None

Table 2. Data Set Characteristics for Automatic Space Management Example (continued)

Data Set Name	Days Since Created	Last Usage (Days)	Size (KB)	GDG Status
NOMIGMC.G4.D1	25	21	20	None
LARGEMC.ML.G1	183	120	1100	None
STANDMC.G1.A1	361	180	200	None
STANDMC.G1.A2	60	7	200	None
STANDMC.G1.A3	60	7	200	None
STANDMC.G2.B1	54	12	600	None
STANDMC.G2.B2	30	14	100	None
STANDMC.G3.C1	110	61	400	None
STANDMC.ML.F1	181	120	190	None
STANDMC.ML.F2	380	121	3	None
STANDMC.ML.F3	120	62	240	None
Note: 1KB = 1024 bytes.				

Starting Automatic Secondary Space Management

The REQUEST parameter of the SETSYS command can affect whether automatic secondary space management runs. When REQUEST is specified, DFSMShsm requests the operator's permission before starting automatic secondary space management. If the operator replies NO, automatic secondary space management does not run until the next Y day in the cycle. If the operator replies YES, automatic secondary space management runs immediately after the reply, regardless of the amount of time that elapsed before the reply.

At the beginning of automatic secondary space management, DFSMShsm requests a list of all defined management classes from SMS. If SMS does not return the list, DFSMShsm considers SMS inactive and issues a message that automatic secondary space management cannot be performed for SMS-managed data sets.

DFSMShsm also checks the last time automatic secondary space management ran, if the ending time was reached, and if all processing was complete. If the ending time was reached but processing was not complete, then the next time processing is scheduled to start, DFSMShsm will resume processing at approximately the point it stopped, if the planned start time has not been changed.

Automatic secondary space management saves the resume key for every 50,000 MCD records read. The value 50,000 is stored in the MGCRCDS# field and can be patched to any other desired value.

The order of running migration cleanup and level 1 to level 2 migration depends on the point of resumption. The level 1 to level 2 migration could have run before migration cleanup if the automatic secondary space management stopped at level 1 to level 2 migration during the last run and did not complete. The threshold of level 1 migration volumes will not be checked again before the running of level migration if the automatic secondary space management functions resume processing at a point other than the beginning.

Using secondary host promotion, you can enable another host to take over secondary space management from a failed host. Secondary space management is

usually run from only one host, so the ability to run the function from another host, should the need arise, can be quite useful.

Related reading:

- For more information about the exceptions to obtaining a new list of management classes, see "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138.
- For more information about secondary host promotion, see the sysplex chapter in *z/OS DFSMShsm Implementation and Customization Guide*.

Migration Cleanup

Before automatic secondary space management starts migration cleanup, it determines if there is any migration cleanup task running in another DFSMShsm host. If there is, the migration cleanup will not be performed on this DFSMShsm host. This avoids any unexpected results that might be created by running two migration cleanup tasks against the same MCDS at the same time.

When migration cleanup starts, it checks to see if any internal TAPECOPY processing is needed, and if so, schedules the internal TAPECOPY MWEs. Internal TAPECOPY processing could be needed because of duplexing errors or ABARS recovery of ML2 volumes.

Migration cleanup deletes the following:

- Expired migrated data sets. Migration cleanup deletes the data sets if:
  - For both SMS-managed and non-SMS-managed data sets, the data set has passed the expiration date indicated in the data set's VTOC entry and the EXPIREDDATASETS(SCRATCH) parameter is specified in the SETSYS command.
  - For SMS-managed data sets, the data set does not have an expiration date but has passed the expiration attributes specified in the management class to which it is associated.
  - For an SMS-managed NOSCRATCH GDS, the data set rolled off while on a level 0 volume and migrated, with rolled off GDG=EXPIRE in the management class and no expiration date in the VTOC.
- Migration copies of the data sets recalled from SDSP data sets. Because multiple recall tasks can access the small data set packing data sets concurrently, recall does not erase the migration copy.

  Even though SDSP data sets used during migration cleanup are periodically released when not in use, and SDSP data sets needed by recall or aggregate backup are released when needed, it is recommended that you do not run secondary space management when automatic primary space management is running in order to avoid contention for SDSP data sets.

  Every 20 seconds, migration cleanup checks to determine if recall or aggregate backup needs an SDSP data set. The value 20 is stored in the MCVT#CHK field and can be patched to any other desired value.
- MCDS data set (MCD) records for nonreconnectable data sets that are older than the number of days that are specified by the *recalldays* value of the SETSYS MIGRATIONCLEANUPDAYS command.
- MCDS data set (MCD) records for reconnectable data sets that have passed their predicted remigration date by the number of days that are specified by the *reconnectdays* value of the SETSYS MIGRATIONCLEANUPDAYS command. This occurs only if the *recalldays* criterion is also met.

- Daily statistics records and volume statistics records that are older than the number of days specified by the *statdays* value from the SETSYS MIGRATIONCLEANUPDAYS command.
- Migration copies not scratched during recall, deletion, or GDG roll-off of non-SMS-managed migrated data sets. These copies exist if an error occurred during recall or deletion, or if a generation data set had not passed its expiration date when the roll-off occurred.

Migration Cleanup Example

Figure 8 shows an example of SMS-managed volumes after migration cleanup has occurred for the conditions shown in Figure 7. Data set STANDMC.ML.F2 is deleted from the ML1 volumes because it has passed the EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS values from management class STANDMC. Data set LARGEMC.ML.G1 is deleted from the ML2 tapes because it has passed the EXPIRE AFTER DAYS NON-USAGE and the EXPIRE AFTER DATE/DAYS values from the management class LARGEMC.

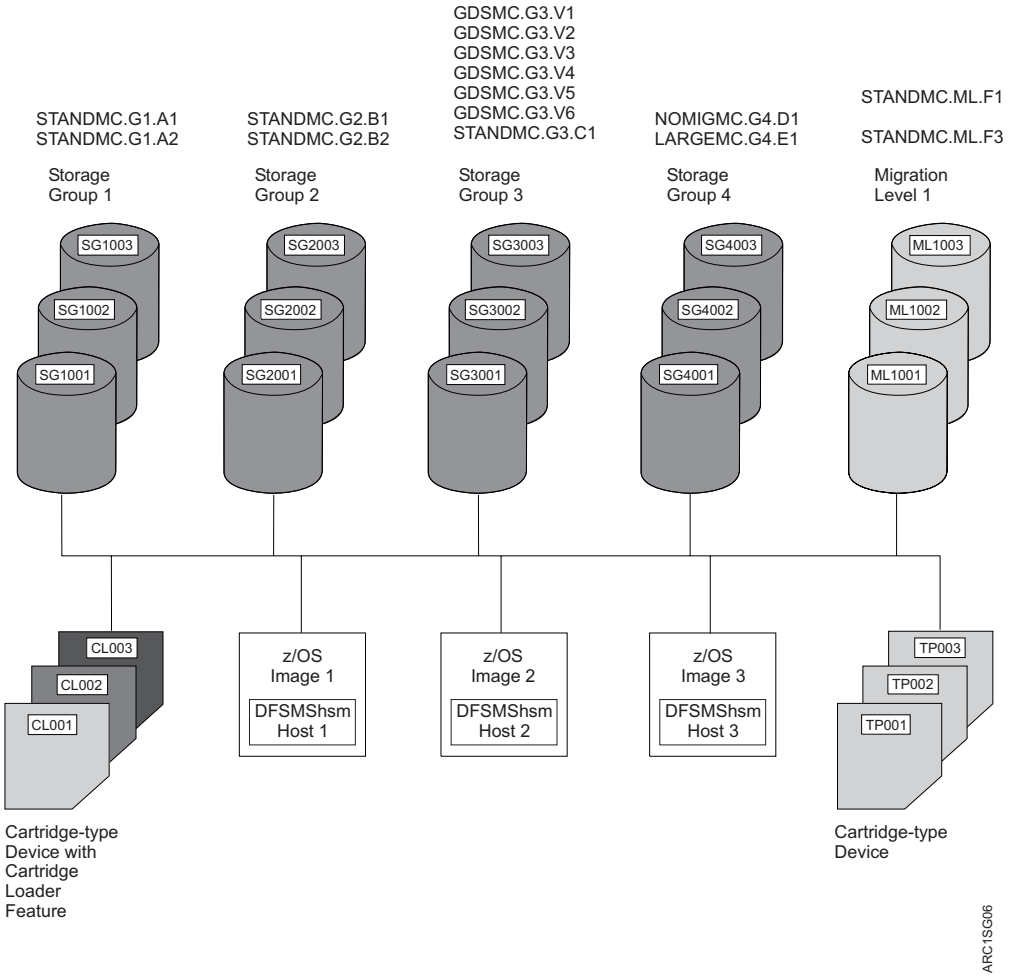


Figure 8. SMS-Managed Volumes after Migration Cleanup

Level 1 to Level 2 Migration

Before automatic secondary space management starts level 1 to level 2 migration, it checks if there is any command level 1 to level 2 running. If yes, the level 1 to level 2 migration of automatic secondary space management will not start. There is no reason to run two level 1 to level 2 migrations at the same time.

Before automatic secondary space management starts level 1 to level 2 migration, it does a space check on the ML1 volumes. If any ML1 volume has an occupancy that is equal to or greater than its high threshold (specified in the ADDVOL command), *all* eligible data sets migrate from *all* ML1 volumes to ML2 volumes. Eligible SMS-managed data sets are those data sets that have passed the value you specified with the LEVEL 1 DAYS NON-USAGE attribute in their management class. Eligible non-SMS-managed data sets are those that have passed the value you specified with the MIGRATIONLEVEL1DAYS parameter of the SETSYS command.

ML2 volumes can be either DASD or tape. The TAPEMIGRATION parameter of the SETSYS command specifies what type of ML2 volume is used. The SETSYS command for DFSMShsm host 2 specifies ML2 migration to tape.

Related reading: For additional information about data set eligibility from Level 1 to Level 2, see "Specifying When Data Sets Become Eligible for Level 1 to Level 2 Migration" on page 81.

Migration from SDSP Data Sets

Although SDSP data sets used during SSM migration cleanup are periodically released when not in use, it is still recommended that you do not run secondary space management when automatic primary space management is running in order to avoid possible contention for SDSP data sets.

Every 20 seconds, level 1 to level 2 migration checks to determine if recall or aggregate backup needs an SDSP data set. The value 20 is stored in the MCVT#CHK field and can be patched to any other desired value.

Migration to a Tape Migration Level 2 Volume

At the beginning of a level 1 to level 2 migration task, DFSMShsm selects the partial tape with the largest percentage that is written. This might be one of the following tapes:

- The tape last used by this task
- A tape ML2 volume from the list of not-associated, partially empty tape volumes
- An empty tape (if available), but only if no partially empty tape is available

DFSMShsm selects tapes differently if you are using the duplex tape option for migration tapes.

DFSMShsm does not migrate a data set to a tape ML2 volume if the data set needs to be backed up.

When a tape unit and a tape volume are allocated, they remain allocated and mounted until the migration is completed or the volume becomes full or is needed by another function. When level 1 to level 2 migration is completed, the tape volume is deallocated and demounted.

Related reading:

- For more information about the duplex tape option, see "The Duplex Tape Option" on page 402.
- For more information about contention for migration volumes, see "Tape Migration Level 2 Volume Contention" on page 117.

Migration to a DASD Level 2 Volume

For each data set that migrates from an ML1 volume to DASD ML2, DFSMShsm uses the set of initial characters of the data set name and the key ranges that you defined to choose the DASD ML2 volume. The *key range* is the total set of data set names whose initial characters are between a low key and a high key in collating sequence.

If DFSMShsm fails the migration of data sets from an ML1 volume to an ML2 volume because the ML2 volume being used with that key range does not have enough space, DFSMShsm does the following:

1. Disassociates the full volume from the key range.
2. Tries to assign an available DASD ML2 volume to the key range in place of the full volume. An available DASD ML2 volume is a volume that meets all of the following criteria:
   - It has been added to DFSMShsm as an ML2 volume. The ADDVOL *volid* MIGRATION(MIGRATIONLEVEL2) command adds ML2 volumes to DFSMShsm.
   - It has not been associated with a key range. The DEFINE MIGRATIONLEVEL2(KEYS(*key . . .*) VOLUMES(*volid . . .*)) command associates ML2 volumes with key ranges.
   - DFSMShsm has never chosen it as a replacement for an ML2 volume.
   - It does not have the DRAIN attribute in its ADDVOL command.
3. Retries the migration of the data set one time.

If there is no ML2 volume to choose as a replacement volume, DFSMShsm stops level 1 to level 2 migration for the key range defined for the full ML2 volume. DFSMShsm continues migrating data sets for all other key ranges.

Related reading:

- For more information about how to define key ranges, see the *z/OS DFSMShsm Storage Administration Reference*.
- For more information about what to do to make new ML2 volumes available when all ML2 volumes are full, see "Making Migration Level 2 Key Range Volumes Available" on page 506.

Output Media Error Handling

During level 1 to level 2 migration of automatic secondary space management processing, if an error condition occurs on any output media (tape or DASD), a message is sent to the operator's console to inform the operator of the error condition, and the function is stopped. DFSMShsm does not remove this message from the console; therefore, the operator must issue a console command to remove the message.

If you are using the duplex tape option for your migration tapes, and the error condition occurs on the alternate volume, the function is not stopped.

When the error condition that caused the level 1 to level 2 migration to end is corrected, the operator can restart level 1 to level 2 migration if the current time is within the automatic secondary space management window. You can use one of the following commands to restart level 1 to level 2 migration:

- RELEASE MIGRATION(AUTO)
- RELEASE MIGRATION

- ADDVOL (volume of the device type in error) and RELEASE MIGRATION or RELEASE MIGRATION(AUTO)

Related reading: For additional information about the duplex tape option, see "The Duplex Tape Option" on page 402.

Installation Exits Called during Level 1 to Level 2 Migration

Two installation exits can be called during level 1 to level 2 migration: the tape data set exit and the second-level migration data set exit.

The tape data set exit (ARCTDEXT) receives control (if it is enabled) when an output tape data set is opened if the tape security option is expiration date (EXPIRATION) or expiration date including password-protected data sets (EXPIRATIONINCLUDE). You can use the tape data set exit to change the expiration date to be recorded in the tape label to a date other than 99365.

The second-level migration data set exit (ARCMMEXT) receives control when a migrated data set is selected for migration. You can use this exit to further control whether data sets migrate from ML1 volumes to other migration volumes. For example, you can use this exit to hold small data sets in SDSP data sets on ML1 volumes instead of allowing them to migrate to ML2.

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

Example of Level 1 to Level 2 Migration

Figure 9 on page 49 shows SMS—managed volumes after level 1 to level 2 migration. In this figure, at least one of the ML1 volumes had reached or exceeded its threshold of occupancy when DFSMShsm performed its space check. Data set STANDMC.ML.F3 migrates from the ML1 volumes to the tape ML2 volumes because it has passed the LEVEL 1 DAYS NON-USAGE value from management class STANDMC.

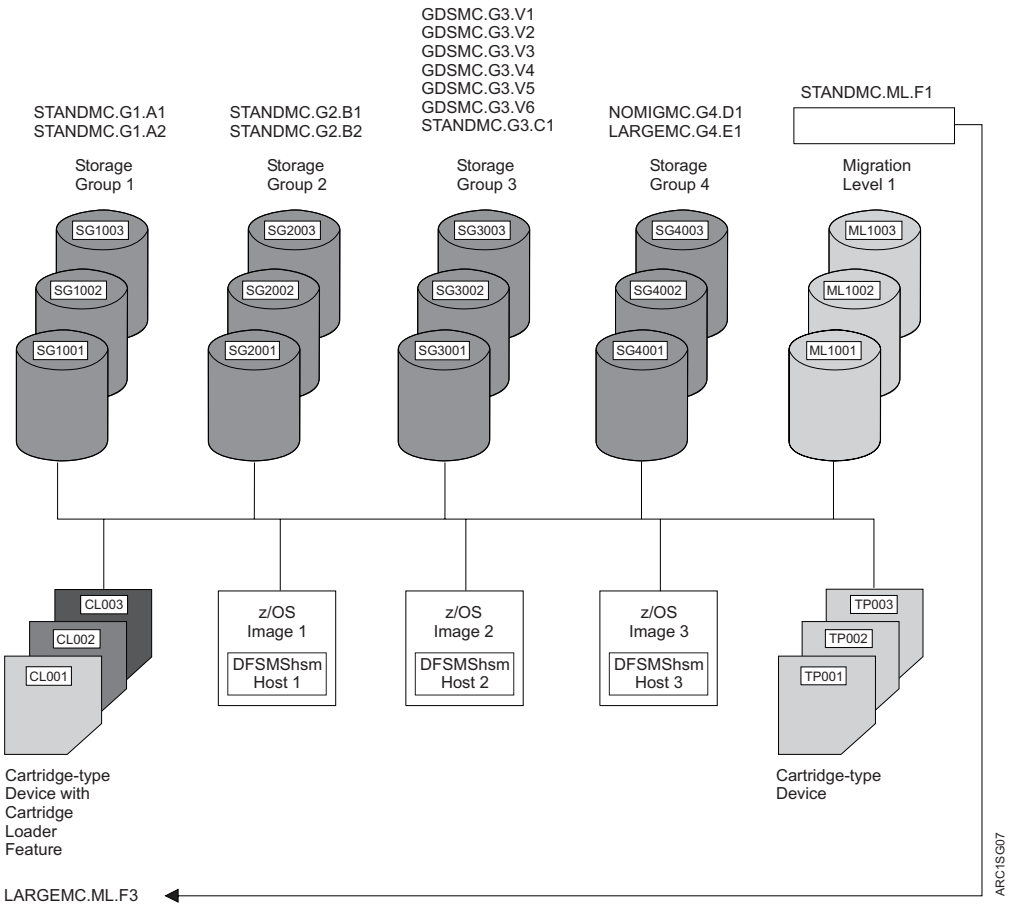


Figure 9. SMS-Managed Volumes after Level 1 to Level 2 Migration

Starting Automatic Primary Space Management

At the beginning of automatic primary space management in each of the DFSMShsm hosts, if SMS is active, DFSMShsm requests a list of all defined management classes and storage groups from SMS. If you have not provided any management class definitions, DFSMShsm builds a default management class definition. If SMS does not return any storage group definitions, DFSMShsm considers SMS to be inactive.

If SMS is active, DFSMShsm processes any SMS volumes that the z/OS image is allowed to process and that have the AM=Y, AM=I, or AM=P attribute, and any non-SMS-managed primary volumes. If SMS is not installed and active, DFSMShsm processes only non-SMS-managed primary volumes. For purposes of the example, it is assumed that SMS is installed and active.

Related reading: For additional information about the exceptions to obtaining a new list of management classes, see "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138.

Automatic Primary Space Management

At the beginning of automatic primary space management, DFSMShsm determines which volumes are candidates for processing. Volumes in storage groups that have a z/OS image specified by the MIGRATE SYSTEM/SYSPLEX NAME have a z/OS image affinity.

DFSMShsm uses the following order of processing as a method of controlling input path contention. This order is subject to prioritizing for SMS volumes restricted to processing by one z/OS image.

1. Affinity SMS volumes
2. Retry of in-use affinity SMS volumes
3. Non-SMS and non-affinity SMS volumes

After DFSMShsm has attempted to process all the candidate volumes eligible for automatic primary space management, it retries the selection of any of these volumes that were found to be in use by another DFSMShsm function.

If multiple DFSMShsm hosts perform automatic primary space management, only the first host to process a given volume performs the function.

DFSMShsm-managed volume space management is performed in two passes.

1. **Pass 1** deletes the appropriate data sets, releases unused space, performs fast subsequent migration, and determines which data sets on the volume are candidates for migration or extent reduction.
2. **Pass 2** processes the candidates for extent reduction and migrates data sets until the low threshold of occupancy is reached.

Pass 1 of Automatic Primary space Management

During pass 1, DFSMShsm examines every data set on a volume, independent of the occupancy of the volume, without moving data sets. Unless the data set is an unmatched data set, DFSMShsm compares the integrity age with the unused age to make sure that it is allowed to process the data set. The unused age is the difference between the current date and the date-last-referenced.

Rule: The unused age must be greater than or equal to the integrity age.

Pass 1 of DFSMShsm-managed volume space management performs the following functions:

- Issues a message for unmatched data sets
- Deletes temporary data sets
- Deletes expired data sets
- Determines data sets that are eligible for migration
- Excludes from migration to tape those data sets that are eligible to be backed up
- Determines data sets that are eligible for extent reduction
- Releases unused space for eligible data sets
- Performs fast subsequent migration for eligible data sets

For each of the data sets, DFSMShsm uses the attributes specified in the management class with which the data set is associated. If no management class is associated with the data set, DFSMShsm uses the default management class.

Related reading:

- For additional information about the integrity age, see “Integrity Age for Space Management” on page 40.

- For additional information about unmatched data sets, see "Unmatched Data Sets."
- For additional information about default management classes, see "Using the Default Management Class" on page 28.

Unmatched Data Sets

If the catalog entry, the data set VTOC entry, or both are missing, the data set is an unmatched data set. For an unmatched data set, the space management and backup exit (ARCSAEXT) is invoked, if the exit is enabled. If the exit indicates that processing continues, or if the exit is not enabled, DFSMShsm issues a message, does no further processing on the unmatched data set, and begins processing the next data set. If the exit indicates that processing is not to continue, DFSMShsm begins processing the next data set without issuing any message.

Deleting Temporary Data Sets

If for some reason (such as a system outage) SMS-managed temporary data sets are not deleted at end of job, DFSMShsm attempts to delete them during automatic primary space management. A flag in the data set VTOC entry indicates that a data set is temporary.

If no cross-system serialization exists, or if cross-system serialization is not performed for temporary data sets, an in-use temporary data set can be deleted. In a JES3 environment, the data set must be two days old before it can be deleted.

Deleting Expired Data Sets

If the EXPIREDDATASET(SCRATCH) parameter is specified in the SETSYS command, DFSMShsm deletes any non-VSAM data set whose VTOC entry shows that the data set has passed its expiration date and any VSAM data set whose catalog entry shows that the data set has passed its expiration date. The management class expiration attributes control the expiration of non-VSAM data sets that do not have an expiration date in their data set VTOC and VSAM data sets that do not have an expiration date in their base cluster catalog entry.

Note: For multivolume non-VSAM data sets, DFSMShsm looks at the VTOC of the first volume.

If concurrent copy was used to back up the data set and a system failure or cancel of DFSMShsm occurred after the concurrent copy session was established, then the data-set-changed indicator can be off even though the data set was not successfully backed up. This can result in the data set prematurely meeting the expiration criteria.

DFSMShsm provides a patch byte that enables users to override the requirement that an SMS-managed data set have a backup copy before it is expired. For more information about this patch, refer to Chapter 16, "Tuning DFSMShsm" in the *z/OS DFSMShsm Implementation and Customization Guide*.

DFSMShsm deletes rolled-off generation data sets that meet either of the following conditions, regardless of the EXPIREDDATASETS parameter of the SETSYS command.

- It is associated with a management class that has the attribute ROLLED-OFF GDS ACTION=MIGRATE and both of the following apply:
  - It does not have an expiration date in the data set VTOC entry.
  - It has passed the date and time limit specified by the EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS attributes in the management class.

- It is associated with a management class with the attribute ROLLED-OFF GDS ACTION = EXPIRE and it has no expiration date in the data set VTOC entry.

Determining Eligible Data Sets for Migration

DFSMShsm checks the COMMAND OR AUTO MIGRATE attribute in the management class to which the data set is associated. If the value of the attribute is BOTH, DFSMShsm determines whether the data set is eligible for migration. Four conditions determine whether a data set is eligible for migration:

- The MIGRATE value of the ROLLED-OFF GDS ACTION attribute
- Whether the data set is a generation that exceeds the value specified by the # GDG ELEMENTS ON PRIMARY attribute in the management class
- Whether the data set is an SMS-managed MVS™ or IMS™ GSAM checkpointed data set
- The number of days since the data set was last used

Rolled-off generation data sets that belong to a management class with the attribute ROLLED-OFF GDS ACTION=MIGRATE are always identified as migration candidates with the highest migration priority.

Non-rolled-off generation data sets that exceed the value specified with the # GDG ELEMENTS ON PRIMARY attribute for their management class are always identified as candidates with priority for early migration. The priority of such data sets is just below that for rolled-off generation data sets.

For both command and automatic migration, DFSMShsm determines whether the data set being migrated is a checkpointed data set. DFSMShsm migration, including extent reduction, is delayed for a fixed number of days for SMS-managed MVS and IMS GSAM checkpointed data sets. The default delay is five days. If the data set is checkpointed, DFSMShsm determines whether the data set is eligible for migration. A checkpointed data set is eligible for migration when the date-last-referenced, plus the number of days the data set is to be treated as unmovable, are less than or equal to the current date. For data set command migration, error messages are issued for ineligible data sets.

DFSMShsm provides a patch byte that enables users to modify the number of days that must have elapsed since the date-last-referenced for a checkpointed data set to be eligible for migration. For more information about this patch, refer to Chapter 16, “Tuning DFSMShsm” in the *z/OS DFSMShsm Implementation and Customization Guide*.

All other data sets, including generation data sets that were not given special migration priority, are identified as migration candidates if the time since they were last used meets or exceeds the value specified by the PRIMARY DAYS NON-USAGE attribute for their management class.

If the COMMAND OR AUTO MIGRATE attribute in a data set’s management class is set to BOTH, with the PRIMARY DAYS NON-USAGE attribute set to 9999, and the # GDG ELEMENTS ON PRIMARY attribute set to 1, then relative zero-generation data sets can remain on a user’s primary disk for up to 30 years. (The # GDG ELEMENTS ON PRIMARY attribute overrides the PRIMARY DAYS NON-USAGE attribute.) When a generation moves from zero to -1, it becomes eligible to migrate immediately, independent of the PRIMARY DAYS NON-USAGE value.

When a data set is identified as a migration candidate, it is also identified for the target migration volume device type: DASD or tape. The device type chosen depends on the value in the LEVEL 1 DAYS NON-USAGE attribute of its management class. If the value of the attribute is 0, the data set is identified for migration to ML2 tape. If the value of the attribute is 1–9999 or NOLIMIT, the data set is identified for migration to ML1 DASD.

Related reading: For additional information about DFSMShsm support for checkpointed data sets, see Chapter 4, "User Data Sets" in the *z/OS DFSMShsm Implementation and Customization Guide*.

Determining Eligible Data Sets for Backup

After DFSMShsm determines that a data set is eligible for migration, it checks whether the data set needs to be backed up. A data set must be backed up if all of the following criteria are met:

- The data-set-changed indicator in the data set VTOC entry indicates that the data set has been changed.
- The management class contains the attribute AUTO BACKUP=Y.
- The management class ADMIN OR USER COMMAND BACKUP attribute contains either of the values ADMIN or BOTH.
- The storage group has the attribute AUTO BACKUP=Y.

DFSMShsm identifies any migration-eligible data set that must be backed up. If the data set that must be backed up is identified for migration to tape, DFSMShsm removes the identification as a migration candidate.

If concurrent copy was used to backup the data set and a system failure or cancel of DFSMShsm occurred after the concurrent copy session was established, then the data-set-changed indicator can be off even though the data set was not successfully backed up. If this happens, migration will not identify the data set as needing a backup copy.

Determining Eligible Data Sets for Extent Reduction

After determining if the data set is eligible for backup, DFSMShsm determines if the data set is eligible for extent reduction. DFSMShsm identifies a non-VSAM single-volume data set as eligible for extent reduction if all of the following criteria are met:

- The data set has as many extents or more than the value specified by the MAXEXTENTS parameter of the SETSYS command.
- The data set also meets data integrity age.
- If the data set is an SMS-managed MVS or IMS GSAM checkpointed data set, it meets the requirements for date-last-referenced plus the number of days the data set is to be treated as unmovable.
- The value of the COMMAND OR AUTO MIGRATE attribute is BOTH.

Notes:

1. DFSMShsm does not perform extent reduction on any extended format sequential data set.
2. Extent reduction will not occur on SMS-managed data sets with COMMAND OR AUTO MIGRATE=NONE specified, or on non-SMS-managed data sets with SETMIG NOMIGRATION specified.

Releasing Unused Space

For all physical sequential, partitioned, and extended format VSAM KSDS single-volume data sets that belong to a management class with the attribute PARTIAL RELEASE=Y (or C for VSAM), DFSMShsm invokes the PARTREL function of DADSM if either of the following conditions is met:

- The data set is not a candidate for extent reduction.
- At least one track will be released by PARTREL.

DFSMShsm issues a message indicating the result of the partial release and the number of tracks freed for each data set if the activity logging message level is FULL or REDUCED.

Performing Fast Subsequent Migration

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

Related reading: For additional information about the eligibility checks that affect reconnection, see "Reconnecting Data Sets with Fast Subsequent Migration" on page 76.

Pass 1 Processing Example

Figure 10 on page 55 shows the results of pass 1 processing. Data set STANDMC.G1.A1 was deleted because it exceeded the EXPIRE AFTER DATE/DAYS and EXPIRE AFTER NON-USAGE values of the STANDMC management class. Data set GDSMC.G3.V1 was deleted because it is a rolled-off GDS with the EXPIRE value in the ROLLED-OFF GDS ACTION attribute in GDSMC management class. Data set NOMIGMC.G4.D1 was deleted because it exceeded the EXPIRE AFTER DAYS NON-USAGE attribute of the NOMIGMC management class. Data set STANDMC.G1.A3, which has not changed from Figure 9 on page 49, was recalled from ML2. It is reconnected during Pass 1 to its original ML2 tape.

Note: The data sets in the NOMIGMC management class never migrate because the COMMAND OR AUTO MIGRATE attribute has the value of NONE, but they are processed for expiration and for PARTIAL RELEASE.

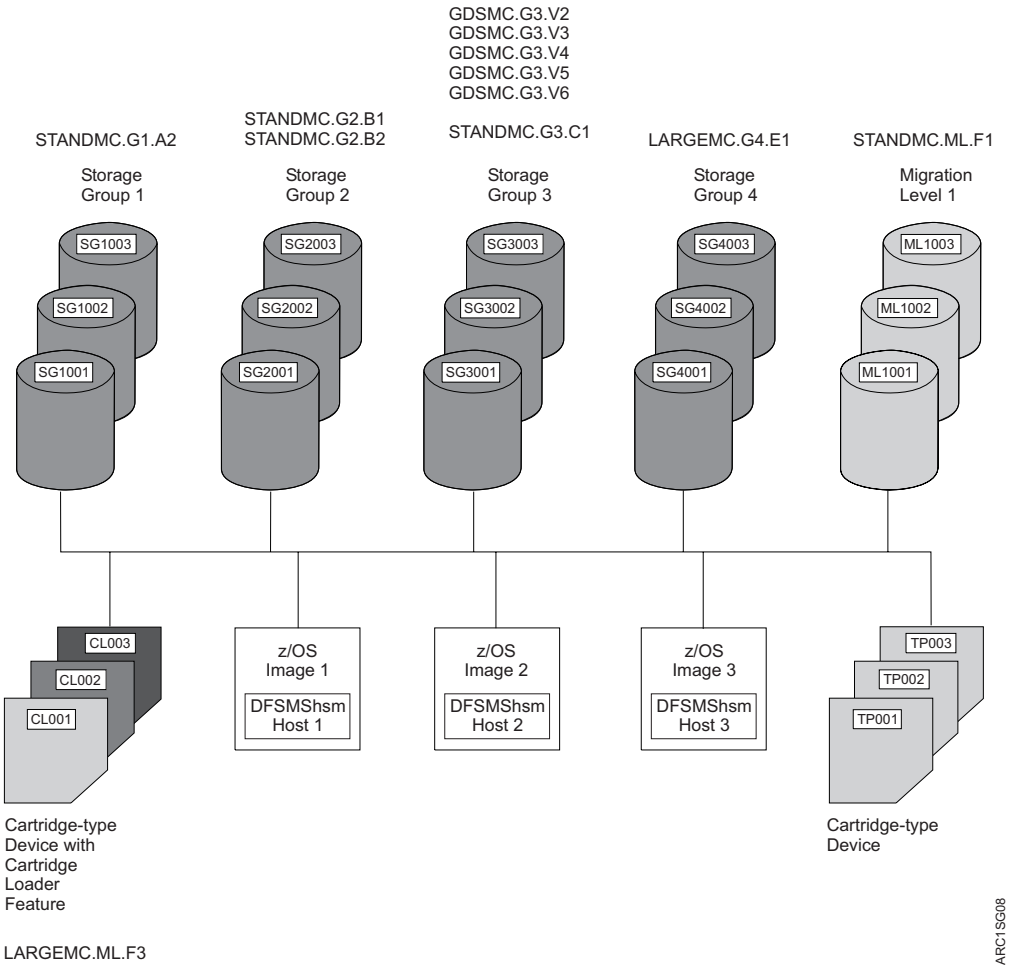


Figure 10. SMS-Managed Volumes after DFSMShsm-Managed Volume Management Pass 1

Pass 2 of Automatic Primary Space Management

During pass 2 of automatic primary space management, DFSMShsm migrates data sets from the volumes until the space occupied on the volumes is equal to or less than the space specified by the low-threshold attribute contained in the storage groups to which the volumes belong. DFSMShsm migrates only those data sets that have been identified for migration or extent reduction by pass 1 of DFSMShsm-managed volume space management. The following steps are performed in pass 2:

1. Migrate data sets that are candidates both for migration and for extent reduction.
2. Migrate data sets that are candidates only for extent reduction. Schedule a recall for each data set.
3. Place the remaining candidates for migration in priority order.
4. Migrate data sets in priority order until the low threshold of occupancy is reached.

Migrate Candidates for Both Migration and Extent Reduction

Any data set that is identified as a candidate for extent reduction always migrates. However, a data set that is identified as a candidate for both migration and extent reduction is not recalled.

Migrate Candidates for Extent Reduction

DFSMShsm migrates the data sets that are candidates for extent reduction and schedules recalls for those same data sets. During the migration, DFSMShsm moves only the valid data from the data sets. Therefore, when the data sets are recalled into fewer extents, they may occupy less space.

Place Candidate Data Sets in Priority Order for Migration

After migrating data sets for extent reduction, DFSMShsm places the remaining candidates for migration in order by priority. DFSMShsm gives the highest priority for migration to rolled-off generation data sets. It gives second priority to non-rolled-off generation data sets that have been identified for early migration. These are the non-rolled-off data sets that exceed the number specified by the # GDG ELEMENTS ON PRIMARY attribute.

After the two highest priority groups, priority is assigned to all other data sets on the basis of size and age. The age, for migration priority purposes, is an eligibility age determined from the last-used date and the value of the PRIMARY DAYS NON-USAGE parameter. Small data sets must be eligible for a longer time to attain the same or higher priority as larger data sets.

Migrate Data Sets

After putting the candidate data sets in priority order, DFSMShsm compares the occupied space on the volume to the low threshold for the volume. If the occupied space is more than the low threshold, DFSMShsm migrates the highest priority data set.

Data sets that fail reconnection remain eligible for normal migration. Data sets that fail reconnection because there is no ability to serialize either the data set or the target ML2 volume are assigned the lowest migration priority.

Data sets are migrated until the occupied space is equal to or less than the low threshold specified by the storage group or until no more migration candidates exist on the volume. At that point, DFSMShsm ends automatic primary space management processing for that volume.

As DFSMShsm selects each data set for migration, it transfers control to the migration exit (ARCMDEXT) if the exit is installed and enabled.

If any data set is directed to tape during DFSMShsm-managed volume space management, a tape volume is allocated and mounted. If you are using the duplex tape option for migration, two tape volumes are allocated and mounted. When a tape volume is allocated for DFSMShsm-managed volume migration, it remains allocated until it becomes full or is needed for some other DFSMShsm function. When the DFSMShsm host finishes DFSMShsm-managed volume migration, DFSMShsm deallocates and demounts the tape volume.

Related reading:

- For additional information about ARCMDEXT, see *z/OS DFSMS Installation Exits*.
- For additional information about having tapes available during DFSMShsm-managed volume space management, see "Automatic Space Management Operation" on page 41.

Output Media Error Handling

You can use both tape and DASD as migration devices concurrently. During such an operation, if migration to either tape or DASD is ended because of an error condition, a message is issued, but migration continues to the device type that is

not affected by the error condition. A highlighted message is sent to the operator's console to inform the operator of the error condition. DFSMShsm does not remove this message from the console; the operator must issue a console command to remove it.

After correcting the error condition that caused the migration to tape or DASD to end, the operator can allow DFSMShsm to resume migration to the device that was in error by issuing one of the following commands:

- RELEASE MIGRATION(AUTO)
- RELEASE MIGRATION
- ADDVOL (for a volume of the device type in error)

If error conditions make both tape and DASD unavailable as migration targets, migration processing ends, a message is issued to the operator, and migration is held.

Names of Migrated Data Sets

When DFSMShsm migrates a data set, it changes the catalog entry for that data set. The data set retains its name, so the users of the data set know how to address it, but the location is changed to MIGRAT. However, DFSMShsm gives the migration copy of the data set a name of the form:

prefix.HMIG.T*ssmmhh.user1.user2.Xyddd*

where:

- *prefix* is the prefix you specify with the SETSYS MIGRATEPREFIX command.
- HMIG and T are constants.

  Note: If two or more data sets with the same first two high-level qualifiers are migrated during the same second on the same day, DFSMShsm assigns a letter other than T to one of these data sets.
- *ssmmhh* is the time in seconds, minutes, and hours.
- *user1* and *user2* are the first two qualifiers of the data set name.
- *Xyddd* is the year and day. X is replaced with a letter that represents the decade as follows:

G = 1970	B = 2020
H = 1980	C = 2030
I = 1990	D = 2040
J = 2000	E = 2050
A = 2010	F = 2060

If, for example, I9012 replaces *Xyddd*, then I9012 represents January 12, 1999.

If you need to access a single-file format tape, you must specify a valid data set name for the single file that comprises up to 255 tape volumes. The following migrated data set name is valid for each set of cartridge-type tape volumes in single-file format:

prefix.HMIGTAPE.DATASET

Only the last 17 characters (.HMIGTAPE.DATASET) are recorded in the tape label. To access the single-file format data set, you must supply any valid *prefix* along with the .HMIGTAPE.DATASET qualifiers.

The following tape data set name for each copy of a cartridge-type tape volume is created by the TAPECOPY command or the duplex tape option:

prefix.COPY.HMIGTAPE.DATASET

Compaction during Migration

If you specify the compaction during migration option, DFSMShsm compacts each data set as it migrates. The first time the data set migrates, DFSMShsm always compacts it. If a migrated data set is recalled and is again a candidate for migration, DFSMShsm checks the compaction history in the MCDS for that data set. If the compaction from the earlier migration did not result in saving at least the percentage of space that you specified with the COMPACTPERCENT parameter of the SETSYS command, DFSMShsm does not perform compaction again.

Note: If the data set is a SAM or VSAM compressed data set, DFSMShsm suspends compaction during migration and records in the MCDS that the data set is striped, compressed, or both.

If the data set is to migrate to an SDSP, the compaction size is always estimated using an internal default of 50%. Previous compaction history is not considered.

Space Management of VSAM Data Sets

DFSMShsm migration functions that exist for non-VSAM data sets also exist for VSAM data sets that are listed in the integrated catalog facility catalog.

The migration of a VSAM data set is performed only on a complete VSAM sphere. The base cluster and all alternate index clusters defined on the base cluster are migrated as part of the data set and are not specified separately. Paths are included automatically in the data set migration command.

When DFSMShsm migrates a VSAM data set, it tracks a maximum of eight component names for that data set. See Figure 11 on page 59, which shows component name limitations when DFSMShsm is recalling a migrated VSAM data set.

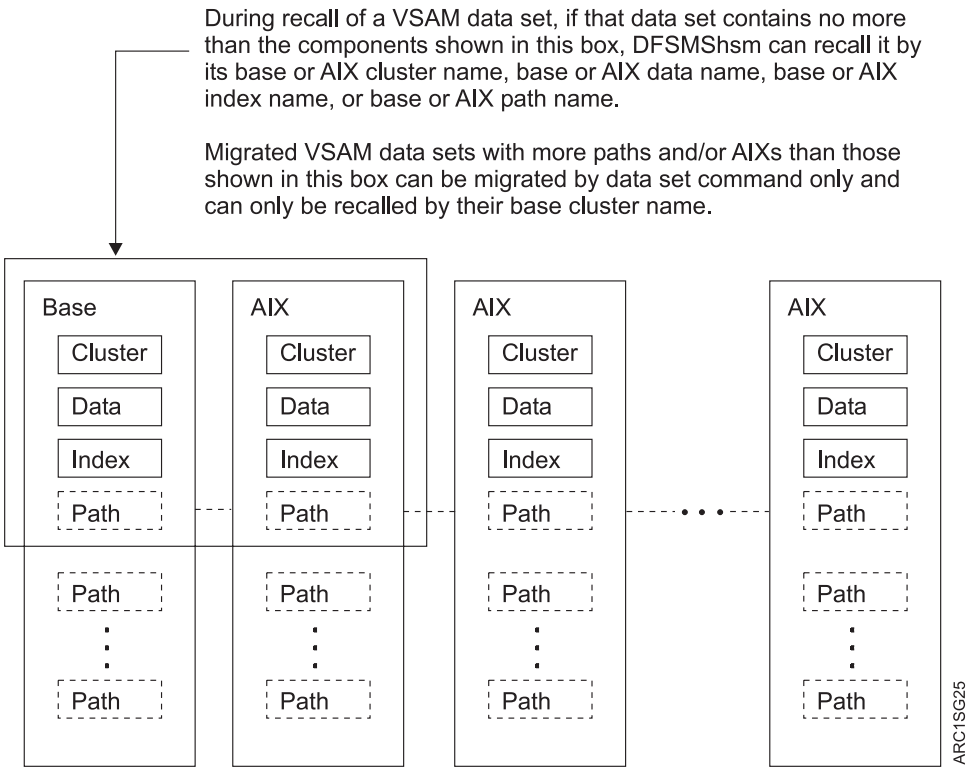


Figure 11. Component Name Limitations when Recalling Migrated VSAM Data Sets

DFSMShsm can migrate multiple-volume SMS-managed VSAM data sets when it processes the volume containing the base data component. DFSMShsm does not migrate a sphere that has a component that needs to be verified.

Data sets cataloged in the integrated catalog facility catalog with more than one alternate index (AIX), more than one path on the base, or more than one path defined on the AIX are not selected for migration during volume processing. These data sets can be migrated by data set command.

DFSMShsm does not migrate a VSAM data set marked for forward recovery, or one with retained locks.

Size Eligibility for Data Sets to Migrate to an SDSP

Data sets that meet certain size requirements can be migrated to SDSP data sets. Data set size is determined as follows:

- **Physical sequential data sets**—by the high-used track, which is stored in the data set VTOC entry
- **Physical sequential data sets with zero secondary allocation**—by allocated size
- **Partitioned data sets (not PDSE)**—by the high-used track, which can include unused space within the data set

  Partitioned data sets can only migrate to SDSPs when the following two conditions are met: (1) the migration request is for a volume (not for a data set) and (2) an AX cell exists in the catalog for the PDS. In these cases, eligible PDSs are migrated to SDSPs using DFSMSdss™ as the data mover.
- **Direct access data sets**—by allocated size
- **VSAM data sets**—by allocated size address (RBA)
- **PDSE data sets**—by the high-used page, which is obtained from DFSMSdfp
- **HFS data sets**—by the high-used page, which is obtained from DFSMSdfp
- **Unmounted zFS data sets**—by allocated size address (RBA)

A candidate data set is eligible for migration to an SDSP data set if its estimated size is less than or equal to the number of KB or 3380 tracks that you specify with the SMALLDATASETPACKING parameter of the SETSYS command. (See "Specifying Small Data Set Packing" on page 36 for an explanation of small data set packing.) A data set's source size in KB in DFSMShsm is calculated as the number of bytes per track for the device type times the number of tracks used by the data set. For each device type in DFSMShsm the number of bytes per track is as follows:

Device Type	Bytes per Track
3380	47,476
3390	56,664

Note: If you specify the size eligibility in tracks, DFSMShsm converts the tracks value to KB with an assumed track size of a 3380 device type.

SDSP Selection

When a user data set meets the criteria to be migrated to an SDSP data set, DFSMShsm uses the first SDSP data set that it finds. DFSMShsm does not try to pick the SDSP data set with the most free space, because the most free space is not necessarily a good criterion for selecting an SDSP data set.

When an SDSP is allocated and open, it remains open until the DFSMShsm-managed volume has been completely processed or the SDSP is needed for recall or aggregate backup.

When a data set fails to migrate because it will not fit into the selected SDSP, DFSMShsm selects a different SDSP when it migrates the next data set to an SDSP. You should consider two error conditions:

- If no SDSP is available to a DFSMShsm host (they have all become full), the data set migrates, but not into an SDSP data set. To avoid this type of migration, define more SDSP data sets than the maximum number of concurrent migration tasks. Because each ML1 volume has no more than one SDSP data set, you must have more ML1 volumes available than the maximum number of migration tasks.
- Do not run automatic secondary space management while you are running automatic primary space management.
- If all SDSP data sets are busy, the migration of the data set fails.

Out-of-Space Condition during Migration

For each data set that migrates to an ML1 volume and not into an SDSP data set, DFSMShsm chooses the ML1 volume with the lowest activity or, as a secondary consideration, the most available space. An out-of-space condition on the ML1 volumes never causes immediate migration from an ML1 volume to an ML2 volume. An out-of-space condition occurs when DFSMShsm:

- Performs migration on a volume and no space exists on any ML1 volume
- Tries to allocate the migration copy and an out-of-space failure occurs

To prevent a large amount of unproductive processing when there is little free space on any ML1, the migration of data sets—caused by an out-of-space failure—retries to the ML1 volume having the most available free space. (The activity to the volume is not considered.) If a second out-of-space condition occurs while targeting the ML1 volume that has the most free space, DFSMShsm does the following:

- If the failing data set is small (less than five tracks), and if this is the first time during this processing that a small data set did not fit on an ML1 volume, the task that encountered the out-of-space condition determines the free space on all ML1 volumes, and the migration of the data set is retried. If the task encounters the out-of-space condition again, the free space determination is not done. In this case, the volume migration is terminated and, if ML1 volumes are the only target devices for the volume migration, migration is held. If tape is being used, that part continues.
- If the data set is not small (greater than or equal to five tracks), DFSMShsm fails the migration for the data set but continues migration with the next data set on the source volume.

The previous processing does not occur if the out-of-space condition is caused by a full VTOC for a volume. In this case, DFSMShsm simply marks the ML1 volume full and tries another.

Catalog Considerations for Migration

DFSMShsm migrates only cataloged data sets. When a data set migrates, the volume serial number of the data set is replaced with MIGRAT in the catalog. DFSMShsm uses the volume serial number of MIGRAT to specify that a data set has migrated. Therefore, an installation must not have any volume with a volume serial number of MIGRAT. If it has, data access failures could result.

DFSMShsm does not recognize JOBCAT and STEPCAT DD statements, so any data sets that require JOBCAT or STEPCAT DD statements appear as uncataloged to DFSMShsm. Because DFSMShsm does not recognize JOBCAT or STEPCAT DD statements, DFSMShsm uses only catalogs available through the standard catalog search. Multilevel alias support is part of the standard catalog search.

Maintenance and Use of the Data Set Date-Last-Referenced Field for Space Management

DFSMShsm initializes the date-last-referenced field of supported data sets in the data set VTOC entry when the volume is first processed for volume space management. When a data set is opened after that, the open routine writes the current date in the date-last-referenced field. DFSMShsm refers to the date-last-referenced field and the PRIMARY DAYS NON-USAGE and LEVEL 1 DAYS NON-USAGE attributes to determine if a data set is eligible for space management. Since the date-last-referenced is set when a data set is opened, a data set that is open for several days can be eligible for migration shortly after it is closed. Allow extra days in the management class parameters for this situation.

If the date-last-referenced field is zero during volume space management, DFSMShsm places the current date into the field. Zeros in the date-last-referenced field mean that volume space management has never processed the data set.

For the majority of ways that DFSMShsm recalls can be initiated, the recall is considered a reference to the data set, so the date-last-referenced field of the data set is set to the current date. As a result, you can preschedule work by recalling data sets that will be required during a later work session. DFSMShsm normally does not automatically migrate these recalled data sets before you can use them.

Certain DFSMShsm activities produce recalls for which the date-last-referenced field is not changed. The date-last-referenced field is not updated during recalls for:

- Certain indirect references causing a recall, such as the MODEL parameter of the IDCAMS DEFINE command, or a catalog access done by CICS® using the ALTER command
- Renaming data sets
- Extent reduction
- Conversion of data sets from one level 0 volume to another level 0 volume

Writing to a Cartridge-Type, Single-File-Format Tape during Migration

During volume migration, DFSMShsm opens the output data set when it processes the first eligible user data set. DFSMShsm writes the migration copy and then updates the control data set records. After current processing has been completed, the tape is already positioned to receive the next migrated data set. If a failure occurs, DFSMShsm marks the cartridge-type tape volume full and selects another tape volume. Migration processing continues.

When the volume migration task has no more volumes to process, the DFSMShsm tape data set is closed and the output tapes are deallocated.

If an I/O error occurs on the source data but does not prevent DFSMShsm from using the cartridge-type tape volume, DFSMShsm adds a data set entry to the TTOC for the failing data set and immediately invalidates the entry.

When end-of-volume is reached while DFSMShsm is writing a 16KB block on a cartridge-type, single-file-format tape volume, DFSMShsm does normal end-of-volume processing.

When individual user data sets migrate to a cartridge-type tape volume, DFSMShsm closes the DFSMShsm tape data set only after all individual data set requests on the migration queue have been processed.

End of Level 0 Volume Processing

At the end of space management of each level 0 volume, DFSMShsm determines the amount of free space remaining on the level 0 volume just processed and the volume's fragmentation. At this time DFSMShsm also passes control to the space management volume exit (ARCMVEXT) if it is turned on.

At the end of processing a level 0 volume, DFSMShsm writes messages ARC0519I and ARC0523I to the migration activity log. The messages record:

- The number of tracks freed by partial release of over-allocated space
- The number of data sets migrated and deleted
- The number of tracks freed by migration
- The time

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

Pass 2 Processing Example

Figure 12 on page 63 shows the results of pass 2 processing. It is assumed that the low thresholds for the volumes have not been reached by pass 1 processing. Compare Figure 10 on page 55 and Figure 12 on page 63.

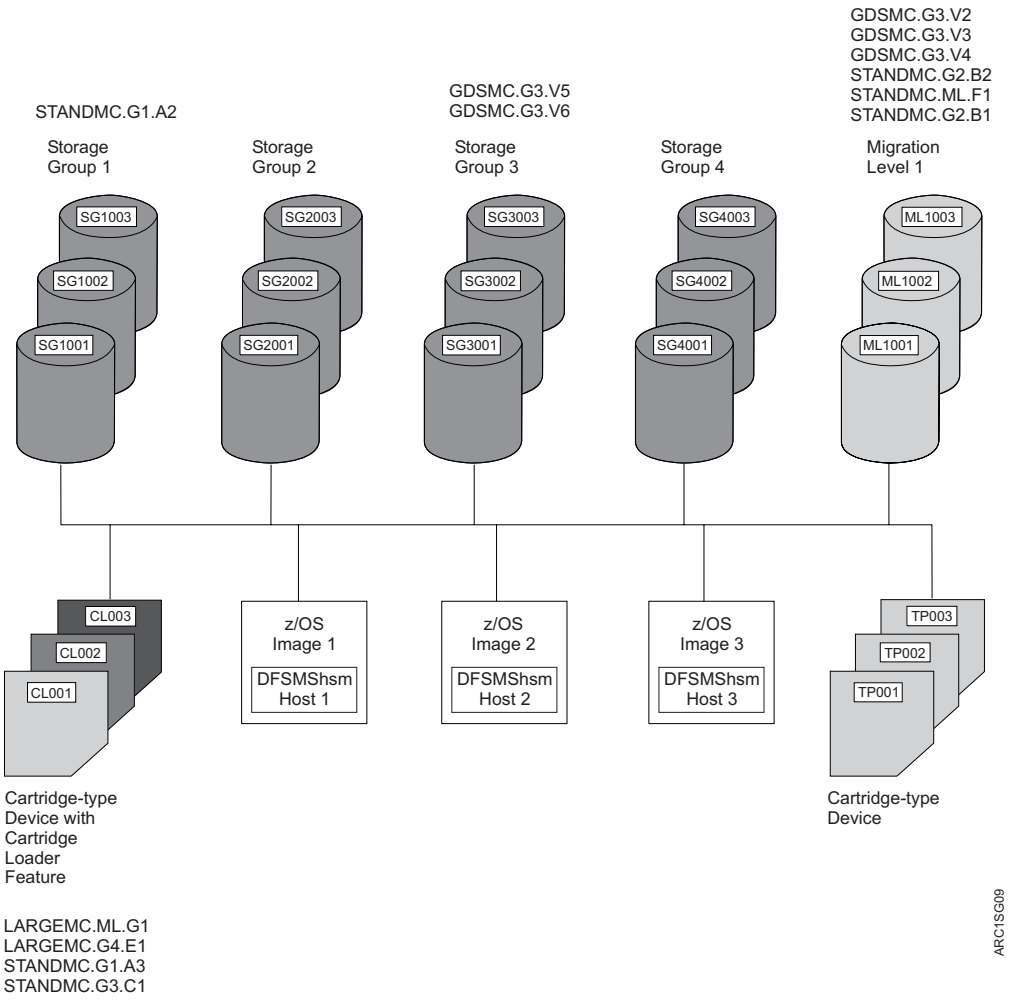


Figure 12. SMS-Managed Volumes after DFSMShsm-Managed Volume Management Pass 2

First look at storage group 1. Because data set STANDMC.G1.A2 has not passed the value for PRIMARY DAYS NON-USAGE (see Table 2 on page 42), it does not migrate.

Now look at storage group 2. Because both STANDMC.G2.B1 and STANDMC.G2.B2 exceed the PRIMARY DAYS NON-USAGE value, both migrate.

Now look at storage group 3. Data sets GDSMC.G3.V3 and GDSMC.G3.V2 migrate first because they exceed the value in the # GDG ELEMENTS ON PRIMARY specification for their management class. Because both are candidates for early migration, either one can be the first one migrated. Next, STANDMC.G3.C1 migrates because its size gives it a priority over GDSMC.G3.V4. Notice that because the unused age of STANDMC.G3.C1 is greater than the value that is specified for LEVEL 1 DAYS NON-USAGE, the data set migrates directly to level 2. Last, GDSMC.G3.V4 migrates because it has exceeded the value for PRIMARY DAYS NON-USAGE.

Finally, look at storage group 4. Data set LARGEMC.G4.E1 migrates because it has passed the PRIMARY DAYS NON-USAGE value. Because the LEVEL 1 DAYS NON-USAGE value for its management class is 0, LARGEMC.G4.E1 migrates to ML2 tape.

You can also compare Figure 12 on page 63 with Figure 7 on page 42 to see how many data sets have been removed or migrated from the DFSMShsm-managed volumes.

Automatic Interval Migration

Hourly throughout the day, except when automatic primary space management is running, DFSMShsm host 2 does interval migration. DFSMShsm hosts 1 and 3 do not do interval migration because the NOINTERVALMIGRATION parameter of the SETSYS command was specified in their ARCCMD*xx* members, except that DFSMShsm hosts 1 and 3 will process volumes in storage group 3 (see page 32). Interval migration can run in one DFSMShsm host while automatic primary space management is running in another DFSMShsm host.

During interval migration, DFSMShsm obtains a list of all volumes in all the storage groups that it processes. It also obtains a list of all the management classes that are active in the SMS configuration except under certain conditions. See "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138 for a discussion of the exceptions.

After receiving SMS information, DFSMShsm checks the occupied space on each volume in the list. For volumes in a storage group with AM=I that have occupied space equal to or exceeding the halfway mark between high and low thresholds, DFSMShsm migrates eligible data sets until the low threshold is reached. When the storage group is AM=Y, migration will only be started after the high threshold is reached. The priority for migrating the data sets is based solely on size. The largest eligible data set migrates first, followed by the next largest. Priority for migrating data sets from volumes in storage groups having AM=I only applies when the specific low threshold exceeds 5%. A data set is eligible for migration if its management class has COMMAND OR AUTO MIGRATION=BOTH and the data set's inactive age has exceeded the value for PRIMARY DAYS NON-USAGE.

During interval migration processing, DFSMShsm does not perform the following functions that are performed during automatic primary space management:
- Partial release
- Extent reduction

If you want the operator to have control of whether interval migration runs or not, specify the REQUEST parameter of the SETSYS command. If you want the operator to be notified of the space occupancy of each volume during the space check, specify the MONITOR(SPACE) parameter of the SETSYS command.

Related reading: For additional information about the REQUEST and MONITOR parameters, see the *z/OS DFSMShsm Storage Administration Reference*.

Standard Interval Migration Compared to Tape Mount Management

Tape Mount Management (TMM) is a methodology that is used to optimize tape subsystem operation and use. It consists of hardware and software facilities, which manage tape data efficiently. By using DFSMShsm in conjunction with your ACS routines to implement the TMM methodology, you can redirect multiple output data sets to DASD, then later have them moved to a single tape, with a single tape mount. This not only improves tape usage, but also greatly reduces the number of tape mounts that are required by your installation.

The following table illustrates the differences between normal interval migration processing and TMM processing:

Table 3. Standard Interval Migration Processing Compared to TMM Processing

DFSMShsm Options / Controls	Interval Migration	TMM with SMS
Hourly option **Note:** Hourly migration is not performed on a host that is currently running Primary Space Management.	Storage group having AM=Y **Note:** Specifying AM=Y for a storage group requests that primary space management and interval migration both be performed on volumes in this storage group. An option of AM=P exists if only primary space management is desired.	Storage group having AM=I **Note:** This option of AM=I is referred to as Tape Mount Management (TMM).
SETSYS INTERVALMIGRATION	Volumes meeting trigger threshold may be processed hourly on this host.	Volumes meeting trigger threshold may be processed hourly on this host.
SETSYS NOTERVALMIGRATION	Volumes are not processed hourly on this host.	Volumes meeting trigger threshold may be processed hourly on this host.
Threshold range	1–99	0–99
Trigger threshold	High threshold	Midpoint between high and low threshold.
Max interval tasks of 0	No hourly migration will be done on this host. **Note:** The option of NOINTERVALMIGRATION provides the same capability for this category of data.	No hourly migration will be done on this host. **Note:** Be aware that if you set the SETSYS MAXINTERVALTASKS parameter to zero, you are preventing hourly migration for all storage groups and non-SMS volumes on this host.
Max interval tasks not 0	'x' tasks available to do hourly migration of volumes meeting trigger threshold.	'x' tasks available to do hourly migrations of volumes meeting trigger threshold.
AMHOST=blank	Hourly migration can be performed on any host permitting interval migration and having interval tasks.	Hourly migration can be performed on any host having interval tasks.
AMHOST=system or system group name **Note:** If you want to allow hourly processing on more than one host but not on all hosts of a sysplex, then you can specify the combination of a sysplex restricting name and also specify SETSYS MAXINTERVALTASKS(0) on the hosts where you do not want to permit hourly migration.	Primary space management and hourly migration limited to this system or group.	Primary space management and hourly migration limited to this system or group.

Automatic Recall

Recall occurs automatically when a data set is referred to by a TSO user, JCL, or a program. The reference to the data set may cause a request that DFSMShsm recall the data set either to a specific volume or to any volume. Unlike non-SMS-managed data, no special considerations apply for SMS-managed data sets between JES2 and JES3 systems. When DFSMShsm receives the request to recall the data set, it passes the following information to the automatic class selection (ACS) routine of DFSMSdfp:

- The name of the data set.
- The name of the storage class associated with the data set when the data set migrated. If the data set was not SMS-managed when it migrated, no storage class is passed.
- The name of the management class associated with the data set when the data set migrated. If the data set used the default management class or was not SMS-managed when it migrated, no management class is passed.
- The name of the data class associated with the data set when the data set migrated. If the data set had no data class or was not SMS-managed when it 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 specified by the request.

  Note: If your storage class ACS routine checks for a valid DASD unit name, be sure to include any unit names that may have existed in your installation at one time and are no longer in use, because the data set could have resided on such a unit when it was migrated.
- 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 indication.
- The data set organization.
- The record organization for VSAM data sets.

If SMS is installed and active, the ACS routine may return to DFSMShsm with a storage class and, optionally, a management class. If the ACS routine returns a storage class, DFSMSdfp selects a volume. The volume selected is not necessarily the volume that was specified by the recall request. The catalog is updated to reflect the current status.

If SMS is installed and active but the ACS routine does not return a storage class, DFSMShsm processes the data set as a non-SMS-managed data set. If the ACS routine does not return a storage class for a data set that was extended format when it was migrated, DFSMShsm fails the recall.

If SMS is not installed and active and the data set was not SMS managed when it migrated, DFSMShsm processes the data set as a non-SMS-managed data set.

If SMS is not installed and active and the data set was SMS managed when it migrated, DFSMShsm fails the recall. If the data set migrated as an SMS-managed data set, DFSMShsm assumes that the data set is still SMS managed and cannot be processed without SMS.

Table 4 summarizes the recall results.

Table 4. Recall Destination Summary

SMS Active	Storage Class Returned	Data Set Migrated As	Data Set Recalled As
Yes	Yes	Either	SMS-managed
Yes	No	Either	Not SMS-managed
No	Not applicable	SMS-managed	Failed

Table 4. Recall Destination Summary (continued)

SMS Active	Storage Class Returned	Data Set Migrated As	Data Set Recalled As
No	Not applicable	Not SMS-managed	Not SMS-managed

Note: For a general discussion of recall requests with regard to the common recall queue, see "Common Recall Queue" on page 109.

Related reading: For additional information about recalling non-SMS-managed data sets, see "Recall" on page 101.

Automatic Recall with DFSMSdss

When DFSMSdss is used as the data mover, DFSMShsm determines if the data set is SMS managed as described under "Automatic Recall" on page 65. However, DFSMShsm allows DFSMSdfp and DFSMSdss to determine which volumes the data set is recalled to.

If a multivolume data set is no longer SMS managed, DFSMShsm attempts to recall the data set. A list of up to five volumes is passed to DFSMSdss. If space permits, the data set is recalled to a single volume. Otherwise, it is recalled as a multivolume data set.

During recall processing with DFSMSdss as the data mover, the &ACSENVIR field contains RECALL for the storage class and management class ACS routines, and RECOVER for the storage group ACS routine. &ACSENVIR is a read-only variable that indicates the environment in which the ACS routine is invoked. The code for the storage group ACS routine should make no distinctions between the RECALL and RECOVER environments.

Recalling Extent-Reduced Data Sets

When a data set migrates for extent reduction, DFSMShsm immediately requests a recall for that data set to the same volume. Under these circumstances, DFSMShsm recalls the data set to the specified volume with the same storage class, management class, and data class as were in effect when the data set migrated. DFSMShsm does not invoke the ACS routine for these data sets.

Automatic Recall during Data Set Allocation

When the allocation process is trying to find a data set, DFSMShsm intercepts the locate and tries to recall the data set if the data set is migrated.

Automatic Recall for Command Deletion of VSAM AIX Clusters

When a command is issued to delete an AIX cluster, the sphere may have been migrated. DFSMShsm intercepts the deletion request, and if the sphere is migrated, DFSMShsm recalls the data set.

Automatic Recall for Data Set Renaming

When a rename or alter request is issued for a migrated data set, DFSMShsm first recalls the data set and then lets the rename or alter process continue, with the exception of an IDCAMS ALTER command issued to change the storage class or management class, or both, of a migrated data set. When an IDCAMS ALTER command is issued to change either the storage or management classes, or both, of a migrated data set, the change is made without recalling the data set. Both DFSMShsm's and catalog's records are updated to reflect the change.

When data sets are recalled for a rename or alter request, DFSMShsm does the following:
- Sets the last backup date to zero
- Sets the data-set-changed indicator in the data set VTOC entry to On
- Sets the date-last-referenced field to the date that the data set had when it migrated

The results of the preceding actions are as follows:
- The renamed data set is backed up the next time that backup runs.
- The renamed data set probably will migrate soon because the date-last-referenced field indicates a long period since the data set was opened.

Automatic Recall during Catalog Updates

A data set, either VSAM or non-VSAM, could be on a migration volume when it is about to have its catalog entry altered. If the catalog routine is requested to UNCATALOG, RECATALOG, DELETE/NOSCRATCH, or ALTER the catalog entry of a data set on a volume with a volume serial of MIGRAT, DFSMShsm first recalls the data set and then lets the catalog routine continue. In this case, DFSMShsm handles the volume selection for the recall request in the same manner as it handles the allocation recall requests. It does not recall the data set to a specific volume.

An IDCAMS ALTER command issued to change the storage class or management class, or both, of a migrated data set is an exception. When an IDCAMS ALTER command is issued to change either the storage or management classes, or both, of a migrated data set, the change is made without recalling the data set. Both DFSMShsm's and catalog's records are updated to reflect the change.

In a non-RACF environment, this function is bypassed, which permits any user to perform catalog maintenance without recalling data sets. In a RACF environment, this function is bypassed for UNCATALOG, RECATALOG, and DELETE/NOSCRATCH if the current RACF logon group is ARCCATGP, which permits only authorized users to perform these requests without recalling data sets.

Note: Automatic recall is bypassed if DFSMShsm receives the DELETE command before any other command. In cases where another component performs an action that invokes DFSMShsm before the DELETE command, an automatic recall occurs. For example, issuing a DELETE command on a migrated data set using some TSO options causes DFSMShsm to recall that data set before processing the DELETE command.

The access methods services REPRO/MERGECAT and CONVERT CATALOG commands do not cause a data set recall, because the catalog entries are not being altered.

A recall may be retried multiple times and lengthen processing time if the necessary resources are not available.

Related reading: For additional information about performing RACF catalog maintenance, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Additional Recall Considerations

When DFSMShsm recalls data sets from tape ML2 volumes, the tape remains allocated and mounted until DFSMShsm has processed all recall requests requiring only that volume, or a TAPERECALLLIMITS parameter has been met. For tape

devices prior to IBM 3480s, the additional requests are processed by file sequence order. For 3480 (single-file-format tape cartridge) and later tape devices, the additional requests are processed by priority. When the recall queue no longer has any recall requests requiring that mounted tape volume, DFSMShsm sets a short timer and waits for another request. If a request needing that single tape arrives, it is started immediately. If the timer expires, DFSMShsm deallocates and demounts the tape volume.

Note: In a common recall queue environment, a single tape mount processes all outstanding requests from all DFSMShsm hosts that require the same tape.

To manage your space better and to reduce job-failure exposures, do not specify volume and unit information in JCL to refer to non-VSAM and VSAM data sets on DFSMShsm-managed volumes. If a non-SMS-managed volume is specified in the JCL, DFSMSdfp and DFSMShsm cannot cooperate in choosing the target volume for the recall. If the data set is non-SMS-managed, the specified non-SMS-managed volume is used as the target of the recall. If the data set is SMS-managed, the recall fails.

Relying on the standard catalog search allows DFSMShsm, DFSMSdfp, and DFSMSdss if specified, to cooperate in choosing the volume and recalling the data set during allocation processing.

The SETSYS VOLCOUNT(NONE | ANY) command affects how the allocation of SMS-managed data sets are performed when DFSMSdss is the datamover. The environment established by the SETSYS VOLCOUNT command can be overridden on command recalls by specifying the DFDSSOPTION(VOLCOUNT(N(*nn*) | ANY)) option of the RECALL command.

Reblocking during Recall

The purpose of reblocking is to make the most efficient use of available space on your DASD. Physical sequential data sets are moved with DFSMSdss. Specifying SETSYS CONVERSION(REBLOCKTOANY) will cause them to be reblocked during recall.

Nonkeyed, blocked, physical sequential data sets that were migrated with DFHSM 2.6.0 (or an earlier release) are also reblocked during recall when SETSYS CONVERSION(REBLOCKTOANY) is used. Partitioned data sets (DSORG=PO) are not reblocked during recall.

Recalling Data Sets with Fast Subsequent Migration

During the recall function, if the recalled data set meets certain fast subsequent migration criteria, DFSMShsm marks the data set as eligible for reconnection to the ML2 tape from which it was recalled. If a data set is not flagged as eligible for reconnection during recall, it will not be considered for reconnection during later migration processing.

Related reading: For additional information about how the recall function marks a data set eligible for reconnection, see "How Recall Affects Fast Subsequent Migration" on page 77.

Additional Space Management during Migration and Recall When DFSMShsm Is the Data Mover

> **Note:**
> Although DFSMSdss is the data mover for most data set types, the following section refers to exception cases for which DFSMShsm is the data mover, including most PDS data sets, as well as special cases of PS, VSAM, and other organizations.

When DFSMShsm is used as the data mover during the migrating and recalling of PS, PDS, and VSAM data sets, it can automatically move a data set to a different device type. Thus, for these data sets, the requested space is always converted by DFSMShsm to a space of approximately equivalent capacity when a device-type change occurs. Also, the original allocation type (that is, the cylinder, track, or average block) and the CONTIG option are maintained.

When DFSMShsm recalls a migrated PS or PDS data set, it requests a primary allocation size approximately equal to the space used (all extents) by the original data set, unless the secondary allocation size is zero. When the secondary allocation size of the original data set is zero, DFSMShsm requests a primary allocation size approximately equal to the space allocated (all extents) at the time of migration. For secondary allocation size, DFSMShsm always keeps approximately that much of the original data set space. The option to release unused space is requested when a nonzero secondary allocation size is specified; release of unused space is not requested when secondary allocation is zero. For PDS data sets, the original number of directory blocks is retained.

For PDS data sets, the migrate and recall results in the freeing of embedded free space (compression), and the user information in the data-set-directory user field is retained.

When the DAOPTION is not used, the allocation for recall of migrated BDAM data sets differs from that of PS and PDS data sets. The primary allocation contains the same number of tracks (including all extents) allocated to the original data set at the time of migration. DFSMShsm maintains the secondary allocation size in a similar manner; that is, the number of tracks is at least as large as that of the secondary allocation size of the original data set. When DFSMShsm recalls a BDAM data set, it does not select a device that has a different track length than that of the device from which the data set was migrated. However, you can direct the recall of a BDAM data set to a device that has a different track length. When DFSMShsm moves BDAM data sets, it moves one input track of data to one output track of data. The data from an original track must fit on a track of the output device. If you direct the recall of a BDAM data set to a particular device, you are responsible for ensuring that the data is accessible in this form on that device. Release of unused space is never requested for BDAM data sets.

The recall of VSAM data sets includes translating space allocation when device types are changed and retaining the type of allocation (cylinder, track, or average block) changes. DFSMShsm does not do any extent consolidation for VSAM data sets. The recall data set is processed in accordance with your original DEFINE request, which can cause the allocated space to grow if you request distributed free space. **Recommendation:** You should have a nonzero secondary allocation size if you request distributed free space.

The migration and recall of non-VSAM data sets normally results in a reduction of the number of extents for multiple-extent data sets, because DFSMShsm's primary allocation size is large enough to contain the entire data set in one extent. If DFSMShsm cannot find such an extent and the CONTIG option is not specified, up to five non-contiguous extents can be used. If DFSMShsm cannot find such an extent and CONTIG is specified, the request fails.

Related reading:

- For additional information about data mover support for DFSMShsm space management functions, see Table 2, "Space Management—Data Mover Support" in Chapter 4, "User Data Sets" in the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about how DFSMSdss moves PS, PDS. and VSAM data sets, see the *z/OS DFSMSdss Storage Administration Guide*.

Command Space Management

In addition to its automatic volume space management functions for SMS-managed volumes, DFSMShsm provides you with the ability, by issuing individual commands, to perform space management functions.

Deleting Migrated Data Sets

To delete migrated SMS-managed data sets, you can use one of the following methods:

- DFSMShsm DELETE command
- Access Method Services DELETE command
- TSO DELETE command
- IEHPROGM utility

Table 5 shows what happens when you specify one of these commands or use the IEHPROGM utility to delete a migrated data set.

Table 5. Commands for Deleting Migrated Data Sets

Command or Utility	Result
DFSMShsm DELETE, Access Method Services DELETE, TSO DELETE or IEHPROGM utility SCRATCH function for SMS-managed data set	Scratches a data set on a DASD migration volume without recalling the data set or marks the data set invalid in the TTOC for any tape ML2 volumes that contain the data set. DFSMShsm then uncatalogs the data set. DFSMShsm deletes all control data set records related to the migration copy. If the data set is migrated to an SDSP, the MCD is updated to indicate that the migration copy needs to be scratched. If a discrete RACF data set profile exists, DFSMShsm deletes it. **Notes:** 1. If the optional parameter **NON-VSAM** is specified on the Access Method Services DELETE command, the migrated data set is recalled before deletion. 2. If you specify the optional parameters **GDG FORCE** on the Access Method Services DELETE command, IDCAMS does not invoke DFSMShsm. You must then issue the DFSMShsm DELETE command for each of the now-uncataloged migration copies.
IEHPROGM utility SCRATCH function for non-SMS-managed data set	Scratches a migrated data set if: • The specified volume is the same one the data set migrated from • The volume is not SMS managed • The data set is not SMS managed DFSMShsm scratches the migration copy of the data set and deletes the MCDS data set record. The data set remains cataloged.

If the deleted data set has backup versions, DFSMShsm keeps the backup versions.

If a data set has been deleted by one of the preceding commands and you want to recover a backup version of the data set, see "Data Set Recovery and Restore" on page 224.

Using the DELETE Command

You can delete migrated data sets under the following conditions:

- For VSAM data sets, make the name specified in the command the name of the base cluster. Specifying the name of any other component causes the command to fail.
- The name specified for the data set (for either VSAM or non-VSAM data sets) must be the name in the catalog. You cannot specify the name that DFSMShsm assigned to the migrated data set.

DFSMShsm fails any DFSMShsm DELETE command that specifies a member name of a partitioned data set.

Deleting Data Sets That Have Not Expired

If the data set has an expiration date and the date has not been 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.

Migrating an Individual Data Set

You can cause individual SMS-managed data sets to migrate by using the MIGRATE or HMIGRATE command. (The HMIGRATE command is a non-DFSMShsm-authorized command that is available to users.) You can migrate a data set by command from either a level 0 volume or an ML1 volume. The migrating data set can move from a level 0 volume to either an ML1 volume or an ML2 volume.

When you issue a MIGRATE (or HMIGRATE) command, DFSMShsm obtains the definition of the management class associated with the data set from SMS. If the management class definition cannot be obtained, DFSMShsm issues an error message and does not migrate the data set. If the data set has no management class associated with it, DFSMShsm uses the default management class.

Data set command migration overrides the management class attributes that control when an SMS-managed data set is eligible to migrate. However, for command data set migration to occur, the COMMAND OR AUTO MIGRATE attribute in the data set's management class must have the value COMMAND or BOTH.

DFSMShsm may perform fast subsequent migration when it migrates an individual data set. If the fast subsequent migration fails, the data set migrates in the normal manner. .

Command migration can occur for either migrated or non-migrated SMS-managed data sets. The data sets can migrate to either ML1 or ML2. Sample commands to migrate data sets are:

```
MIGRATE  DATASETNAME(SAMPLE.DATASET)
MIGRATE  DATASETNAME(SAMPLE.DATASET1)  MIGRATIONLEVEL2
```

Consider the first example. If the data set is on an SMS-managed volume, the LEVEL 1 DAYS NON-USAGE attribute in its management class controls where the data set migrates to. If the value of the LEVEL 1 DAYS NON-USAGE attribute is not zero, the data set migrates to ML1. If the value of the LEVEL 1 DAYS NON-USAGE attribute is zero, the data set migrates to an ML2 tape volume. If the data set has already migrated to an ML1 volume, the data set does not migrate to an ML2 volume because the MIGRATIONLEVEL2 parameter is not specified.

Now consider the second example. If the data set is on an SMS-managed volume or an ML1 volume, it migrates to an ML2 volume no matter what value is specified in the LEVEL 1 DAYS NON-USAGE attribute.

If the target migration device type is tape and the data set needs to be backed up, the data set does not migrate and an error message is issued.

Two installation exits are invoked from the MIGRATE or HMIGRATE commands: the data set migration exit (ARCMDEXT) and the second-level migration data set exit (ARCMMEXT).

DFSMShsm invokes ARCMDEXT whenever a data set command migration request causes a data set to migrate from a Level 0 volume to a DFSMShsm Level 1 or Level 2 volume. In addition, an indicator in the ARCMDEXT parameter list indicates a data set that is migrating by a data set command (rather than by volume migration).

DFSMShsm invokes ARCMMEXT whenever a data set command migration request causes a data set that is already on ML1 to migrate to ML2 (using the ML2 keyword on the data set migration command). In addition, an indicator in the ARCMMEXT parameter list indicates that this data set is migrating by a data set command (rather than by volume migration).

Related reading: For additional information about the fast subsequent migration function, see "Reconnecting Data Sets with Fast Subsequent Migration" on page 76.

Migrating an SMS-Managed Volume

You can cause data sets to migrate from an SMS-managed volume by using the MIGRATE command. The MIGRATE VOLUME command may override some management class attributes that control when an SMS-managed data set is eligible to migrate, depending on the form of the command. The MIGRATE VOLUME command migrates only those data sets for which the management class COMMAND OR AUTO MIGRATE attribute has the value of BOTH.

When you issue a MIGRATE VOLUME command, DFSMShsm obtains a list of the management class definitions required for processing the SMS-managed data sets. If the list cannot be obtained, DFSMShsm issues an error message and fails the command. If the management class definition for a data set cannot be found in the list, DFSMShsm issues an error message and does not process the data set. If a data set does not have a management class associated with it, DFSMShsm uses the default management class.

Fast subsequent migration may be performed when a data set is migrated as a result of a MIGRATE VOLUME command. If the fast subsequent migration fails, DFSMShsm migrates the data set in the normal manner.

When the MIGRATE VOLUME command is issued for an SMS-managed volume, that volume is considered to be DFSMShsm-managed. DFSMShsm will maintain control records for the volume until the DELVOL command is issued against it.

This description is only for SMS-managed volumes; therefore, the volume migration is only from level 0 volumes to ML1 or ML2 volumes. Sample commands to migrate data sets from SMS-managed volumes are:

```
MIGRATE  VOLUME(SG1001)
MIGRATE  VOLUME(SG2003  MIGRATE(0))
MIGRATE  VOLUME(SG1002)  DAYS(0)
```

Consider the first form of the command in the example. For this form of the command, DFSMShsm processes the volume in the same way as is described in "Automatic Primary Space Management" on page 50.

Now consider the second and third forms of the command. These forms of the command override the PRIMARY DAYS NON-USAGE attribute of the management class so that all data sets on the volume that have the COMMAND OR AUTO MIGRATE attribute value of BOTH migrate. Any data set with a LEVEL

1 DAYS NON-USAGE attribute value of 1–9999 or NOLIMIT migrates to an ML1 volume. Any data set with a LEVEL 1 DAYS NON-USAGE attribute value of 0 migrates to an ML2 tape volume. If an attempt to migrate a data set fails, DFSMShsm issues an error message and leaves the data set on the SMS-managed volume. If a nonzero value or no value is specified for the MIGRATE or DAYS parameter, DFSMShsm fails the command and issues an error message.

Related reading: For additional information about the fast subsequent migration function, see "Reconnecting Data Sets with Fast Subsequent Migration" on page 76.

Recalling SMS-Managed Data Sets

You can recall individual SMS-managed data sets from migration volumes to level 0 SMS-managed volumes by using the RECALL and HRECALL commands. (The HRECALL command is a non-DFSMShsm-authorized command that is available to users.)

When you issue a RECALL or HRECALL command for an SMS-managed data set, processing is the same as that for automatic recall (see "Automatic Recall" on page 65).

When a data set is recalled to a non-SMS-managed volume, the volume to which it is recalled is selected as described in "Recall" on page 101.

If you want to return an SMS-managed data set to a non-SMS-managed volume, you can use the FORCENONSMS parameter of the RECALL command. If you use the FORCENONSMS parameter, the volume to which the data set is recalled is selected as described in "Recall" on page 101.

Migrated extended format VSAM data sets can be recalled only as extended format VSAM data sets. Migrated extended format sequential access method (SAM) data sets can be recalled as extended format or nonextended format data sets.

If FORCENONSMS is specified on the RECOVER command to force an extended format data set to be recovered or restored to a non-SMS-managed volume, the data set is recovered as a nonextended data set. The maximum size of the non-SMS SAM data set is one volume.

If an out-of-space condition causes the recall of an SMS-managed data set to fail, the RECALL DFDSSOPTION(VOLCOUNT(N(*nn*) | ANY)) command can be used to ensure the successful recall of the data set. The RECALL DFDSSOPTION(VOLCOUNT(N(*nn*) | ANY)) command overrides the environment established by the SETSYS VOLCOUNT command. You can use the QUERY SETSYS command to determine the current operating environment.

Recalling Direct Access Data Sets

Target volume selection for command recalls of BDAM data sets can be influenced by using the DAOPTION parameter of the RECALL command. This parameter allows you to specify the acceptable target volume device types (in terms of track size) and how the recalled data is accessed (relative track or relative block). For instance, using the RELBLK option of the DAOPTION parameter allows a data set to be recalled to a device type with a smaller track size than the original level 0 volume. In this case, data is moved to "fill out the track" rather than in a track to track image as is the case for the RELATER and SMATTER options.

You can only use the DAOPTION parameter when recalling a data set as non-SMS-managed. You can use the FORCENONSMS parameter to recall a data set that would otherwise be returned as SMS managed.

Reconnecting Data Sets with Fast Subsequent Migration

When a data set that has been migrated to ML2 tape is recalled, the migration copy of the data set remains on the tape. DFSMShsm CDS records recognize that copy as an invalid copy. When the data set that is recalled is remigrated in the normal manner, it is copied to a new tape. Eventually, tapes with invalid data sets are recycled, and only the still-valid data is consolidated onto new tapes.

The fast subsequent migration function allows an unchanged data set that is recalled from a single ML2 tape (for input processing only) to be reconnected to that ML2 tape. This eliminates unnecessary data movement and tape mounts during remigration. It also reduces the need to recycle the tape. Reconnection can take place during individual data set migration or during volume migration.

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

During the migration cleanup function, you can increase the likelihood of a successful reconnection with the SETSYS MIGRATIONCLEANUPDAYS command. If you use this command, you can delay the deletion of the MCD records according to a value that you have specified. DFSMShsm calculates the date that the migration cleanup function deletes MCD records. The basis for the date is the value supplied by the *reconnectdays* parameter. Remember that this ability to keep the MCD records may also cause your MCDS to grow.

In addition, only the SETSYS USERDATASETSERIALIZATION environment supports reconnection. This environment indicates that DFSMShsm is running in a single-host environment or, if there is more than one host, that DFSMShsm is running with global resource serialization (or an equivalent product).

Related reading: For additional information about the *reconnectdays* parameter, see "Specifying How to Do Migration Cleanup" on page 38.

Specifying Fast Subsequent Migration

You can specify whether you want DFSMShsm to reconnect a data set that has been recalled from a single ML2 tape to that same migration copy. Use the RECONNECT (NONE | ALL | ML2DIRECTEDONLY) subparameter of the SETSYS TAPEMIGRATION command to specify reconnection.

The NONE keyword specifies that you do not want to perform reconnection. Data sets that are recalled when this setting is in effect will not be marked as reconnectable. Nor will they be candidates for reconnection during a subsequent migration, even if the setting at the subsequent migration allows reconnection. NONE is the system default.

The ALL keyword specifies that you want all data sets that are eligible for migration to reconnect to the ML2 tape, even if the data set would otherwise be migrated to ML1. However, any data sets that you direct to ML1 with the FORCML1 parameter on the ARCHMIG macro will not be reconnected.

The ML2DIRECTEDONLY keyword specifies that you want reconnection only on data sets that are eligible for direct migration to ML2 tapes. Data sets that are not eligible for direct migration to ML2 tapes migrate normally with no attempt to reconnect.

Attention!
If you specify either the RECONNECT(ALL) or the RECONNECT(ML2DIRECTEDONLY) parameters, ensure that no other product in the installation has reset the change flag in the VTOC data set entry. If it has, DFSMShsm will be unaware that the data set has changed since it was last recalled. When the data set changes, the ML2 migration copy can no longer be used for reconnection.

Other program products include DFSMSdss. Using the RESET keyword—either within the DFSMShsm DUMPCLASS or externally in DFSMSdss batch jobs—can affect DFSMShsm processing.

How Recall Affects Fast Subsequent Migration

During the recall function, DFSMShsm marks a data set as a candidate for reconnection only if it meets all of the following conditions:

- The data set is recalled from ML2 tape.
- The data set is recalled from tape while running in a reconnect environment.
- The data set does not span tapes.
- The SETSYS TAPEMIGRATION(RECONNECT(NONE)) option is not in effect.
- SETSYS USERDATSETSERIALIZATION is the specified option.

In addition to marking the data set as a reconnection candidate, the recall function sets an MCD field that specifies when the data set is predicted to remigrate. The migration cleanup function of secondary space management uses this field. The predicted remigration date is the date of recall plus the number of unreferenced days at the time of the previous migration.

Note: Data sets that are recalled on down-level systems can never be candidates for reconnection. Only data sets recalled on an OS/390® V2R10 (or later) system are eligible for reconnection.

Recalled data sets that have not changed can be reconnected to the ML2 tape from which they were recalled. All data sets types are eligible for reconnection except VSAM data sets that have an AIX or path defined.

Reconnecting Data Sets during Migration

The fast subsequent migration function affects data sets that are recalled and then remigrated. In order for data sets to be eligible for reconnection directly back to ML2 tape, they must meet all standard migration eligibility checks as well as a reconnectability check. DFSMShsm does not attempt reconnection if either of the following is true:

- The MIGRATE command specifies the CONVERT parameter.
- The ARCHMIG macro specifies the FORCML1=YES parameter.

Reconnection Eligibility Criteria: A data set can be reconnected only if it meets all of the following conditions:

- The data set was backed up prior to the original migration to tape.

- DFSMShsm marks the data set as reconnectable at recall.
- The data set has not changed since it was last recalled.
- DFSMShsm has not backed up the data set since it was last recalled.
- The ML2 tape has not been recycled since the data set was last recalled.
- The SETSYS TAPEMIGRATION(RECONNECT) option allows reconnection.
- The SETSYS USERDATASETSERIALIZATION option is in effect.
- The catalog entry for the migrating data set has not changed since it was last recalled.

Note: Although a data set may be an eligible reconnection candidate, that data set may still fail reconnection. For example, another DFSMShsm function could currently be recycling the tape ML2 volume to which reconnection is being attempted. This data set remains eligible for normal migration.

How Fast Subsequent Migration Affects Migration Cleanup

The migration cleanup function may keep MCD records for reconnectable data sets longer than for nonreconnectable data sets. This increases the likelihood that DFSMShsm will reconnect the data set at the next migration.

Note: Run the migration cleanup function on an OS/390 V2R10 (or later) system to ensure that the MCD records are kept for a longer period of time. This will provide a better chance for reconnection

Related reading: For additional information about how to specify migration cleanup, see "Specifying How to Do Migration Cleanup" on page 38.

Using ARCMDEXT to Control Fast Subsequent Migration Eligibility

On an exception basis, you can use the ARCMDEXT installation exit to redirect those migrations that target ML1, so that they now target ML2 tape instead. If you have specified TAPEMIGRATION(RECONNECT(ALL)), you can use this exit to deny reconnection for specific data sets. The fast subsequent migration function provides the ARCMDEXT exit with information in the parameter list as to whether a specific data set is reconnectable.

Related reading: For additional information about the ARCMDEXT exit, see *z/OS DFSMS Installation Exits*.

Chapter 3. Space Management of Non-SMS-Managed Storage

DFSMShsm manages non-SMS-managed storage on a volume basis. Non-SMS-managed volumes that are to be automatically managed by DFSMShsm are known as *primary* volumes. That is, all the data sets on a volume are managed to the same specification, whether for days-not-used on the primary volume, age for deletion, or permission for automatic migration. In addition, DFSMShsm chooses the volume to receive the recalled data sets based on the volume pools you have defined, as well as the characteristics given the volumes when you define them.

Differences between processing for SMS-managed and non-SMS-managed data sets are as follows:

- Two additional management options 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 basis.
- Target volumes for recall are selected by DFSMShsm rather than by DFSMSdfp.
- Volumes are individually identified to DFSMShsm.

Related Reading: For information about using DFSMShsm with Linux, see Appendix C, "Linux-z/OS DFSMShsm Dump or Restore HOW-TO," on page 625.

Differences in What the Example System Shows

The non-SMS-managed-storage space management example performs the same functions as were described for SMS-managed storage:

- Automatic space management
- Interval migration
- Automatic recall
- Command space management

During automatic primary space management, DFSMShsm deletes certain list, utility, and expired data sets. DFSMShsm does not perform partial release for unused allocated space on non-SMS-managed data sets but does perform extent reduction. In addition, the fast subsequent migration function allows unchanged data sets that are recalled from ML2 tape to reconnect to that original tape.

Subtasks for Space Management of Non-SMS-Managed Storage

To manage non-SMS-managed storage, you perform the same subtasks as for managing SMS-managed storage except for the subtasks involved with specifying storage management attributes to SMS. That is, you issue all the same SETSYS, ADDVOL, and DEFINE commands that have been specified so far. You do not define storage groups or management classes. In addition, you perform the following subtasks:

- 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 is defined to DFSMShsm.
- Specify when data sets become eligible for level 1 to level 2 migration.

© Copyright IBM Corp. 1984, 2004

- Specify recall characteristics for primary volumes.
- Define primary volumes to DFSMShsm.
- Define pools of volumes.

Specifying the Age for Deleting List and Utility Data Sets

During automatic space management, DFSMShsm deletes list and utility data sets from primary volumes. DFSMShsm does not migrate utility or list data sets during space management, nor does it back up utility and list data sets during availability management. However, it does process them if they are specified in the MIGRATE DATASETNAME(*dsn*) or BACKDS(*dsn*) commands.

All utility data sets must be at least two days old before DFSMShsm scratches them. The utility data sets that are deleted by DFSMShsm have the following types of data set names:

Data Set Name
HSM.MDS.D*ccc*
mcvtauid.MDB.D*ccc*
SYS*nnnnn*.T*nnnnnn*.?F*ccc*
SYS*nnnnn*.T*nnnnnn*.?V*ccc*
SYS*nnnnn*.T*nnnnnn*.?A*ccc*
N---N.FORMAT.D*nnnnn*.T*nnnnnncc*
N---N.SUBMIT.D*nnnnn*.T*nnnnnncc*
N---N.MERGE.D*nnnnn*.T*nnnnnncc*
N---N.RUN.D*nnnnn*.T*nnnnnncc*
N---N.SUB.D*nnnnn*.T*nnnnnncc*
N---N.R.D*nnnnn*.T*nnnnnncc*
N---N.M.D*nnnnn*.T*nnnnnncc*
N---N.SPRINT.D*nnnnn*.T*nnnnnncc*
N---N.DOSPR.D*nnnnn*.T*nnnnnncc*
*SYSUT
The variables shown in the data set names have the following values: • *n*s is a string of numeric characters whose length is represented by the number of *n*s. This string usually consists of a time when preceded by a T or a date when preceded by a D. • *c*s is an arbitrary length of arbitrary characters for the remaining portion of the data set name. • N---N represents the first qualifier in the data set name. • *mcvtauid* is the DFSMShsm-authorized user identification specified with the UID parameter in the DFSMShsm startup procedure. • *?*s is a string of arbitrary characters whose length is represented by the number of *?*s.

All data set names must comply with the MVS data set naming conventions.

Before a list data set is scratched and uncataloged, it must meet the current DFSMShsm value for the SCRATCHFREQUENCY parameter of the SETSYS command. List data sets that are deleted by DFSMShsm have the following last qualifiers:

- LIST

- LINKLIST
- OUTLIST

In the example system, because only DFSMShsm hosts 1 and 3 do automatic space management for the non-SMS-managed volumes, the following command is added to the ARCCMD*xx* member in DFSMShsm hosts 1 and 3:

```
SETSYS SCRATCHFREQUENCY(5)
```

Specifying the Minimum Age for Migrating Data Sets

The DAYS parameter of the SETSYS command specifies the minimum age for migrating data sets. If you do not specify this parameter, the minimum age is the same as the minimum you can specify for the serialization and processing-unit environment in your system. In the example system, the minimum age is specified as:

```
SETSYS DAYS(7)
```

The value specified by the DAYS parameter of the SETSYS command is used only if no value is specified for the MIGRATE parameter of the ADDVOL command or the MIGRATE parameter of the MIGRATE command. An ADDVOL command with the MIGRATE(0) parameter specified also causes SETSYS DAYS to determine the migration age.

Specifying When Data Sets Become Eligible for Level 1 to Level 2 Migration

For non-SMS-managed storage, the age at which data sets become eligible for migration from level 1 to level 2 is the same for all non-SMS-managed data sets in the system. The MIGRATIONLEVEL1DAYS parameter of the SETSYS command specifies the minimum time that a data set must remain unused on the combination of level 0 and level 1 before it becomes eligible to migrate from level 1 to level 2. For example, for a MIGRATIONLEVEL1DAYS value of 31, a data set might stay unused on level 0 for 31 days, or it could stay on level 0 until it is eligible to migrate to level 1 and reside on level 1 until it was unused for a total of 31 days. In the example system, the following command is added to the ARCCMD*xx* member in DFSMShsm host 2:

```
SETSYS MIGRATIONLEVEL1DAYS(60)
```

(The command is added only in DFSMShsm host 2 because only DFSMShsm host 2 does automatic level 1 to level 2 migration.)

If you want DFSMShsm to do automatic migration from ML1 to ML2 volumes, you must specify the thresholds of occupancy parameter (of the ADDVOL command) for the ML1 volumes.

Specifying Recall Characteristics for the Type of Volume

For non-SMS-managed data sets, DFSMShsm selects the type of volume to which a data set can be recalled. The characteristics for a volume to be eligible to receive recalled data sets are the same for all non-SMS-managed data sets in the system. The RECALL parameter of the SETSYS command specifies the general volume destinations for the recalled data sets. In the example system, the following command is added to the ARCCMD*xx* member in all DFSMShsm hosts:

SETSYS RECALL(ANYSTORAGEVOLUME(LIKE))

This command causes any volume with the use attribute of storage and availability for automatic recall to be eligible for selection as a destination for any recalled data set not directed to a pool volume.

An alternative value for the subparameter of the RECALL command is PRIVATEVOLUME. The PRIVATEVOLUME subparameter specifies that data sets not directed to a pool can be recalled to any volume with the use attribute of public, private, or storage that has AUTORECALL specified in the ADDVOL command.

The LIKE subparameter that was specified causes DFSMShsm to select as a target volume for recall only a volume with recall attributes like those of the volume from which the data set migrated. The alternative UNLIKE subparameter allows DFSMShsm to select a volume with unlike recall attributes.

Related reading:

- For additional information about pool volumes and recall, see "Defining Pools of Volumes" on page 85.
- For additional information about recall attributes, see "Recall Attributes" on page 85.

JES3 Considerations

The RECALL parameter should be consistent among the ARCCMD*xx* members of all the JES3 DFSMShsm hosts. The subparameter to be used is selected in the DFSMShsm host that performs the converter/interpreter locate function.

Defining Primary Volumes to DFSMShsm

If multiple DFSMShsm hosts are requested to process automatic primary space management, only the first DFSMShsm host to process a given volume will perform the function. However, should you find it necessary to limit processing of certain volumes to one DFSMShsm host, you can do so with the following parameters:

- Specifying the AUTOMIGRATION parameter on the ADDVOL command in the DFSMShsm host that you want to perform automatic primary space management on the volume.
- Specifying the NOAUTOMIGRATION parameter on the ADDVOL command in the DFSMShsm hosts that you do not want to perform automatic primary space management on the volume.

At the same time that you make the volume known to DFSMShsm, you specify the following characteristics:

- Whether to perform automatic space management on the volume
- Whether the volume is available for automatic recall of all data sets
- The space management technique for the volume

The ADDVOL command defines non-SMS-managed volumes to DFSMShsm. You cannot use the ADDVOL command for SMS-managed volumes. Non-SMS volumes that are added with the AM parameter on the ADDVOL command work in a similar way to a volume added with an AM=Y storage group. See Table 3 on page 65 for a discussion of standard interval migration processing.

Note: A maximum of eight z/OS images can ADDVOL a primary volume at the same time.

For the example system, the following ADDVOL commands are added to the ARCCMD*xx* member in DFSMShsm hosts 1 and 2:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOMIGRATION -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOMIGRATION -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOMIGRATION -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)
```

The first parameter of the command specifies the volume serial number. Notice that each of the volumes to be managed individually must be added. Primary volumes must be online and permanently resident or reserved.

The second parameter specifies the unit type.

The third parameter specifies that the volume is to be added as a primary volume. Notice that this parameter has subparameters that define whether and how automatic space management is to be performed.

The AUTOMIGRATION subparameter specifies that DFSMShsm is to perform automatic volume space management on the volume. The alternative NOAUTOMIGRATION prevents DFSMShsm from performing automatic volume space management.

The AUTORECALL | NOAUTORECALL subparameter specifies whether the volume is generally available as a recall volume. Notice that AUTORECALL is specified only for volumes GP0001, GP0002, and GP0003. The UG and VP volumes are indicated as NOAUTORECALL because they are members of pools. See the discussion of recall to pools in "Defining Pools of Volumes" on page 85.

The DELETEBYAGE | DELETEIFBACKEDUP | MIGRATE subparameters specify the management technique for the volume. For all volumes in the example system, DFSMShsm migrates the data sets that have not been opened in the number of days specified with the MIGRATE subparameter. You can specify any value equal to or larger than the integrity age up to 999. If you specify the DELETEBYAGE subparameter, DFSMShsm deletes any data set without an expiration date or with an expiration date that has passed, and that has not been opened in the number of days specified with the DELETEBYAGE subparameter. You can specify any value equal to or larger than the integrity age (see "Integrity Age for Space Management" on page 40) up to 999. If you specify the DELETEIFBACKEDUP subparameter, DFSMShsm deletes any data set that has not been opened in the number of days specified with the DELETEIFBACKEDUP subparameter if a current backup version of the data set exists. You can specify any value equal to or larger than the integrity age up to 999.

If you do not specify a value for the number of days with the MIGRATE subparameter, DFSMShsm uses the value specified with the DAYS parameter of the SETSYS command.

The THRESHOLD parameter in these ADDVOL commands operates in the same way as the THRESHOLD attribute in the storage groups. However, for volumes managed by DELETEBYAGE and DELETEIFBACKEDUP techniques, the thresholds are ignored and all eligible data sets on the volume are processed.

Attributes of Primary Volumes

When you use the ADDVOL command to add a primary volume to DFSMShsm, you assign attributes to that primary volume. You can choose the following primary volume attributes:

- AUTOMIGRATION or NOAUTOMIGRATION
- AUTORECALL or NOAUTORECALL
- AUTOBACKUP or NOAUTOBACKUP
- BACKUPDEVICECATEGORY
- AUTODUMP or NOAUTODUMP

The use of the first two primary volume attributes is described in this chapter. The use of the last three attributes is described in Chapter 7, "Availability Management of Non-SMS-Managed Storage," on page 261.

Each DFSMShsm host can decide whether primary volumes are processed by automatic space management, automatic backup, or automatic dump. However, if an automatic process controls a primary volume in any DFSMShsm host, DFSMShsm considers the volume to be automatically controlled for that process. For example, assume you specify the following in DFSMShsm hosts 1 and 2:

```
ADDVOL VOL006 PRIMARY(AUTOMIGRATION) UNIT(3390)
```

Then you specify the following in DFSMShsm host 3:

```
ADDVOL VOL006 PRIMARY(NOAUTOMIGRATION) UNIT(3390)
```

DFSMShsm considers VOL006 to be controlled by automatic space management.

Recall Attributes

The primary volume attributes of AUTOMIGRATION, AUTOBACKUP, and BACKUPDEVICECATEGORY are also designated as recall attributes. That is, DFSMShsm considers the values of these attributes in selecting target volumes for recalling non-SMS-managed data sets. The relationship of these attributes to those of the available volumes is what you specify when you use the LIKE and UNLIKE values with the ANYSTORAGEVOLUME and PRIVATEVOLUME subparameters of the RECALL parameter of the SETSYS command.

Because DFSMShsm hosts 1 and 2 are to do all automatic space management for the non-SMS-managed volumes, you can add the following commands to the ARCCMD*xx* member in DFSMShsm host 3:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)
```

Defining Pools of Volumes

To ensure that recalled non-SMS-managed data sets go to the correct volumes, you can organize your non-SMS-managed volumes into pools. You can define either

pools that accept only specified data set names during recall or pools that accept for recall any data set that migrated from them. In this section, you are first shown how to define the pools, and then the effects of the pools are described.

You use the DEFINE command to define pools of volumes. The DEFINE command associates a pool ID with a list of volume serial numbers. In the example system, the commands added to the ARCCMD*xx* member in all DFSMShsm hosts are:

```
DEFINE POOL(GROUP1 VOLUMES(UG1001 UG1002 UG1003))

DEFINE POOL(GROUP2 VOLUMES(UG2001 UG2002 UG2003))

DEFINE VOLUMEPOOL(SAMPLE VOLUMES(VP0001 VP0002 VP0003))
```

The first two examples specify data set pools. The data set pool list specifies DFSMShsm primary volumes where DFSMShsm is to recall data sets whose set of initial characters match that pool ID. For the first example, data sets with an initial character sequence of GROUP1 are recalled to it. For the second example, data sets with an initial character sequence of GROUP2 are recalled to it.

The third example specifies a volume pool. A volume pool associates volumes from which data sets migrate with other volumes, to aid in the selection of target volumes for recall.

You can specify nonresident volumes in a volume pool, to allow recall of data sets that migrated from volumes that no longer exist in the system. Not all volumes in a volume pool are required to be managed by DFSMShsm. You can specify SMS-managed volumes in volume pools, although they will never be selected from the pool for non-SMS-managed data.

Data sets that migrate from a volume in a volume pool are recalled to an online volume that has been added to the list of volumes managed by DFSMShsm using the ADDVOL command. The target volume selection is based on the volume pool configuration at the time of recall. To change the definition of a volume pool (adding or deleting volumes from the pool), the DEFINE command must include all the volumes you want as part of the pool, including any volumes you selected with a previous DEFINE command if you still want them as part of the pool. The volume pool is limited to 140 volumes.

All volumes in a pool need to have the same geometry. For example, you could have a volume pool that included a 3390 Model 3, and a 3390 Model 9 because they share the same geometry. This means that a data set originally allocated on one model could be recalled to a different model.

A primary volume can be in any number of data set or volume pools and can also be available for general recall use if it is online, permanently resident or reserved, and has the AUTORECALL subparameter in its ADDVOL command.

In a loosely coupled multiple DFSMShsm-host environment:

- A data set or volume pool should 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 should be removed from DFSMShsm by the DELVOL command on *all* DFSMShsm hosts, and from any

ADDVOL commands that may be in the startup member. It may remain in the pool definition. In this way, DFSMShsm can translate a recall request to a currently active volume in the pool.

Pools in Systems Running under JES2

In systems running under JES2, you can define two kinds of pools: data set pools and volume pools. Data set pools and volume pools are known as *user-defined* pools. You can define these pools at any time, either during the startup procedure or after DFSMShsm has been started. Similarly, you can use the ADDVOL command to add volumes to DFSMShsm control at any time.

Any volume that has the AUTORECALL subparameter in its ADDVOL command is eligible to be used for recall of any data set that is not restricted to a pool of volumes. This provides a kind of default pool.

Pools in Systems Running under JES3

In systems running under JES3, three kinds of pools exist. In addition to the user-defined pools that you define as described earlier, DFSMShsm configures a general pool from the unit control block associated with primary volumes and from information obtained from the ADDVOL command. The general pool is a collection of all primary volumes that:

- Are online.
- Are permanently resident or reserved.
- Have a primary volume attribute of AUTORECALL.
- Are mounted with the use attribute of storage or private. (The private volumes are included only if you specify the SETSYS RECALL(PRIVATE) command.)

Volumes in the general pool can also be defined in one or more data set or volume pools. The general pool is the default pool where DFSMShsm recalls migrated data sets if they are not restricted for recall to a user-defined pool.

Specifying Pools in Systems Running under JES3

Because DFSMShsm returns a list of volumes to JES3 at converter/interpreter time, the system pools that are running under JES3 have restrictions on when they can be established. You can specify pool configurations and recall options only at designated times. You must specify the pool configuration command in the ARCCMD*xx* member that is used when you start DFSMShsm with HOSTMODE=MAIN. (A pool configuration that is specified when you are starting HOSTMODE=AUX is ignored.) You cannot change pool configurations until the MAIN DFSMShsm restarts. Pool configurations and recall options must be consistent in all z/OS images that run DFSMShsm with the appropriate HOSTMODE and perform the JES3 locate function.

The ADDVOL, DEFINE, DELVOL, and SETSYS commands determine the configuration of pools of volumes. When DFSMShsm is used with JES3, the DFSMShsm general pool and user-defined pools can be configured only as a result of the commands in the ARCCMD*xx* member processed when DFSMShsm is being initiated. Certain variations of these DFSMShsm commands are accepted only during that initialization period. The one exception is the DELVOL command. The DELVOL command is accepted at any time after initialization by any DFSMShsm that is running in the JES3 image. Therefore, JES3 does not expect a data set on a volume that is unusable. For example, you could use the DELVOL command to delete a primary volume that has I/O errors, so that DFSMShsm does not tell JES3 to use that volume.

To ensure that pools function correctly in a system running under JES3, consider the following conditions:

1. Before you can define pools in each z/OS image, you must specify the following commands for each DFSMShsm host that you are starting in that image:
   - A SETSYS JES3 command. The JES3 parameter of the SETSYS command determines the need for the general pool. For a JES3 system, DFSMShsm builds the general pool when you start HOSTMODE=MAIN. Because information from the ADDVOL command is used to build the general pool, DFSMShsm must know that JES3 is the job entry system before the first ADDVOL command is issued for a primary volume. Once the job entry subsystem is determined, you cannot change it until DFSMShsm is restarted.
   - A SETSYS RECALL command with its appropriate subparameter. The ANYSTORAGEVOLUME and PRIVATEVOLUME subparameters of the RECALL parameter of the SETSYS command are not accepted after the initialization period of DFSMShsm has passed, because the method of recalling data sets determines the volumes that qualify for the general pool.
   - An ADDVOL command for primary volumes that DFSMShsm is to manage.You must specify the PRIMARY parameter of the ADDVOL command for a volume before that volume is included in any pool. Subsequent ADDVOL commands for the same DFSMShsm primary volume must not change the information used by DFSMShsm to define pools. The volume must still:
     – Be online (see the note in item 2)
     – Be permanently resident or reserved
     – Have the same automatic recall parameters specified

     The ADDVOL command can be issued at any time for volumes other than DFSMShsm primary volumes just as in a DFSMShsm environment without JES3.

   Note: Be sure that you include the ADDVOL command specifications in the ARCCMD*xx* member of SYS1.PARMLIB so that DFSMShsm recognizes the volumes at each startup. If ADDVOL commands for ML1 volumes are not in the ARCCMD*xx* member, DFSMShsm does not recognize that they are available unless you issue an ADDVOL command at the terminal for each ML1 volume.
2. The POOL parameter of the DEFINE command can be issued only during initialization of DFSMShsm. DFSMShsm does not accept more than one DEFINE command for the same pool ID during initialization. All DEFINE commands issued for volumes to be included in a data set pool must be preceded by an ADDVOL command for each volume.

   Note: For data set pooling, volumes that are offline at DFSMShsm startup time are added to DFSMShsm as offline and are not added to the data set pool. When these volumes become available, DFSMShsm must be restarted to add the volumes as online and include them in the data set pool.
3. The VOLUMEPOOL parameter of the DEFINE command can be issued only during initialization of DFSMShsm. DFSMShsm will not accept more than one DEFINE command for the same pool ID during initialization. At least one volume in the volume pool must have been added with a previous ADDVOL command to the list of volumes DFSMShsm manages or owns.

4. You can specify other parameters of the DEFINE command as if in a DFSMShsm environment without JES3.
5. Be sure that you define the same data set and volume pools in all z/OS images that can process the volumes.
6. To ensure that recall is to a volume that you choose, do not refer to any data set eligible for space management before the appropriate pool configuration in that z/OS image has been defined to DFSMShsm.
7. Specify each DFSMShsm primary volume with the same parameters (for a given volume) in each z/OS image. The mount status of permanently resident or reserved prevents DFSMShsm primary volumes from being demounted. One of these attributes must be in effect for each DFSMShsm primary volume.
8. When a local z/OS image assumes the function of the global image, any required DFSMShsm functions that are performed by the previous global image must be active on the new global image. Specify the same pool configuration on each image to protect this type of JES3 activity.
9. Each JES3 z/OS image can have unique user-defined pools. However, if your system uses unique user-defined pools whose volumes are not accessible to the global z/OS image, you must use job classes or JES3 JCL to direct all jobs that access the unique user-defined pools to the z/OS image that can access them. This is a requirement when both the global and the local z/OS images request JES3 locates. If any z/OS image has access to a primary volume in a data set pool, it must have access to all primary volumes in that data set pool, or the job could fail. If you decide not to give all JES3 z/OS images access to all primary volumes, use caution when you define and operate with unique data set pool configurations.
10. If any device containing a volume in a data set pool is designated as a device managed by JES3, all other volumes in that data set pool must reside on JES3-managed devices. JES3 schedules jobs only on z/OS images that have access to all volumes that the jobs require.
11. Any z/OS image that runs the job must have access to all the volumes passed to JES3 by DFSMShsm.

Example of JES3 Pooling

Figure 13 on page 90 shows three z/OS images in a JES3 and DFSMShsm environment. The following describes some pertinent facts about the figure:

- Volumes A, B, C, D, F, G, X, and Y make up the primary volume list. All the z/OS images have access to the online storage volumes X and Y. However, because only X and Y have the AUTORECALL primary volume attribute, non-pooled data sets are recalled only to volumes X and Y.
- Local z/OS images 1 and 3 have access to volumes A and B.
- Volumes A and B are both in data set pools USER1 and USER3. Jobs that access these volumes can be run only by z/OS image 1 and z/OS image 3, even though z/OS image 2 can set up the jobs.
- z/OS image 1 has access to volume Q. DFSMShsm does not manage volume Q.
- z/OS image 3 has access to volumes F and G, which make up data set pool USER2.
- Volumes C and D are in a volume pool. If a data set migrated from either volume C or D with a high-level qualifier other than USER1, USER2, or USER3, the migrated data set is recalled to either volume C or D.

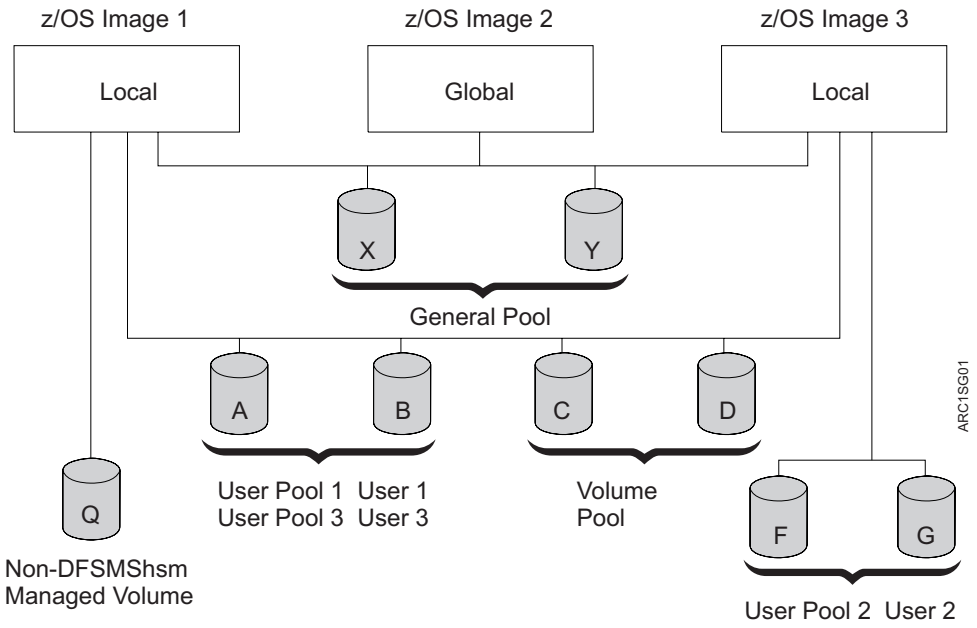


Figure 13. Multiple-Processing Units in a JES3 and DFSMShsm Environment

You must issue these commands on all DFSMShsm hosts on all three z/OS images:

```
SETSYS JES3   RECALL(ANYSTORAGEVOLUME)

ADDVOL   X   UNIT(3390)    PRIMARY(AUTORECALL)
ADDVOL   Y   UNIT(3390)    PRIMARY(AUTORECALL)
ADDVOL   A   UNIT(3390)    PRIMARY(NOAUTORECALL)
ADDVOL   B   UNIT(3390)    PRIMARY(NOAUTORECALL)
ADDVOL   C   UNIT(3390)    PRIMARY(NOAUTORECALL)
ADDVOL   D   UNIT(3390)    PRIMARY(NOAUTORECALL)
ADDVOL   F   UNIT(3350)    PRIMARY(NOAUTORECALL)
ADDVOL   G   UNIT(3350)    PRIMARY(NOAUTORECALL)
DEFINE   POOL(USER1   VOLUMES(A  B))
DEFINE   POOL(USER3   VOLUMES(A  B))
DEFINE   POOL(USER2   VOLUMES(F  G))
DEFINE   VOLUMEPOOL(SHARED  VOLUMES(C  D))
```

Any of the three z/OS images in Figure 13 can successfully run a JES3 locate routine if the appropriate catalogs are accessible.

Automatic Space Management Processing for Non-SMS-Managed Storage

Processing of non-SMS-managed storage differs from that of SMS-managed storage.

In the example system, DFSMShsm performs space management on the system configuration shown in Figure 14 on page 91. Remember, however, that the system for which the example system is built is that shown in Figure 4 on page 17 and that Figure 14 is merely a subset that is used for simplification. When space management starts (first on DFSMShsm host 1 and later on DFSMShsm hosts 2 and 3), the space management is being done for all of the systems shown in Figure 4. Table 6 shows the characteristics for each data set in the various groups of

volumes shown in Figure 14. The names of the data sets are not precisely as the system would generate them. They have been shortened and modified for convenience in using them as examples.

Table 6. Data Set Characteristics for Automatic Space Management Example (Non-SMS-Managed Volumes)

Data Set Name	Inactive Age (Days)	Size (KB)	GDG Status
GROUP1.SAMPLE.A1	16	217	None
GROUP1.SAM.LIST	6	21	None
GROUP2.SAMPLE.B1	5	210	None
U1234.G3.G4	20	20	None
U1234.G3.G5	16	605	None
U2345.G3.G6	61	350	None
LARGE.ML.C1	16	1100	None

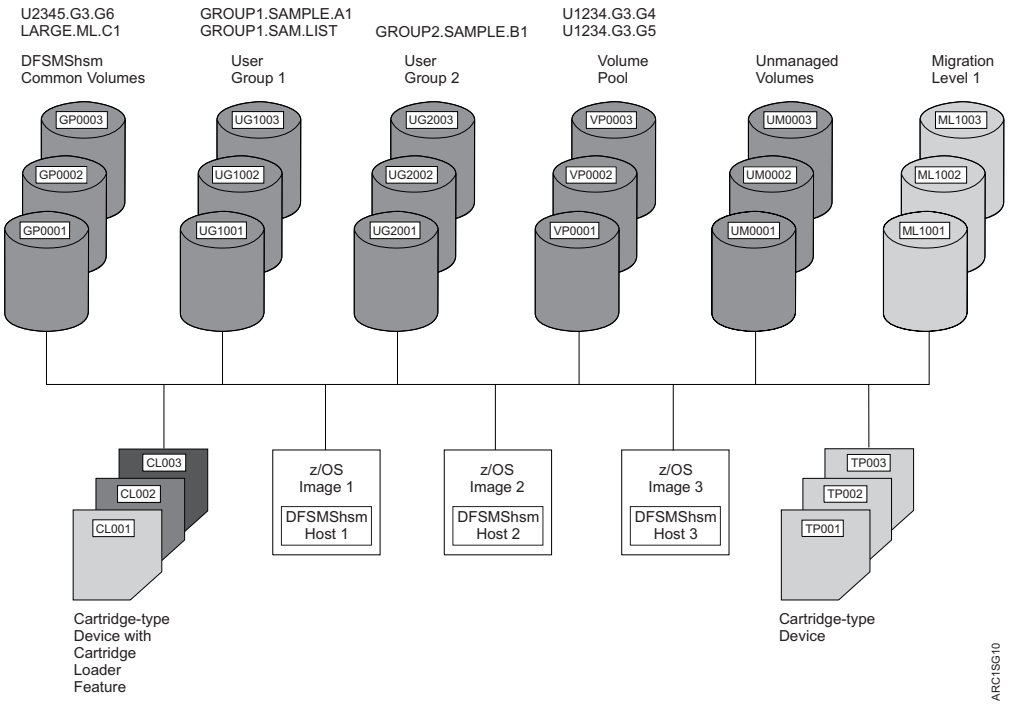


Figure 14. Example System for Non-SMS-Managed Volumes before Automatic Space Management

Automatic Secondary Space Management

Although migration cleanup and level 1 to level 2 migration are performed on non-SMS-managed data sets, they operate the same whether the system is processing SMS-managed or non-SMS-managed storage. In the example system, they are performed by DFSMShsm host 2.

During migration cleanup, DFSMShsm uses the EXPIREDDATASETS(SCRATCH) parameter of the SETSYS command to determine whether to delete expired non-SMS-managed data sets. It uses the *statdays* and *recalldays* subparameters of the MIGRATIONCLEANUPDAYS parameter of the SETSYS command to determine

when to delete statistics records and MCD records for recalled data sets respectively. The use of these parameters is discussed in "Migration Cleanup" on page 44.

During level 1 to level 2 migration of non-SMS-managed data sets, DFSMShsm uses the MIGRATIONLEVEL1DAYS parameter of the SETSYS command to determine when to migrate data sets from migration level 1 to migration level 2.

For performance reasons, level 1 to level 2 migration is not performed during automatic secondary space management in a non-SMS and direct-to-tape migration environment. DFSMShsm assumes there are no migrated data sets on any ML1 volumes that need to be moved to ML2 volumes in this environment. If, for any reason, there are migrated data sets residing on any ML1 volumes that are to be moved to ML2 volumes in a non-SMS and direct-to-tape migration environment, the MIGRATE ML1 command should be used.

Related reading: For additional information about the use of the MIGRATIONLEVEL1DAYS parameter, see "Specifying When Data Sets Become Eligible for Level 1 to Level 2 Migration" on page 81.

Automatic Primary Space Management

During automatic primary space management, DFSMShsm does space management on each primary volume with the AUTOMIGRATION primary volume attribute. The space management technique that you specify with the ADDVOL command determines the kind of space management that DFSMShsm does: data set deletion, data set retirement, or migration.

Automatic primary space management for non-SMS-managed volumes is performed in two passes.

Pass 1

During pass 1, DFSMShsm examines every data set on the volume (independent of the occupancy of the volume), and checks the integrity age to ensure that it can process the data set. The inactive age must be more than or equal to the integrity age. The inactive age is the difference between the date of processing (current date) and the value in the date-last-referenced field of the data set VTOC entry. (For a discussion of integrity age, see "Integrity Age for Space Management" on page 40.) However, for non-SMS-managed data sets, DFSMShsm does not perform the following actions:

- 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, DFSMShsm does perform the following actions:

- Delete list and utility data sets
- Delete data sets that have passed their expiration dates
- Determine eligible data sets for migration, including fast subsequent migration
- Determine eligible data sets to be backed up
- Determine eligible data sets for extent reduction

For each data set on the volume, DFSMShsm uses the parameters specified in the SETSYS and ADDVOL commands in that DFSMShsm host.

Programming Interface Information

As DFSMShsm considers each data set for eligibility for processing, it passes control to the space management and backup installation exit (ARCSAEXT) if the installation exit is enabled. (Refer to *z/OS DFSMS Installation Exits* manual for a discussion of ARCSAEXT.) If the installation exit indicates that processing is to continue (or the installation exit is not enabled), DFSMShsm processes the data set. If the installation exit indicates that processing is not to continue, DFSMShsm begins processing the next data set.

End of Programming Interface Information

Deleting List and Utility Data Sets: As DFSMShsm processes each data set, it deletes those utility data sets that are more than two days old and whose names follow one of the formats listed under "Specifying the Age for Deleting List and Utility Data Sets" on page 80. It also deletes those list data sets having names with the final qualifier as specified under Specifying the Age for Deleting List and Utility Data Sets, and older than the age specified with the SCRATCHFREQUENCY parameter of the SETSYS command.

Deleting Expired Data Sets: For non-SMS-managed data sets, DFSMShsm deletes those data sets that have passed their expiration date. For non-VSAM data sets this would be the expiration date in the data sets' VTOC entries and for VSAM data sets this would be the expiration date in the data sets' catalog entries. For these data sets, the EXPIREDDATASETS parameter of the SETSYS command controls whether the data sets are deleted.

Note: For multivolume non-VSAM data sets, DFSMShsm looks at the VTOC of the first volume.

Determining Eligible Data Sets for Migration: For non-SMS-managed data sets, the criteria for whether a data set is eligible for migration include the inactive age of the data set compared to the *days* value specified with the migration technique parameter of the ADDVOL command for the volume on which the data set resides. If the inactive age is larger than or equal to the *days* value, the data set becomes eligible for migration. If no *days* value has been specified on the ADDVOL command or the *days* value is equal to zero, DFSMShsm uses the value specified with the DAYS parameter of the SETSYS command. The *days* value, whether taken from the ADDVOL command or the DAYS parameter of the SETSYS command, is called the minimum migration age.

Related reading:

- For additional information about how to determine the inactive age, see "Pass 1" on page 92.
- For additional information about the reconnection function, see "Reconnecting Data Sets with Fast Subsequent Migration" on page 76.

Determining Eligible Data Sets for Backup: DFSMShsm must determine which data sets are eligible for backup so that it can schedule the data sets for backup if they migrate or prevent migration if the migration target volume is a tape volume. A non-SMS-managed data set is eligible for backup if all of the following apply:

- The data-set-changed indicator in the data set VTOC entry indicates that the data set has been changed.
- The volume on which the data set resides has the AUTOBACKUP attribute in *any* DFSMShsm host in the system.

- The data set is supported for automatic backup, as indicated in the table called "Availability Management—Data Mover Support" in Chapter 3 of the *z/OS DFSMShsm Implementation and Customization Guide*.

Determining Eligible Data Sets for Extent Reduction: The criteria for whether a data set is eligible for extent reduction are the same as for SMS-managed data sets.

Pass 1 Processing Example: Figure 15 shows the results of pass 1 processing for non-SMS-managed volumes in the example. Data set GROUP1.SAM.LIST is deleted because it is a list data set that is older than the SCRATCHFREQUENCY value set with the SETSYS command.

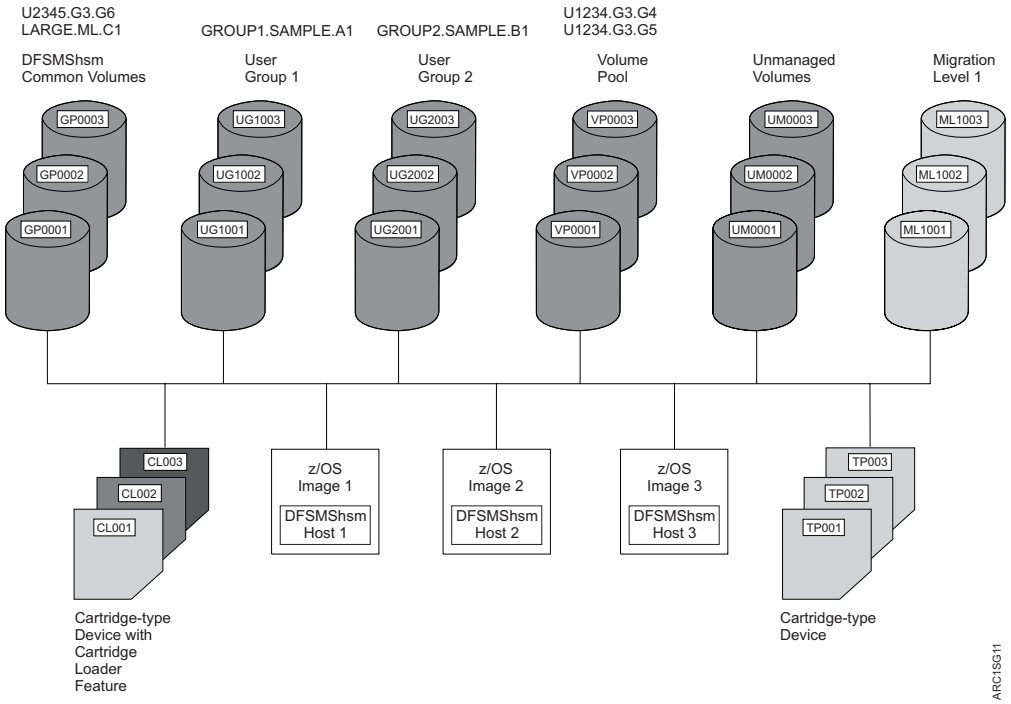


Figure 15. Example System for Non-SMS-Managed Volumes after Primary Volume Management Pass 1

Pass 2

Although the functions performed in pass 2 of primary volume space management for non-SMS-managed volumes are similar to those for phase 2 processing of SMS-managed volumes, the manner of processing is considerably different. Pass 2 processing does the following tasks:

- Sorts the candidates for migration into priority order
- Performs space management on the volume

Sorting the Candidate Data Sets into Priority Order: For volumes that have the space management technique of migration, DFSMShsm sorts the names of the data sets into migration priority order, high priority to low priority. No consideration is given to generation data sets; all data sets on the volume receive their priority on the basis of age and size. Note, however, that the age used for the computation is the inactive age as opposed to the eligibility age used for SMS-managed data sets.

Performing Space Management on the Volume: During primary space management for non-SMS-managed volumes, DFSMShsm processes the volume according to the one technique specified by the ADDVOL command for the volume. Thus, all the eligible data sets on the volume are managed to the same inactive age and by the same management technique. The available techniques are:

- Delete by age (deletion)
- Delete if backed up (retirement)
- Migration (with or without thresholds of occupancy)

Data Set Deletion: For those primary volumes with the DELETEBYAGE space management technique, DFSMShsm scratches low-activity cataloged and uncataloged data sets based on the inactive age of the data sets. You specify the inactive age when you specify the space management technique. DFSMShsm scratches only data sets whose expiration dates have been reached or data sets without expiration dates. For data set deletion, DFSMShsm ignores any thresholds you specify and scratches all eligible data sets.

As DFSMShsm examines each data set for deletion, it passes control to the data set deletion installation exit (ARCADEXT).

Related reading: For additional information about the ARCADEXT installation exit, see *z/OS DFSMS Installation Exits*.

Data Set Retirement: For those primary volumes with the DELETEIFBACKEDUP space management technique, DFSMShsm checks the backup control data set (BCDS) to be sure that a current backup version of the data set exists before DFSMShsm scratches the data set based on its inactive age. You specify the inactive age when you specify the space management technique. DFSMShsm scratches a data set even if the expiration date has not been reached.

DFSMShsm creates backup versions when automatic or command backup runs. If a backup version is made on the date the data set is last opened, DFSMShsm does not consider the backup version to be current. This means that DFSMShsm must make a backup version of a cataloged data set on the calendar day *after* the last activity of the data set. For non-VSAM data sets, the backup must be made after the last-referenced day. For VSAM data sets, the backup must be made after the last-update day. Therefore, if you want to use data set retirement often, schedule backup to run soon after midnight.

By running backup soon after midnight, you reduce your exposure to having the last backup made on the same date that the data set was last changed. If the data set is changed before midnight, automatic backup only occurs on the next calendar date, so the dates would not be the same. There are two possible scenarios where exposures might occur:

- Data sets that were changed and then backed up by command on the same day
- Data sets that were changed between midnight and the time that automatic backup encountered them in the course of its processing

Table 7 shows what DFSMShsm considers to be the current backup version of migrated and unmigrated data sets.

Table 7. Current Backup Version of Data Sets

Type of Data Set	Current Backup Version
Not migrated VSAM	DFSMShsm made the backup version since the last day someone updated the data set.
Not migrated, cataloged, non-VSAM	The update flag is off and the date-last-referenced that existed when the data set was backed up is earlier than the date the data set was last backed up.
Uncataloged, non-VSAM	The update flag is off and DFSMShsm made a backup version of the data set from the volume being processed.
Migrated VSAM	The update flag is off and DFSMShsm created the backup version after the date DFSMShsm last migrated the data set.
Migrated non-VSAM	The update flag is off and the date-last-referenced that existed when the data set was migrated is earlier than the date the data set was last backed up.

During data set retirement, DFSMShsm marks the latest backup version as a retirement version and gives it special protection against inadvertent deletion.

As DFSMShsm examines each data set for retirement, it passes control to the data set deletion installation exit (ARCADEXT).

Migration: For those primary volumes with the MIGRATE space management technique defined on the ADDVOL command, DFSMShsm uses the minimum migration age or the low threshold of occupancy specified for the volume to determine how many data sets are to be migrated. You specify the minimum migration age when you specify the space management technique. During primary volume space management, DFSMShsm uses the priority order determined in the earlier step if thresholds have been specified for the volume. If no thresholds have been specified, DFSMShsm just migrates each candidate data set that it finds on the volume. The end of DFSMShsm's migration of data sets depends on whether you have specified thresholds of occupancy for the primary volume.

As DFSMShsm examines each data set for migration, it passes control to the data set migration installation exit (ARCMDEXT).

With Thresholds of Occupancy: Data sets migrate in priority order from the primary volume until the low threshold of occupancy is reached or until the age of the oldest remaining data set is less than the minimum migration age.

Example: If a primary volume has a low threshold of occupancy of 80% and a minimum migration age of four days, DFSMShsm migrates data sets until at least 20% of the volume is available or until all data sets not referred to within the last three days have migrated.

Without Thresholds of Occupancy: DFSMShsm uses the minimum migration age of the data sets in selecting data sets to migrate from the volume.

Example: If the minimum migration age is seven days, data sets migrate from the primary volume until DFSMShsm finds that all remaining data sets have been referred to within the last six days. The minimum migration age is limited by the integrity age.

Related reading

- For additional information about the affect of using ARCMDEXT for data set retirement or migration, see *z/OS DFSMS Installation Exits*.
- For additional information about the data set integrity age, see "Integrity Age for Space Management" on page 40.

Space Management of VSAM Data Sets

DFSMShsm migration functions that exist for non-VSAM data sets also exist for VSAM data sets cataloged in the integrated catalog facility catalog.

The migration of a non-SMS-managed VSAM data set is performed only on a complete VSAM sphere. The base cluster and all alternate index clusters defined on the base cluster are migrated as part of the data set and are not specified separately. Paths are included automatically in the data set command migration.

When DFSMShsm migrates a VSAM data set, it tracks a maximum of eight component names for that data set. Figure 16 shows component name limitations when DFSMShsm is recalling a migrated VSAM data set.

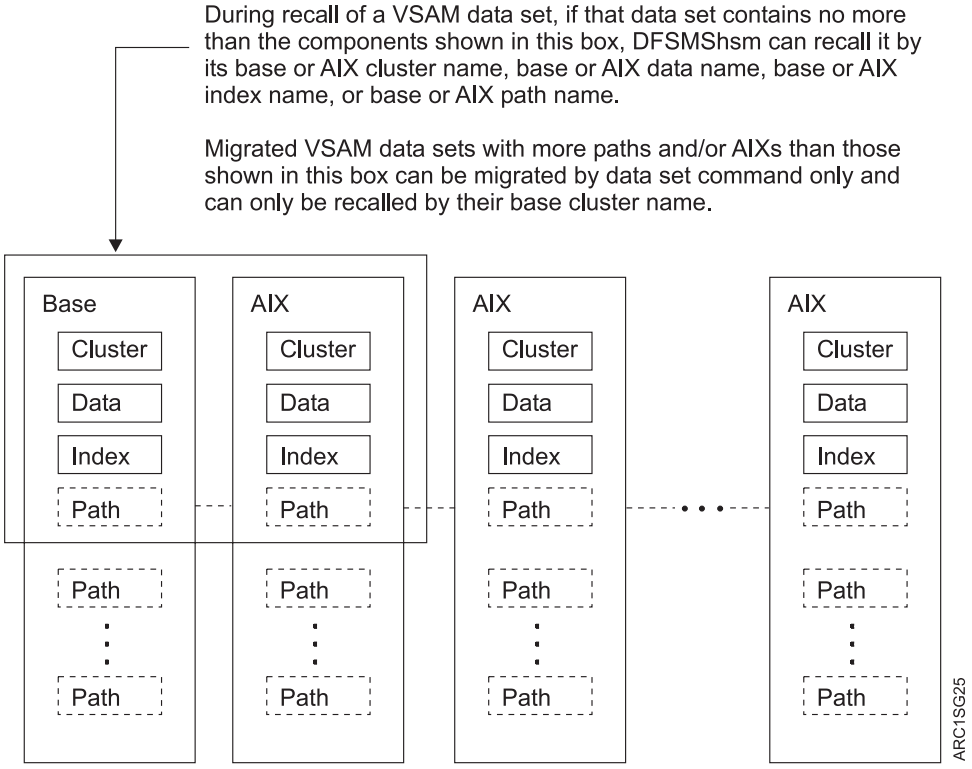


Figure 16. Component Name Limitations When Recalling Migrated VSAM Data Sets

A non-SMS-managed VSAM data set can be migrated only if it is completely contained on one volume. DFSMShsm does not migrate a sphere that has a component that needs to be verified.

Data sets cataloged in the integrated catalog facility catalog with more than one alternate index (AIX), more than one path on the base, or more than one path defined on the AIX are not selected for migration during volume processing. These data sets can be migrated by data set command migration.

DFSMShsm does not migrate a VSAM data set marked for forward recovery, or one with retained locks.

Related reading: For additional information about the data sets not supported by space management, see the *z/OS DFSMShsm Implementation and Customization Guide*.

JES3 Considerations

For the recall of SMS-managed data sets, JES2 and JES3 appear to be the same. However, for non-SMS-managed data sets, the following differences can occur when DFSMShsm is used on a single DFSMShsm-host or a multiple DFSMShsm-host configuration in a JES3 environment using the benefits of JES3 data set reservation as provided by JES3 setup.

- Volumes are considered in terms of data set pools, volume pools, and general pools. The general pool, configured by DFSMShsm, is the default pool that contains primary volumes that are online and permanently resident or reserved and that have the automatic recall attribute. From this set, volumes are selected for undirected recall.
- For migrated data sets, JES3 associates the requested data set with a subset of five or fewer candidate pool volumes supplied by DFSMShsm. JES3 treats the data set as a multiple-volume data set.
- For data sets that have not migrated, JES3 associates the requested data set to the serial number of the primary volume on which the data set resides. DFSMShsm prevents migration of the data set for today and three more days.
- DFSMShsm recalls the migrated data set to a specific volume serial number in the candidate volume list when the data set is opened or to the best candidate primary volume in the subset of five or fewer pool volumes when the data set is allocated.

This discussion is concerned only with the function of delaying migration. Associating migrated data sets with volumes for recall is described in "Recall" on page 101, a discussion of automatic recall for non-SMS-managed data sets.

During the post-scan phase, JES3 calls its locate module to locate the cataloged data sets.When the JES3 locate routine finds a primary volume serial number for a requested data set in the computing system catalog, JES3 associates the specified data set with that volume serial number. Asynchronously with the JES3 locate routine, DFSMShsm passes a management work element (MWE) to prevent migration of the requested data set for up to three days (default value) by recording the primary volume serial number and a prevent-migration date in the MCDS data set (MCD) record. The prevent-migration date is the last day that migration is prevented. If the job does not run within a prevent-migration period, the prevent-migration date expires, and the data set can migrate automatically or by command.

A JES3/DFSMShsm migration window exists between the asynchronous processing of the JES3 locate routine and the DFSMShsm management work element. If migration of the data set occurs in the JES3/DFSMShsm migration window, DFSMShsm modifies the MCD record so that the recall of the migrated data set occurs to that primary volume serial number that was returned to JES3. If you enter a command to migrate a data set associated with a volume that has a current prevent-migration date, DFSMShsm fails the command. A message is issued to indicate that JES3 has an outstanding request for the data set, and migration is prevented until the prevent-migration date has expired.

The prevent-migration period is a default value of 3 in the management communication vector table (MCVT). The default value is added to the current date to create the prevent-migration date and the hold-volumes date in the MCD

record. You can change this value. Although the value can range from 0–255, the value should be consistent with the maximum time a job can exist in JES3 before it is run.

JES3 directed-recall and prevent-migration requests are processed as DFSMShsm commands. In a multiple DFSMShsm-host environment, these requests may not be processed until the completion of a command that is currently being processed. Normally, the delay has little effect on the time required to process the JES3 request. However, some performance degradation may be noticed if the DFSMShsm command processor is waiting for the availability of a resource.

Related reading: For additional information about using the PATCH command for the prevent-migration period, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Performing Extent Reduction on the Volume

After DFSMShsm performs migration for the non-SMS-managed volumes, it performs extent reduction. Extent reduction for non-SMS-managed data sets is performed in the same way as for SMS-managed data sets.

Note: DFSMShsm does not do extent reduction if you have specified SETSYS TAPEMIGRATION(DIRECT) and you have not specified an ML1 volume. The extent-reduction process would cause contention between migration and recall for the target tape ML2 volume.

Installation Exits during Automatic Primary Space Management

Programming Interface Information

Control is passed to installation exits during automatic primary space management for non-SMS-managed data sets in the same way as for SMS-managed data sets. Delete by age and delete if backed up are not applicable to SMS-managed data sets; therefore, ARCADEXT is not called in processing SMS-managed data sets.

Pass 2 Processing Example: Figure 17 on page 100 shows the results of automatic primary space management after the second pass. Data set GROUP1.SAMPLE.A1 migrated because its inactive age is greater than the minimum migration age specified on the ADDVOL command. Because it is small enough that compaction will probably reduce it to less than 110KB (1KB = 1024 bytes), it migrated to an SDSP data set. Data set GROUP2.SAMPLE.B1 did not migrate because it has not passed the minimum migration age.

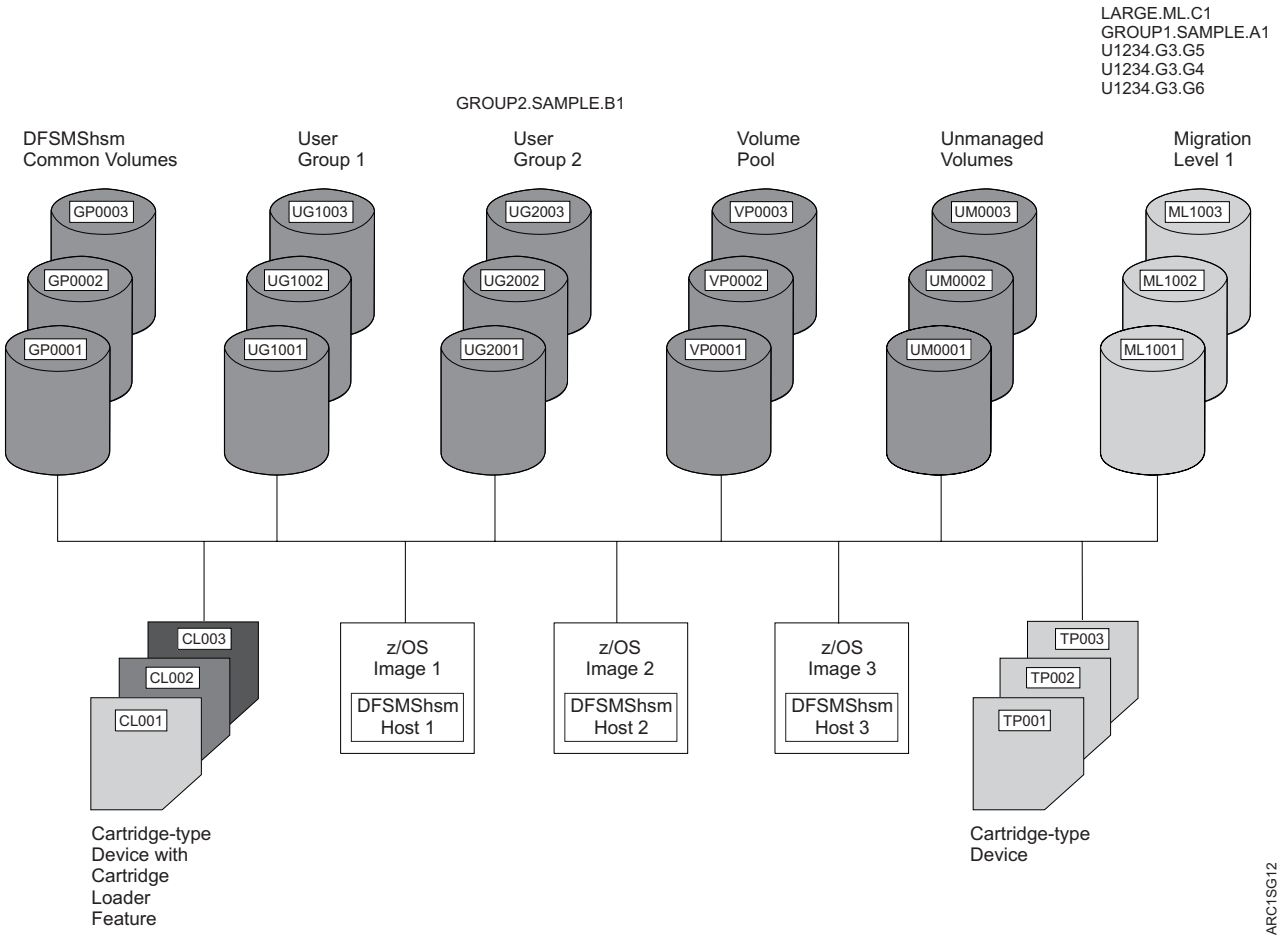


Figure 17. Example System for Non-SMS-Managed Volumes after Primary Volume Management Pass 2

Data sets U1234.G3.G4 and U1234.G3.G5 migrated because of their inactive ages. Assuming that both were on the same volume, U1234.G3.G5 migrated first because of the priority algorithm.

Data sets U2345.G3.G6 and LARGE.ML.C1 migrated because of their inactive ages. Observe that although U2345.G3.G6 has exceeded the MIGRATIONLEVEL1DAYS parameter specified in the SETSYS command, it migrated to ML1. If it is not recalled, it migrates to ML2 the next time that level 1 to level 2 migration is performed.

Observe also that LARGE.ML.C1 is of a size that would put it in management class LARGEMC so that it would migrate directly to tape if it were SMS-managed. However, the example leaves it migrating to ML1 DASD because direct-to-tape migration for non-SMS-managed data sets has not been provided. Its migration could be prevented, or it could be directed to tape by using the migration installation exit (ARCMDEXT) that is called for each data set that is migrated.

End of Programming Interface Information

Related reading: For additional information about the ARCMDEXT installation exit, see *z/OS DFSMS Installation Exits*.

Automatic Interval Migration

For non-SMS-managed volumes, interval migration occurs only from primary volumes that possess the following conditions:

- Have defined thresholds of occupancy
- Have exceeded their high threshold of occupancy
- Have the primary volume attribute of automatic migration
- Have been added for automatic migration in a DFSMShsm host that has interval migration specified
- Have the space management technique of migration

DFSMShsm performs interval migration for non-SMS-managed data sets in the same way as for SMS-managed data sets. However, after DFSMShsm finishes migrating eligible data sets on non-SMS-managed volumes during interval migration, it performs extent reduction for the volumes.

Recall

Recall for non-SMS-managed data sets that are not recalled to SMS-managed volumes differs from that for SMS-managed data sets primarily in that, for non-SMS-managed data sets, DFSMShsm selects the volumes to which the data set is recalled. For recall target volume selection, DFSMShsm considers volumes in the following terms:

- Data set pools
- Volume pools
- General pool (JES3), a single pool that DFSMShsm configures

Related reading:

- For additional information about how to define pools, see "Defining Pools of Volumes" on page 85.
- For additional information about recall requests and the common recall queue, see "Common Recall Queue" on page 109.

Recall Volume Selection

When DFSMShsm receives a non-directed recall request (either by command or by allocation with no volume specified), it builds a list of up to five primary volumes with the most available space. The space on the volumes is based on the last space check performed on the primary volumes. DFSMShsm selects the primary volumes as follows:

1. If the data set being recalled has a high-level qualifier that matches that of a data set pool, the five volumes come from the data set pool.
2. If the volume from which the data set migrated is part of a volume pool, the candidate volumes come from the volume pool.
3. 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 subparameters selected for the RECALL parameter of the SETSYS command
   - The volumes having the AUTORECALL subparameter specified in the ADDVOL command and having MIGRATE as the space management technique

- The volumes being online
- The use attribute of the volume
  - Attribute of storage for the RECALL(ANYSTORAGEVOLUME) parameter
  - Attribute of storage, public, or private for the RECALL(PRIVATEVOLUME) parameter

In building the list of candidate volumes from the default pool, DFSMShsm considers the recall attributes described in "Recall Attributes" on page 85 and the LIKE|UNLIKE subparameters specified with the RECALL parameter of the SETSYS command. If LIKE is specified, DFSMShsm builds the list only from volumes with recall attributes that match the recall attributes of the volume from which the data set migrated. If fewer than five volumes have like attributes, the list of candidate volumes contains fewer than five volumes.

If UNLIKE is specified, DFSMShsm builds the list from all the volumes in the default pool in the following order:

1. If five or more volumes exist with like recall attributes, all the volumes in the candidate list have recall attributes that match those of the volume from which the data set migrated.
2. If not enough like volumes exist, DFSMShsm selects as many volumes with recall attributes that match those of the volume from which the data set migrated as possible, then selects unlike volumes in the following order:
   a. Volumes with matching backup attributes (AUTOBACKUP|NOAUTOBACKUP and BACKUPDEVICECATEGORY)
   b. Volumes with matching migration attributes (AUTOMIGRATION|NOAUTOMIGRATION)
   c. Volumes with no matching attributes

   Within this group, each subgroup is placed in order by the amount of available space on the volume. Thus, the volumes with like attributes are placed (in order by free space) ahead of the volumes with unlike attributes.

DFSMShsm never requires matching recall volume attributes for data sets under the following conditions:

- A data set being recalled to a user-defined pool
- A data set migrated by command from a volume with the DELETEBYAGE management technique
- A data set migrated by command from a volume not managed by DFSMShsm
- A data set that migrated from a primary volume that was added to only one host in a multiple DFSMShsm-host environment

A primary volume managed by DELETEIFBACKEDUP or DELETEBYAGE space management techniques will never be selected by DFSMShsm as the target of a nondirected recall.

Calling the Recall Exit

Programming Interface Information

After the list of candidate volumes has been assembled, DFSMShsm passes control to the recall installation exit (ARCRDEXT) if the installation exit is active. DFSMShsm passes the complete list of eligible volumes, not just the five that it has selected, to the installation exit. The five volumes that DFSMShsm has selected are at the top of the list. The installation exit can change the volume selection so that

different volumes are the five volumes at the top of the list, thereby changing the volumes that DFSMShsm uses for the recall. Another point of divergence from the processing for SMS-managed data sets is that control is not passed to the installation exit during the recall of SMS-managed data sets.

Note: In a common recall queue environment, the ARCRDEXT exit is invoked by the host that selects the request for processing.

End of Programming Interface Information

Allocating Recalled Data Sets to Volumes

After the list of candidate volumes has been assembled from the installation exit, DFSMShsm or DFSMSdss requests that the data set be allocated to the first volume on the candidate list. If the allocation fails for lack of space, DFSMShsm or DFSMSdss requests allocation to the next volume on the list. If this allocation also fails for lack of space, DFSMShsm or DFSMSdss tries the next volume, and proceeds down the list until either a volume with adequate space is found, or all the volumes on the list have been tried.

When DFSMShsm is the data mover, a recall can fail even though enough total space exists on the volume if the space is too fragmented to allocate the total space for the data set in five extents.

Automatic Recall during Non-VSAM Data Set Opening

If the user specifies both volume and unit information during allocation of a non-SMS-managed volume or if the allocation process is not done, a migrated data set will still be on a migration volume when the open routine tries to open the data set. When the non-VSAM open routine does not find the data set VTOC entry for a non-VSAM data set, DFSMShsm intercepts the abnormal end and tries to recall the data set. Enough space must exist for DFSMShsm to recall the data set to the volume the open routine specifies. Otherwise, the recall fails and the open routine abnormally ends.

If a user allocates an uncataloged data set and mistakenly specifies the wrong volume, the user receives a VTOC entry-not-found abnormal end. However, if a migrated data set exists with the same data set name as the data set the user is allocating, DFSMShsm can recall the migrated data set to a user's volume. The user might be able to open the data set, if it is unprotected, even though it could belong to someone else.

Automatic Recall during VSAM Data Set Opening

If the user specifies both volume and unit information during allocation of a non-SMS-managed volume or if the allocation process described in "Recall Volume Selection" on page 101 is not done for the data set, a migrated data set to be opened will still be on a migration volume when the open routine tries to open the VSAM data set. The VSAM open routine issues a catalog locate request to get information about the cluster from the catalog. DFSMShsm intercepts the locate request, recalls the data set to the primary volume it migrated from, and returns the catalog information that the VSAM open routine requested. Enough space must exist for DFSMShsm to recall the data set to the volume the open routine specifies; otherwise, the VSAM open routine fails.

Automatic Recall for Data Set Command Scratching

A non-SMS-managed, non-VSAM data set could be on a migration volume when it is about to be scratched. If a scratch is requested for a data set from a volume with a volume serial number of MIGRAT, DFSMShsm translates the request into a request to delete a migrated data set.

If the scratch request does not specify MIGRAT and results in a VTOC entry-not-found condition, the data set still could have migrated. If the scratch routine cannot find the VTOC entry for the data set on the specified volume, DFSMShsm receives a request to delete the data set if the volume specified with the VTOC entry-not-found condition is the same volume the data set migrated from last. If both of these conditions are met, DFSMShsm processes the scratch request; otherwise, the scratch fails.

A data set is not scratched or deleted by DFSMShsm if the expiration date of the data set in the data set VTOC entry is not reached and if the purge bit in the scratch parameter list is not on. When DFSMShsm receives the request to delete the migrated data set, it recalls the data set, and the scratch request continues for the recalled data set. The volume that the scratch request specified is used for the data set recall must contain enough available space to recall the data set. Otherwise, the recall and the scratch request for the data set fail.

JES3 Considerations

When the JES3 converter/interpreter is processing a data set that is migrated, it requests a list of recall volumes from DFSMShsm. DFSMShsm selects a list of recall volumes as described in "Recall Volume Selection" on page 101 and returns the list to the JES3 converter/interpreter. DFSMShsm also records the list of volumes in the MCD record. For a subsequent request for that data set from the JES3 converter/interpreter, DFSMShsm returns the same list of volumes.

When DFSMShsm returns the list of volumes, the JES3 converter/interpreter associates those volumes with the data set. Because DFSMShsm returned five or fewer volumes, JES3 associates the data set as a multiple-volume data set.

When volumes are selected from a pool, DFSMShsm uses a count of the number of migrated data sets associated with each pool volume in the general pool and the available volume space determined by the last space check. The migrated data sets are those that have been directed to a volume but have not been recalled yet. Therefore, the algorithm spreads the association of migrated data sets to all volumes in the pool.

DFSMShsm holds each subset of five or fewer volumes today and three more days by recording in the MCD record a hold-volumes date and the five or fewer volume serial numbers.The *hold-volumes date* is the last date that DFSMShsm restricts recalls for the data set to the five or fewer volumes.

DFSMShsm recalls migrated data sets when MVS allocates (nonspecific request and the volume serial number is MIGRAT) or opens (specific request and the VTOC entry is not found) the data set on a volume managed by DFSMShsm. For a subset of five or fewer pool volumes—either data set, volume, or general—DFSMShsm directs the recall of the migrated data set to one of the five or fewer volumes. DFSMShsm fails the recall request if not enough space is available on any of the volumes.

If you direct the recall of a migrated data set to a DFSMShsm-managed volume other than the subset associated by JES3 and before the hold-volumes date expires,

DFSMShsm fails the recall. A message is issued to indicate that the volume serial number you requested is not on the JES3 list. If you enter a command to delete a migrated data set before the hold-volumes date expires, DFSMShsm fails the command. A message is also issued to indicate why the delete request failed.

If space is not made available on these volumes and you must recall the data set before the hold-volumes date expires, you can reset the hold-volumes date by issuing the command:

HSENDCMD FIXCDS D *dsn* **PATCH(X'60' X'000000')**

Be sure that the appropriate volume and pool configuration of the z/OS image that will eventually process a job is described to the z/OS image where the JES3 converter/interpreter locate request is initiated during JES3 setup of the job. The DFSMShsm hosts in those z/OS images must have the same SETSYS RECALL subparameter specified.

Recall Example

Figure 18 on page 106 shows the results of recalls (either automatic or by command) that allow DFSMShsm to select the volume to which the data set is recalled. Because GROUP1.SAMPLE.A1 has the high-level qualifier of GROUP1, for which a data set pool that contains the volumes in User Group 1 was defined, it is recalled to one of the volumes in that user group. Because U1234.G3.G4 migrated from a volume in Volume Pool and a volume pool that contains those volumes was defined, U1234.G3.G4 returns to one of the volumes in the pool. LARGE.ML.1 returns to one of the common volumes (general pool in JES3) because neither of the following applies:

- It does not have the high-level qualifier to direct it to one of data set pools defined for the user groups.
- It did not migrate from one of the volumes in the volume pool.

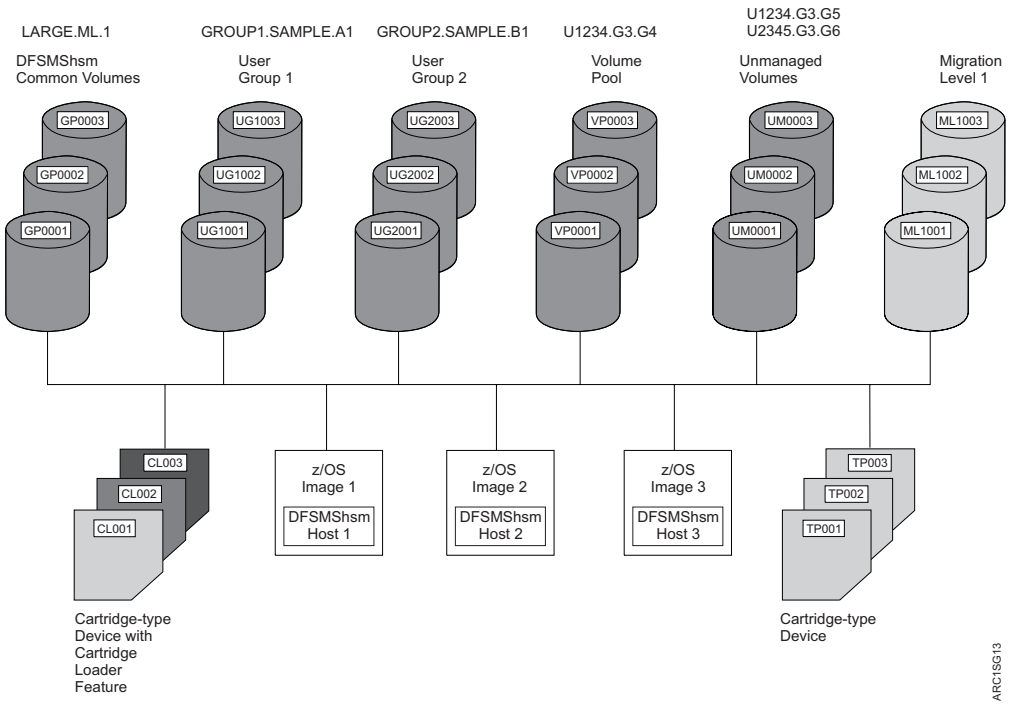


Figure 18. Example System for Non-SMS-Managed Volumes after Undirected Recall of Data Sets

Command Space Management

For non-SMS-managed storage, you can perform the same command space management as you can for SMS-managed storage:

- Delete migrated data sets
- Migrate data sets
- Migrate level 0 volumes
- Migrate level 1 volumes
- Recall data sets

For non-SMS-managed data sets, migrating level 0 volumes works somewhat differently. In addition, for non-SMS-managed data sets you can:

- Migrate from a single primary volume all data sets not used within the number of days you specify
- Delete by age from a single primary volume all data sets not used within the number of days you specify
- Delete if backed up from a single primary volume all data sets 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

Command Volume Space Management

Two forms of the MIGRATE command control volume space management for DFSMShsm-managed volumes:

```
MIGRATE VOLUME(volid)
MIGRATE VOLUME(volid DELETEBYAGE(days) |
 DELETEIFBACKEDUP(days)|MIGRATE(days))
```

The MIGRATE VOLUME command cannot be used for ML1 volumes.

For the first form, DFSMShsm performs space management for the volume in the same way as if the volume were being processed by automatic primary space management. That is, it performs all the functions described in "Automatic Primary Space Management" on page 92 for the volume, using the parameters specified in the ADDVOL command for the volume. Thus, the management technique and the inactive days specification from the ADDVOL command apply to this processing of the volume.

For the second form of the MIGRATE command, the processing described in Automatic Primary Space Management is performed, but the management technique and the number of inactive days is taken from the MIGRATE command. Thus, you can apply different technique and inactive day specifications to the volume when you do command volume migration.

Related reading: For additional information about the commands to use for volume space management of DFSMShsm-owned volumes, see Chapter 13, "Managing DFSMShsm Media," on page 483.

Command Volume Space Management (Primary)

You can cause all non-SMS-managed primary volumes to be processed by issuing the following command:

```
MIGRATE PRIMARY
```

This command has the same effect for all level 0 volumes that have been added to the system with an ADDVOL command as if the primary volume space management part of automatic primary space management for non-SMS-managed volumes had begun.

Controlling Migration of Data Sets and Volumes

You can use the SETMIG command to control migration for specific data sets, groups of data sets, or volumes. The control you can exercise is to allow all migration, to allow migration only by command, or to allow no migration. For example, if you issue the following command:

```
SETMIG VOLUME(GP0002) NOMIGRATION
```

no migration can occur from volume GP0002 until the SETMIG command is reissued to allow migration or until DFSMShsm is reinitialized. The SETMIG command for volumes and data sets is not normally included in the ARCCMD*xx* member. The SETMIG command for levels (groups of data sets) *is* usually included in the ARCCMD*xx* member.

Recalling a Data Set to a Specific Volume

You can recall a non-SMS-managed data set to a specific non-SMS-managed volume. For an example, see Figure 17 on page 100. If data set GROUP1.SAMPLE.A1 were recalled with either an automatic or command undirected recall, it would return to one of the volumes in User Group 1. However, if you issue the following command:

```
RECALL GROUP1.SAMPLE.A1 UNIT(3390) VOLUME(VP0003)
```

the data set is recalled to volume VP0003. If the ACS routine converts the data set to an SMS-managed data set (returns a storage class), SMS selects the volume to which the data set is recalled.

Recalling an SMS-Managed Data Set to a Non-SMS-Managed Volume

You cannot recall an SMS-managed data set to a specific SMS-managed volume. You can, however, recall an SMS-managed data set to a specific non-SMS-managed volume and remove the data set from SMS management. For an example, see Figure 12 on page 63 and Figure 17 on page 100. If you issue the following command:

```
RECALL GDSMC.G3.V5 UNIT(3390) VOLUME(GP0003) FORCENONSMS
```

the data set is recalled to GP0003 and is no longer under SMS management.

Recalling an SMS-Managed BDAM Data Set to a Non-SMS-Managed Volume

You can direct the recall of an SMS-managed BDAM data set to a non-SMS volume with a smaller track size than the volume from which it was migrated, using the DAOPTION and the FORCENONSMS parameters of the RECALL command. If, for instance, data set DA00001.V1 was migrated from a 3390 volume, you could recall that data set to a 3380 volume (GP3380) by issuing the following command:

```
RECALL DA00001.V1 UNIT(3380) VOLUME(GP3380) -
 DAOPTION(RELBLK) FORCENONSMS
```

The data set is recalled to GP3380, no longer under SMS management, and is ready to be accessed via relative block accessing.

Recalling an SMS-Managed Sequential Extended-Format Data Set to a Non-SMS-Managed Volume

You cannot recall an SMS-managed sequential extended-format data set to a non-SMS-managed volume by modifying your ACS routine to filter the data set to such a volume. However, if you issue the following command:

```
RECALL dsname FORCENONSMS
```

you can recall an SMS-managed sequential extended-format data set to a non-SMS-managed volume as a data set that is not sequential extended-format.

Chapter 4. Other Space Management Considerations

In addition to the aspects of space management discussed in the two preceding chapters, you need to consider the following topics:

Topic	See Page
Common Recall Queue	109
Processing Priority of Recalls and Deletions	116
Migration Volume Contention Priorities	117
Specifying the Desired Number of ML2 Partial Tapes	123
Preventing the Processing of Data Sets during Space Management	131
Deferred Generation Data Sets	131
Security Considerations for Automatic Space Management	132
Access Method Services Considerations	135
GUARANTEED SPACE Attribute Considerations	135
DFSMShsm Retrieval of Management Classes and Storage Groups	138
Deletion of Dynamic I/O Device Considerations	138

Common Recall Queue

The common recall queue (CRQ) is a single recall queue that is shared by multiple DFSMShsm hosts. This queue is implemented through the use of a coupling facility (CF) list structure. The CRQ balances the recall workload across multiple address spaces and enables optimization for priority and single-tape mounts. In some parallel sysplex® configurations, all hosts are not connected to all devices. The CRQ enables hosts that are not connected to a device, like a tape drive, to place recall requests on the CRQ. Once on the CRQ, the request can be selected by hosts that are connected to the device.

© Copyright IBM Corp. 1984, 2004

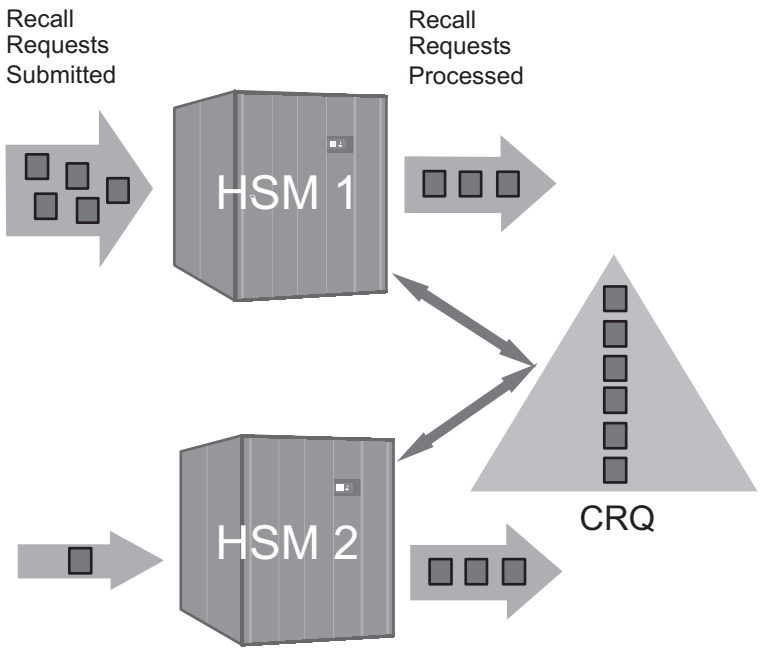


Figure 19. The Common Recall Queue

As Figure 23 shows, a CRQplex consists of one or more DFSMShsm hosts that are connected to the same CRQ. There may be multiple CRQplexes within a single HSMplex. In general, each HSMplex maintains a unique CRQ.

This CRQ support provides the following enhancements to recall processing:

Workload Balancing Alternating with all other participating hosts in the CRQplex, each host attempts to select requests from the CRQ. Each host selects requests while it has available recall tasks and the CRQ is not empty.

Priority Optimization Because all recall requests are placed and sorted on a common queue, the highest priority outstanding requests in the CRQplex are selected before lower priority requests. This prioritization occurs regardless of which system the requests originated on.

Tape Mount Optimization When a recall task mounts an ML2 tape to perform a recall, that task processes all outstanding single tape recall requests in the CRQ that require that tape. Thus, a single tape mount recalls multiple recall requests that require the same tape, even if those requests originated on different hosts.

Request Persistence Outstanding recall requests on the CRQ are maintained even when the originating DFSMShsm host shuts down or fails, or the originating system fails. Additionally, those requests are eligible for processing by other active hosts, even while the original host is unavailable.

Flexible Configurations The CRQ support enables configurations in which not all hosts are connected to all devices, and in

which not all hosts process recalls. For example, you can configure a host that is not connected to any tape drives to process only recalls that do not require ML2 tapes. Also, you can configure the HSMplex so that all hosts accept recall requests and place them on the CRQ, but only a subset of the hosts actually process the requests.

Related reading: For additional information about how to configure each host in an HSMplex for a CRQ environment, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Connecting to a Common Recall Queue List Structure

Before the CRQ function can be used, an appropriately defined list structure must be configured in the active coupling facility resource manager (CFRM) policy. After configuring the CFRM policy, update your DFSMShsm ARCCMD*xx* member to include SETSYS COMMONQUEUE(RECALL(CONNECT(*basename*))).

When SETSYS COMMONQUEUE(RECALL(CONNECT(*basename*))) is specified in ARCCMD*xx*, DFSMShsm connects to the specified structure during startup processing. If the SETSYS command is issued after startup, DFSMShsm connects to the structure and then moves any requests on the host's local recall queue to the CRQ.

If the attempt to connect fails due to an environmental error, DFSMShsm retries the connection when the coupling facility signals that the structure is available. Even though the connection may be unsuccessful, the DFSMShsm host is associated with the specified CRQ until a SETSYS COMMONQUEUE(RECALL(DISCONNECT)) command is issued.

A CRQ cannot be shared between separate HSMplexes. To enforce this, each CRQ is associated with the HSMplex that allocated it. DFSMShsm fails any attempt to connect to a CRQ that is not associated with the HSMplex in which a connection request originates.

Related reading: For additional information about configuring the list structure, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Populating the Common Recall Queue

A host that is participating in CRQplex-wide recall processing attempts to place all of its recall requests onto the CRQ. Requests are placed on the CRQ if the following conditions are true:

- The host has a successful connection to the CRQ.
- The hold status of the host and CRQ eligibility allows the request to be placed on the CRQ. The following conditions must be true to allow placement on the CRQ:
  - The host must not have HOLD COMMONQUEUE(RECALL(PLACEMENT)) or higher level of Hold in effect.
  - For WAIT requests, at least one host that is connected to the CRQ must be eligible to process the request. This is what is meant by "CRQ eligibility". This requirement prevents a WAIT request from being placed onto the CRQ and waiting indefinitely to be processed. For example, if HOLD RECALL(TAPE) is in effect on all hosts connected to the CRQ, then all WAIT

recall requests that require an ML2 tape will fail. If at least one host does not have HOLD RECALL(TAPE) in effect, then those requests are placed.

- The CRQ is less than 95% full. Or, if it recently exceeded the 95% threshold, it has dropped back below 85%.

If recall requests are unexpectedly not placed onto the CRQ, issue the QUERY ACTIVE command and examine message ARC1540I. It will indicate which of the above conditions is not being met.

Any request that cannot be placed onto the CRQ is placed on the local queue of the originating host.

Recall requests that can be placed onto the CRQ are sorted by WAIT or NOWAIT status and priority. WAIT requests are always a higher priority than NOWAIT requests. Requests of the same priority are sorted in FIFO order, with the exception that NOWAIT requests of priority 50 are subsorted with requests from multiple users. This subsorting prevents a single user with many requests from blocking other users' requests for an extended period of time.

The host that initiates the recall request invokes user exit ARCRPEXT, if the exit is enabled. The processing host invokes user exit ARCRDEXT, if the exit is enabled. You can ensure consistent processing by using common RD exits throughout the CRQplex. The RP exit may be used to assign requests that originate on one host a higher or lower priority in relation to the priorities set by RP exits on other hosts. The default priority for these requests is 50. Refer to *z/OS DFSMS Installation Exits* for a description of these user exits.

The SETSYS EMERGENCY command does not affect the placement of recall requests onto the CRQ. This allows a host that is in emergency mode to place its requests onto the CRQ and enables other active hosts to process those requests. Use the HOLD COMMONQUEUE(PLACEMENT) command to prevent recalls from being placed onto the CRQ when the emergency mode is in effect.

Selecting Requests from the Common Recall Queue

Alternating with other hosts, each host attempts to select requests from the CRQ as long as the host has available recall tasks and the CRQ is not empty. Hosts select the highest priority request that they are capable of processing. Also, hosts select recall requests that are on the host's local queue before requests on the CRQ that are of the same or lower priority. (Requests may be on a host's local queue if the requirements for placement, as mentioned in the previous section, have not been met.)

When a host selects a request that requires a single ML2 tape, that host processes all single tape requests on the CRQ from that tape. After processing each request from the tape, the processing host selects the next highest priority request that requires that tape. Requests with the same priority are processed in first-in-first-out order. After processing all the requests and after a short wait for any additional requests, the tape is demounted.

If recall requests are unexpectedly not being selected for processing, issue the QUERY ACTIVE command and examine message ARC1541I. A status of anything other than CONNECTED or a hold level of anything other than NONE are probable reasons why certain recall requests are not being selected.

Another cause for certain recall requests not to be selected is that they require an unavailable ML2 tape. If the ML2 tape does not become available within 30 minutes, these requests will time out.

Processing Recall Requests on the Common Recall Queue

A host processes a request selected from the CRQ just like a request that is submitted on its own system, with two exceptions. First, messages that are routed to TSO are sent to the original host to issue to the appropriate user. Second, when the processing host completes a recall request, it notifies the original host which then posts any WAIT requests as complete.

The processing host issues console and operator messages; messages directed to the DFSMShsm logs go to the logs for the processing host. Only the processing host creates and stores functional statistics records (FSRs). The originating host, if different from the processing host, does not have an FSR for the recall. The processing host accumulates the daily statistics record (DSR) statistics, and the DSR stores recall statistics for the entire HSMplex.

A recall request can be canceled until the point that it has been selected for processing by a host in the HSMplex. The cancel request must be submitted on the same host that originated the recall request that is to be canceled.

Note: The DFSMShsm host that places the recall request on the CRQ is the host that checks the user's authority to issue the recall command. Authority checking is performed using the DFSMShsm AUTH command or with the RACF FACILITY class. In a common recall queue environment, there is the possibility that when a recall command is placed on the CRQ (which is not checked by RACF), a DFSMShsm host that is running z/OS V1R5 and later can select the same recall request. This same recall request is not checked by RACF because RACF checking is performed prior to placing the recall on the CRQ.

Disconnecting from the Common Recall Queue

To disconnect a host from the CRQ, use the optional parameter SETSYS COMMONQUEUE(RECALL(DISCONNECT)). When the DISCONNECT command is received, the DFSMShsm host disconnects itself from the CRQ by performing the following actions:

- Directing all new recall requests to its local queue instead of the CRQ
- Moving all of its requests currently on the CRQ to its local queue
- Completing any remote requests that it had previously selected
- No longer selecting requests from the CRQ

You must wait for the disconnection to complete before attempting to connect to another CRQ. A message indicates when the disconnection is complete.

Note: Disconnecting from the CRQ should be done only on an exception basis.

Processing of Common Recall Queue Requests during Shutdown

With the CRQ, individual DFSMShsm hosts automatically disconnect from the structure during shutdown processing. Therefore, for normal processing, do not issue SETSYS COMMONQUEUE(RECALL(DISCONNECT)) before shutting down

DFSMShsm. Doing so moves all of the host's requests back to the local queue, which causes WAIT requests to be failed with message ARC1192I and to be purged from the CSA.

When a host is connected to the CRQ when shutdown processing is initiated, an outstanding WAIT recall request that is on the CRQ is failed, but a new NOWAIT request replaces this request, so that this request is preserved. The new request has the same priority as the original request, but is a NOWAIT request instead of a WAIT request. The new request is eligible for processing by any eligible host still connected to the CRQ. NOWAIT requests remain available in the CRQ.

Note: Batch WAIT requests are not converted. They remain in the CSA queue so that the request can time-out if it is not processed in a timely manner.

Deleting the Common Recall Queue Structure

If you need to delete the CRQ structure, perform the following steps:

1. Verify that the CRQ is empty by using the following command:

   QUERY COMMONQUEUE(RECALL)
2. Disconnect all DFSMShsm hosts from the CRQ structure by using the following command:

   SETSYS COMMONQUEUE(RECALL(DISC))
3. Verify that all DFSMShsm hosts have disconnected by using the following command:

   D XCF,STR,STRNAME=SYSARC_*basename*_**RCL**
4. Delete the CRQ structure by using the following command:

   SETXCF FORCE,STRUCTURE,STRNAME=SYSARC_ *basename*_**RCL**

Processing Failures in a Common Recall Queue Environment

DFSMShsm processes the following types of failures or errors:

- Coupling facility or coupling facility structure failures
- Coupling facility link communication failures
- z/OS image failures
- DFSMShsm primary address space failures
- Full list structure failures

Coupling Facility or Coupling Facility Structure Failure

If the coupling facility fails in such a way that the CRQ list structure is lost, each DFSMShsm host responds in the following ways:

1. Each host continues processing requests on its local queue and remote requests that are already selected.
2. Each host places onto its local queue the local copies of its outstanding requests and all new requests.
3. If a host starts a local request that is currently being recalled by a remote host, the user is notified that the request is currently being processed by another host.

4. Each host attempts to reconnect to the CRQ until the connection is successful. After a host successfully reconnects to the CRQ, it moves all of its unselected local requests from its local queue onto the CRQ. Requests that are currently running continue to completion.
5. The host resumes normal processing.

Coupling Facility Link Failure

If one or more hosts lose their connection to the CRQ, the CRQ list structure remains intact, but those hosts can no longer participate in the CRQ activities. A host that loses its connection performs the same steps as with a coupling facility failure. Hosts that remain connected while another host has lost its connection respond in the following ways:

- Each host finishes the remote requests that originated on the host that lost connectivity.
- Each connected host continues to select and process the requests that the failed host placed onto the CRQ before losing connectivity.

z/OS Image Failure

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. The requests that originated on the failed host remain eligible for selection. In-process requests on the failed host remain on the queue and are made available for other hosts to restart them.

If the failing host is processing a request from ML2 tape, then 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 will not be reset until the failing host restarts.

DFSMShsm Primary Address Space Failure

If a DFSMShsm host fails, all of its recall requests on the CRQ remain intact and eligible for processing by other active hosts. 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 is processing a request from ML2 tape, then 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 will not be reset until the failing host restarts.

Full List Structure Failure

If the list structure becomes 95% full, DFSMShsm routes all new requests to the local queues. Routing recall requests to the CRQ resumes once the structure drops below 85% full. The list structure is not allowed to become 100% full so that requests that are in-process can be moved between lists within the structure without failure. When the structure approaches maximum capacity, you should increase the size by altering the structure to a larger size or by rebuilding it. A rebuild must be done if the maximum size has already been reached. (The maximum size limit specified in the CFRM policy must be increased before the structure is rebuilt.) DFSMShsm relies on the XES structure full monitoring support to issue warning messages about the fullness of a structure.

Rule: If one of the above failures occurs, issue the AUDIT COMMONQUEUE command after each of the hosts has restarted or reconnected, or both.

Related reading: For additional information about altering or rebuilding a list structure, see the *z/OS MVS Programming: Sysplex Services Guide* and *z/OS MVS System Commands*.

Processing Priority of Recalls and Deletions

You may receive complaints from end users that an inordinate amount of time was required to service a long series of NOWAIT requests for recalls or deletions. The delay occurs only if DFSMShsm is heavily loaded with recall and deletion requests in relation to the number of tasks that are permitted to process them. The delay is caused by the priority scheme that DFSMShsm uses to prevent one user from monopolizing the system and from tape mount optimization processing.

The same queue of management work elements (MWEs) processes all recall and deletion requests that are submitted to one DFSMShsm host. This queue is called the recall queue. DFSMShsm gives all WAIT requests a higher priority (position) on the queue than NOWAIT requests. The NOWAIT requests are interleaved according to the user ID of the submitter. All users start one NOWAIT request before any user starts a second NOWAIT request. Processing of the requests is generally from the oldest to the newest requests.

An exception to the order of processing is when doing recalls from tape, where DFSMShsm reduces the number of mounts by doing several recalls from a given tape while the tape is continuously mounted. If a recall task has just completed a recall from a tape, it processes the highest-priority request that is currently on the queue for that same tape, no matter where the request is positioned on the queue. DFSMShsm continues to process only recalls from this single tape for a duration of time that is specified by the TAPERECALLLIMITS parameter. After the time specified by the TAPERECALLLIMITS parameter is reached, DFSMShsm determines whether a higher-priority recall request that needs this tape exists on another DFSMShsm host. If so, it ends the recall processing on the tape by the current DFSMShsm host, freeing the tape for use by other hosts. If no higher-priority request exists, DFSMShsm continues to process any recalls requested from the tape. DFSMShsm checks after every recall for a higher-priority recall request from other hosts.

When performing recalls from tape, DFSMShsm also determines if there is a higher-priority tape recall request from the same host, for a different tape, that cannot be performed because DFSMShsm is already processing at the maximum permitted tape recall task limit. If so, it ends recall processing on the tape that has met the specified TAPERECALLLIMITS processing time, enabling the higher-priority recall request from DASD or a different tape to take place.

Programming Interface Information

You can use the return-priority installation exit ARCRPEXT to change the priorities of these requests as the requests are put on the queue. ARCRPEXT allows you to assign a relative queuing priority such that WAIT requests can be prioritized relative to other WAIT requests and NOWAIT requests prioritized relative to other NOWAIT requests. You can, for example, give recalls a higher priority than deletes, or give interactive requests a higher priority than requests from batch processing. You can even negate the interleaving scheme described earlier in this section for interactive NOWAIT requests. (In the case of a NOWAIT request that is initiated from a volume recover request, the queuing priority is initialized with the value that is propagated from the volume recover request passed earlier to the exit.) For more details on ARCRPEXT, refer to *z/OS DFSMS Installation Exits* .

Note: With regard to the common recall queue, ARCRPEXT is invoked by the host that the request originated on.

End of Programming Interface Information

Migration Volume Contention Priorities

Because DFSMShsm migration volumes can also be involved in recall, recycle, tape copying, and aggregate backup activities, you must understand the usage priorities for your migration volumes. DASD ML1 volumes with small data set packing data sets allocated on them and tape ML2 volumes can interact with other functions that have a greater usage priority. When this interaction occurs, you may not get the results you are expecting.

Related reading:

- For additional information about the contention issues that can arise with DASD ML1 volumes with SDSP data sets allocated on them and tape ML2 volumes, see "SDSP Data Set Contention" on page 125 and "Tape Migration Level 2 Volume Contention".
- For additional information about the contention details related to the common recall queue, see "Common Recall Queue Environment" on page 123.

Tape Migration Level 2 Volume Contention

A tape ML2 volume can be in use in only one of the following ways at any single point in time:

- As a target volume for migration data output. Primary space management, interval migration, secondary space management, and data set and volume command management all generate migration data output.
- As a target volume for recycle.
- As a source volume for recycle.
- As a source volume for recall.
- As a source volume for TAPECOPY.
- As a source volume for ABACKUP.
- As a target volume for ARECOVER.

This section describes what DFSMShsm does to resolve volume contention among the different tasks that request the same tape ML2 volume.

Single DFSMShsm-Host Environment

This section describes the processing that occurs when a tape migration volume is needed but is not available because another task is using the tape volume.

Migration or Recycle Output Needs Volume: Independent of what type of migration or recycle task needs an output tape, if the tape initially selected is in use by any task in DFSMShsm, then the output task simply selects another target tape.

Recall Needs Volume: If recall needs a data set that is on a tape volume being used for ML2 output, then recall causes the output task to relinquish that tape and continue its output using a different tape. This applies to all forms of migration: primary space management, interval migration, secondary space management, command data set migration, and command volume migration. It also applies to recycle output.

The recall function also causes tapes to be taken away from other tasks, such as recycle input, TAPECOPY input, and from recall itself.

- For recycle input, the recycle function ends processing on the input connected set and moves on to the next best connected set.
- For TAPECOPY input, the tape takeaway function is optional. After the first recall attempt there is a time allowance before the function initiates the takeaway. This gives TAPECOPY a chance to complete. All the copying done up to the point of the tape takeaway is discarded if the TAPECOPY task does not complete.
- There is a form of recall versus recall contention, and takeaway within a host. If the maximum number of tape recall tasks are running and taking a long time, a high-priority recall for a different tape may not get started because there are no available tasks. This is possible when many requests exist for each tape and you are using tape mount optimization. For this condition DFSMShsm provides a task takeaway consideration at a specified amount of time after a tape is first mounted for recalls. After the specified amount of time is met, a higher-priority request causes recall processing to end on the current tape, freeing a task for the highest-priority request.

Because a takeaway does not happen instantaneously, the recall task does not wait, but leaves this recall on the queue to be retried at approximately two-minute intervals. This enables recalls that are readily performed to proceed. When migration or recycle processing is complete for the data set being processed, the allocated volume is released and a new volume is selected for migration or recycle processing.

The recall function can also contend with ABACKUP and ARECOVER.

- If ABACKUP is reading an ML2 tape and recall needs that tape, there are two possibilities:
  - If the recall request is a NOWAIT request, recall repeatedly tries to access the tape until it reaches the specified limit for retries. A NOWAIT recall does not take tapes away from ABACKUP.
  - If the recall request is a WAIT request, recall repeatedly tries to access the tape. However, it does not flag ABACKUP to give up the tape until ten minutes after the first time recall finds the tape is in use. This ten minute delay gives ABACKUP time to complete its use of the tape. After ABACKUP is notified, recall repeatedly tries to access the tape until (unlikely in this case) it reaches the specified limit for retries.
- If ARECOVER is writing an ML2 tape and recall needs that tape, then recall keeps retrying to access that tape until it reaches the specified limit for retries. Recall does not take tapes away from ARECOVER.

When recall completes its use of a tape taken away from other tasks, the tape is released and is available for continued use by other tasks.

If a tape needed by recall does not become available to the recall task after 15 retries, DFSMShsm sends message ARC0380A to the operator. The operator can give one of the following answers:

WAIT This reply causes the recall task to reset its count of attempts, and begin another sequence of up to 15 retries at two-minute intervals.

CANCEL This reply causes the recall task to end without even allocating the

tape, which causes the recall to fail. If you know that the tape will not become available in a reasonable amount of time, use this choice.

MOUNT This reply causes the recall to go ahead and request that the subject tape be allocated and mounted. Use this choice only when you are certain the tape can be mounted. An example is if DFSMShsm says that the tape is in use by a host that is down and is not running.

Related reading:

- For additional information about the delay time associated with ABACKUP and how to change the delay time, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about changing the number of recall retries, see the *z/OS DFSMShsm Implementation and Customization Guide*, under supported patches. If the number of retries is patched to zero, the recall immediately fails.

ABACKUP Needs Input Volume: If ABACKUP needs the migration tape that is in use by recall, ABACKUP performs retries at 20-second intervals until the tape becomes available. It continues to retry for a period of 30 minutes, whichever occurs first. If the time expires, then the operator has the option to retry or to fail the ABACKUP. There is no takeaway function by ABACKUP from recall.

The ABACKUP function can cause takeaway from recycle input and from TAPECOPY input.

RECYCLE The recycle function ends the input connected set and moves on to the next best connected set.

TAPECOPY The TAPECOPY function is optional, and there is a time allowance after the ABACKUP function is first attempted before it initiates the takeaway. This occurs so that the TAPECOPY might complete, rather than discard, its work.

If ABACKUP needs the migration tape that is in use by migration or recycle output, ABACKUP initiates a tape takeaway. It then performs retries at 20 second intervals until the tape becomes available, or for a period of 30 minutes, whichever occurs first. If the time expires, the operator has the option to redo the set of retries or to fail the ABACKUP processing, as described in message ARC6254A.

If ABACKUP needs the migration tape that is in use by another ABACKUP, then it skips processing that tape and proceeds to process other migration and user tapes. It later reattempts to process the tape that was in use.

Recycle Input Volume In Use: If migration or recall is using the volume that is needed as a source volume for recycle, and the recycle request is a specific RECYCLE VOLUME(volser) request, the recycle fails. If the recycle request is a generic RECYCLE request, then the recycle of that connected set is skipped and message ARC0845I is issued to indicate such. Recycle processing continues with the next eligible connected set. If someone issues a RECYCLE command for a specific volume and that volume is the target recycle volume, DFSMShsm selects another target ML2 volume. DFSMShsm then recycles the specified volume.

Table 8 is a summary of possible volume contention in a single DFSMShsm-host environment or within the same DFSMShsm host in a multiple DFSMShsm-host environment.

Table 8. Volume Contention in a Single DFSMShsm-Host Environment

Function That Has Volume	Function That Needs Volume	Result
Migration	Migration	No contention.
Recall	Migration	Migration selects a new target volume.
Recycle (source)	Migration	Migration selects a new target volume.
Recycle (target)	Migration	No contention.
Migration, Recycle	Recall	Recall initiates a tape takeaway, and periodically retries the recall. When migration or recycle processing is complete for the data set being processed, the allocated volume is released and a new volume is selected for migration or recycle processing.
Recall	Recall	Terminate in favor of a higher priority recall from a different tape if running at maximum tape tasks, and it has been greater than specified amount of time since tape began to be continuously mounted.
ABACKUP	Recall	Recall periodically retries accessing the data set on the tape. If Recall is a WAIT request, after a delay, it initiates a tape takeaway, which causes ABACKUP to skip any remaining data sets that are needed by ABACKUP on the tape and process from other migration and user tapes. ABACKUP later tries to process those data sets that were skipped earlier.
ARECOVER	Recall	Recalls are retried until the tape is available or a retry limit is reached.
TAPECOPY (source)	Recall	If tapecopy takeaway is enabled, Recall retries for 15 minutes before requesting tape takeaway. If TAPECOPY has not completed its copy within one minute of the tape takeaway request, the copy fails.
Migration	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recall	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recycle (source)	Recycle (source)	No contention.
Recycle (target)	Recycle (source)	Recycle selects a new target volume.
Migration	Recycle (target)	No contention.
Recall	Recycle (target)	Recycle selects a new target volume.
Recycle (source)	Recycle (target)	Recycle selects a new target volume.
Recycle (target)	Recycle (target)	Recycle selects a new target volume.

Table 8. Volume Contention in a Single DFSMShsm-Host Environment (continued)

Function That Has Volume	Function That Needs Volume	Result
Migration, Recycle	ABACKUP	ABACKUP initiates a tape takeaway and periodically retries the data set. When migration or recycle processing is complete for the data set being processed, the allocated volumes are released. A new target tape is selected for migration or recycle output contention, or a new connected set is selected if there is recycle source contention.
Recall	ABACKUP	ABACKUP is retried until the tape is available or a retry limit is reached.
ABACKUP	ABACKUP	ABACKUP skips over the tape in use and processes from other migration tapes. It reattempts to process the skipped tape later.
TAPECOPY (source)	All except Recall and ABACKUP	Function fails.
TAPECOPY (source)	ABACKUP	If tapecopy takeaway is enabled, ABACKUP retries for 15 minutes before initiating a tape takeaway. If TAPECOPY has not completed its copy within one minute of noting the takeaway request, the copy fails.

Multiple DFSMShsm-Host Environment

This section describes the processing that occurs when a tape ML2 volume is needed but is not available because either another task in this DFSMShsm host, or a task in another host is using the tape volume. The contention processing is so similar for multiple tasks or hosts to the processing described in the preceding sections for a single host environment, that only the differences are discussed here. The differences include the following situations:

Recall Needs Tape In Use by Recall on a Different Host: The following occurs when DFSMShsm is doing recall mount optimization, which means that several recalls occur end to end while a tape is continuously mounted. Since this could go on for an indefinite length of time, DFSMShsm has the SETSYS option TAPERECALLLIMITS. TAPERECALLLIMITS provides a time value after which a tape can be taken away from recall by recall on another DFSMShsm host. The takeaway occurs only if the request on the other host is of higher priority than the request that would otherwise be performed without demounting the tape. The priority is usually based on time of arrival, but the priority can be changed by using the ARCRPEXT installation exit.

Note: A common recall queue environment needs only one tape mount. The host that mounts the tape processes all requests regardless of which hosts originate the requests.

Recycle Needs Tape In Use by Recycle on a Different Host: Since recycle can occur concurrently on different DFSMShsm hosts, it is possible for a user on one host to request recycle of a tape that is currently in use by recycle on another host. In this case the request fails, regardless of whether the request was for use of the tape for recycle input or recycle output.

Table 9 on page 122 is a summary of possible volume contention in a multiple DFSMShsm-host environment.

Table 9. Volume Contention in a Multiple DFSMShsm-Host Environment

Function That Has Volume	Function in Other Host That Needs Volume	Result
Migration	Migration	No contention.
Recall	Migration	Migration selects a new target volume.
Recycle (source)	Migration	Migration selects a new target volume.
Recycle (target)	Migration	No contention.
Migration, Recycle	Recall	Recall initiates a tape takeaway, and periodically retries the recall. When migration or recycle processing is complete for the data set being processed, the allocated volume is released and a new volume is selected for migration or recycle processing.
Recall	Recall	A specified period of time after an ML2 tape is mounted, and when a recall completes, the tape is taken away if a request exists on another host that is of higher priority than the request that normally would have been performed on the recalling host from the already mounted tape.
ABACKUP	Recall	Recall periodically retries accessing the data set on the tape. If Recall is a WAIT request, after a delay, it initiates a tape takeaway, which causes ABACKUP to skip any remaining data sets that are needed by ABACKUP on the tape and process from other migration and user tapes. ABACKUP later tries to process those data sets that were skipped earlier.
ARECOVER	Recall	Recalls are retried until the tape is available or a retry limit is reached.
TAPECOPY (source)	Recall	If tapecopy takeaway is enabled, Recall retries for 15 minutes before requesting tape takeaway. If TAPECOPY has not completed its copy within one minute of noting the tape takeaway request, the copy fails.
Migration	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recall	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recycle (source)	Recycle (source)	Recycle skips the volume because another host is recycling it.
Recycle (target)	Recycle (source)	Recycle fails.
Migration	Recycle (target)	No contention.
Recall	Recycle (target)	Recycle selects a new target volume.
Recycle (source)	Recycle (target)	Recycle selects a new target volume.
Recycle (target)	Recycle (target)	No contention.

Table 9. Volume Contention in a Multiple DFSMShsm-Host Environment (continued)

Function That Has Volume	Function in Other Host That Needs Volume	Result
Migration, Recycle	ABACKUP	ABACKUP initiates a tape takeaway and periodically retries the data set. When migration or recycle processing is complete for the data set being processed, the allocated volumes are released. A new volume is selected for migration or recycle output contention, or a new connected set is selected if there is recycle source contention.
Recall	ABACKUP	ABACKUP is retried until the tape is available or a retry limit is reached.
ABACKUP	ABACKUP	ABACKUP skips over the tape in use and processes from other migration tapes. It reattempts to process the skipped tape later.
TAPECOPY (source)	All except Recall and ABACKUP	Function fails.
TAPECOPY (source)	ABACKUP	If tapecopy takeaway is enabled, ABACKUP retries for 15 minutes before initiating a tape takeaway. If TAPECOPY has not completed its copy within one minute of noting the takeaway request, the copy fails.

Common Recall Queue Environment

With all hosts in an HSMplex connected to the same common recall queue (CRQ), no contention for the same tape exists within the recall function. This makes the recall tape takeaway from recall function obsolete. With multiple CRQplexes in the same HSMplex, recall tape takeaway from recall manages contention between the separate CRQplexes. With one or more hosts not connected to a CRQ, recall tape takeaway from recall manages contention between the unconnected hosts and the CRQ.

For hosts connected to a CRQ, recall task takeaway becomes involved only when recalls are directed to a host's local queue. It then becomes possible for that host's tape tasks to process multiple requests from single tapes while a higher priority tape request is on its local queue.

Using the recall takeaway function, any host that is connected to the CRQ can process the recalls from a tape that was marked for takeaway by another host that was connected to the same CRQ.

Related reading: For additional information about CRQplex configuration, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Specifying the Desired Number of ML2 Partial Tapes

ML2 tapes are written by the various migration functions and by the recycle of ML2 tapes. A relatively small number of ML2 partially filled tapes are needed as substitutes for tapes that are receiving output when they are taken away by a recall task. The partially filled ML2 tapes that are used as substitutes are not currently associated with specific ML2 output tasks for any hosts.

These substitute tapes are ML2 "floaters." To have the recycle process limit the number of floaters to *nnn*, specify

SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(nnn)

To allow an unlimited number, specify

SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(NOLIMIT)

Floaters are writeable ML2 tapes that are not assigned or associated with a specific task in a specific host. When a floater substitutes for a tape that is taken away, it becomes an associated tape, restricted for output to that one specific task.

An output tape taken away by recall is not immediately available to receive output again. It becomes available for output on the host where the recall occurred as the recall completes. A long queue of recalls from that tape would likely cause a delay in considering such a tape for output again.

DFSMShsm hosts other than the one doing the recall do not become aware of a tape's availability for output until their own in-storage selection table is depleted and rebuilt. For performance reasons, in-storage selection tables are not rebuilt more often than every 45 minutes.

A configuration where a storage group uses multiple SMS tape libraries can affect the number of floaters that are needed. As functions such as primary space management, interval migration, or recycle start, they can allocate a drive from any of the libraries associated with the storage group. A tape must be in the same library as the drive that is allocated. This tends to increase the number of floaters needed, even without a high tape take-away rate. The tape associated with the previous processing of a task may not be part of the library that contains the allocated drive. In this case, a floater is used, and the previously associated tape is placed in its host's TVT.

From the preceding descriptions, you can see how data usage and configuration can result in a need for a higher number of ML2 floaters. Having floaters is simply a trade-off between having a few additional under-utilized ML2 tapes, or recycling the additional few under-utilized ML2 tapes. The DFSMShsm design is to extend a partially filled ML2 tape instead of using an empty tape. However, in cases where partially filled ML2 tapes do not exist, DFSMShsm requests a scratch tape.

It is *only* when recycle of ML2s occurs that the value specified by ML2PARTIALSNOTASSOCIATEDGOAL is used. The default used for ML2PARTIALSNOTASSOCIATEDGOAL is 10. This value applies to the total number of ML2 not-associated partial tapes in an entire HSMplex, but only for tapes that meet the recycle selection criteria of PERCENTVALID and any SELECT filters. If the number of such floaters, at the time a recycle command runs, exceeds the specified value, then the excess tapes are included as candidates for recycle. The floaters that are thus recycled are those having the lowest percentage of valid data.

You can use the LIST TTOC SELECT(NOTASSOCIATED) command to get a list of all not-associated ML2s if you want to know how many or which ones they are. If you observe a large number of not-associated ML2 partials being recycled, and this seems to occur consistently, then you probably want to increase the SETSYS

specification to reduce the recycle load. This is just a trade-off between having the few under-utilized tapes or recycling the tapes so you do not have them (at least for a while).

You can use the LIST TTOC SELECT(ASSOCIATED) command to get a list of all ML2 partials that are associated with specific ML2 output tasks. If some of these are associated with tasks that are unused, or seldom used, then you may want to specify DELVOL UNASSIGN so that they will be part of the set of floaters. To find out which ML2 tapes are associated with what tasks, display the various host MCR records using the FIXCDS command. If they are old-technology cartridges, or recorded in old technology not being written anymore, you should use the DELVOL MARKFUL command so they can be recycled when they meet that criteria.

One approach for determining a unique setting for your installation is to mark the existing ML2 partial tapes as full. Then set ML2PARTIALSNOTASSOCIATEDGOAL to NOLIMIT for a while, and issue the LIST TTOC command with SELECT(ASSOCIATED) and SELECT(NOTASSOCIATED) every few days. When the sum of these two categories settles down or reaches a maximum, your migration workload indicates that you can set ML2PARTIALSNOTASSOCIATEDGOAL to or slightly below the number of not-associated partials indicated by the LIST TTOC command.

SDSP Data Set Contention

Because SDSP data sets reside on ML1 volumes, they can be involved in various ways with DFSMShsm processing of level 0 volumes and ML2 volumes. For example, an SDSP data set can be a target for automatic primary space management and, at the same time, the SDSP can be a source for a user's recall of a small data set from an SDSP data set. SDSP data sets can be included in aggregate backup and SDSP data sets require cleanup after a user recalls a migrated data set. Additionally, the AUDIT and FREEVOL functions can be run against volumes on which SDSP data sets reside.

Because SDSP data sets are potentially involved in various DFSMShsm processes, there are times when concurrent DFSMShsm processing activities cause contention for SDSP data sets. Figure 20 on page 126 shows that an SDSP data set can be effected by various DFSMShsm processes. Table 10 on page 126 shows the usage priorities for SDSP data sets and describes how DFSMShsm resolves contention for SDSP data sets.

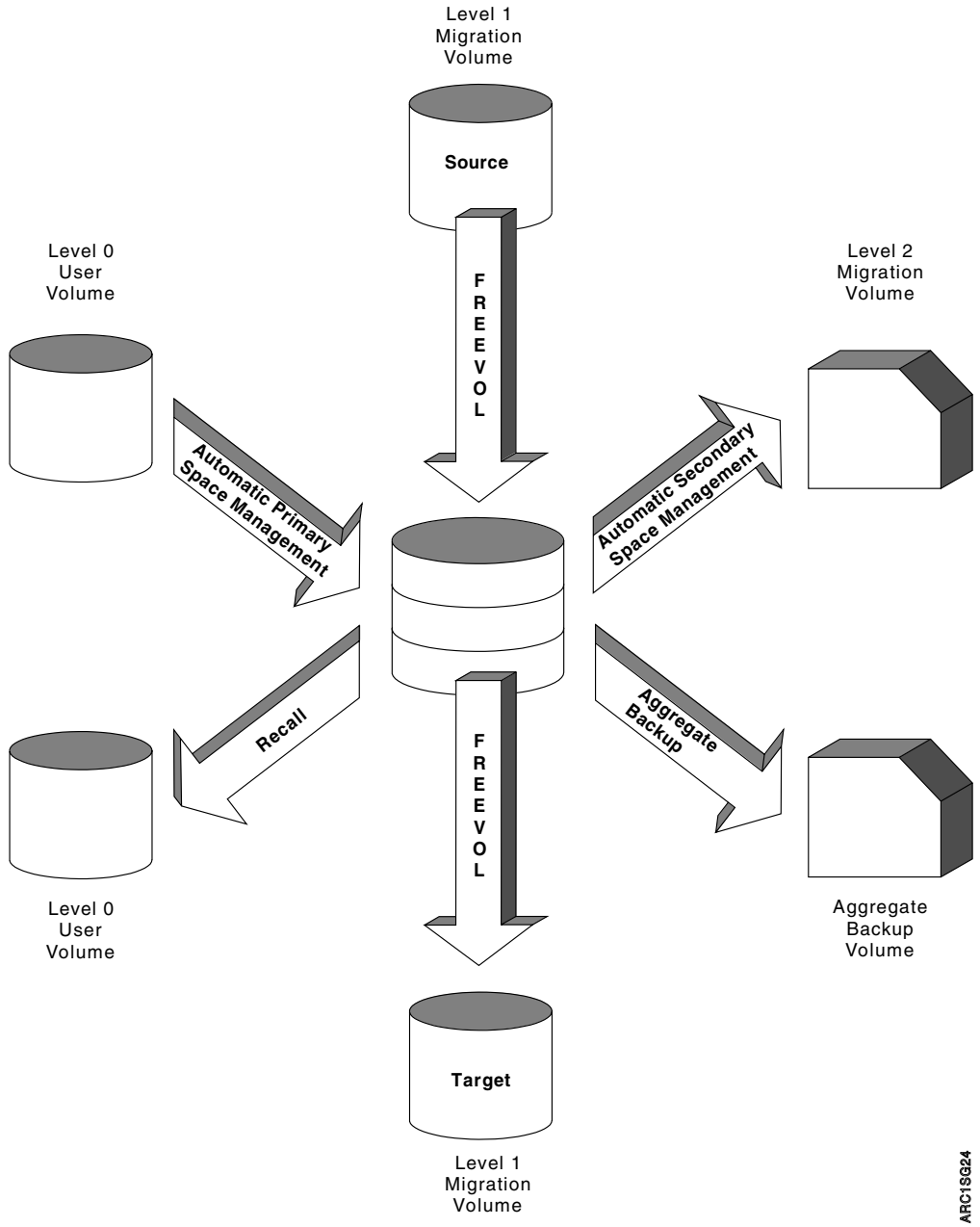


Figure 20. SDSP Contention Environment. SDSP data sets (on ML1 volumes) are potentially involved in various DFSMShsm processes.

Table 10. SDSP Contention in Single and Multiple DFSMShsm-Host Environments

Function That Has SDSP	Function That Needs SDSP	Result
Migration (L1 to L2)	Migration (L0 to L1)	If all other SDSP data sets are presently in use, this small user data set is skipped and, most likely, can migrate to an SDSP during your next migration window.
Recall		
FREEVOL (source)		
AUDIT MEDIACONTROLS (source)		
Aggregate backup		

Table 10. SDSP Contention in Single and Multiple DFSMShsm-Host Environments (continued)

Function That Has SDSP	Function That Needs SDSP	Result
Migration cleanup or migration (L1 to L2)	Migration (L0 to L1)	Migration cleanup and level 1 to level 2 migration are functions of automatic secondary space management. Because SDSP data sets can be open for a long time during migration cleanup or during level 1 to level 2 migration, it is likely that they will be in use. If all SDSP data sets are presently in use, this small user data set is skipped and, most likely, can migrate during your next migration window. **Note:** Do not run automatic primary space management and automatic secondary space management concurrently.
Migration (L0 to L1)	Recall	Migration processing gives up its use of this SDSP data set to the recall function that requests the SDSP data set. If all SDSP data sets are presently in use, the migration of this small user data set is skipped and, most likely, can migrate during your next migration window.
Recall	Recall	The first recall request to access the SDSP data set completes its processing. Other recall requests retry the SDSP data set request and, most likely, will be successful.
FREEVOL (source) AUDIT MEDIACONTROLS	Recall	Because FREEVOL and AUDIT MEDIACONTROLS are relatively long-running processes, the recall request will either retry until successful or retry until timing out and failing.
Migration cleanup	Recall	Migration cleanup gives up its use of this SDSP data set to the recall function that requests the SDSP data set. Migration cleanup skips its processing of the SDSP data set.
Aggregate backup	Recall	Recall processing retries the recall request and, most likely, will be successful after the aggregate backup has finished.
Migration (L0 to L1) Recall AUDIT MEDIACONTROLS Migration cleanup Aggregate backup	FREEVOL (source)	FREEVOL processing fails.
Migration (L0 to L1) Recall AUDIT MEDIACONTROLS Migration cleanup Aggregate backup	FREEVOL (target)	If the target SDSP data set cannot be made available to FREEVOL processing, this small data set is skipped and cannot be moved from the source SDSP data set.

Table 10. SDSP Contention in Single and Multiple DFSMShsm-Host Environments (continued)

Function That Has SDSP	Function That Needs SDSP	Result
Migration (L0 to L1) Recall FREEVOL AUDIT MEDIACONTROLS Migration cleanup Aggregate backup	AUDIT MEDIACONTROLS	If this SDSP data set is in use by other functions, the AUDIT request for the SDSP data set fails.
Migration (L0 to L1) Recall FREEVOL AUDIT MEDIACONTROLS Aggregate backup	Migration cleanup	If the SDSP that is needed by migration cleanup is in use, this small user data set is skipped and it is deleted the next time automatic secondary space management is run.
Migration (L0 to L1)	Aggregate backup	Migration processing gives up its use of the SDSP data set. If all SDSP data sets are presently in use, this small user data set is skipped and, most likely, can migrate during your next migration window.
Recall	Aggregate backup	Aggregate backup will retry its request for the SDSP and will eventually be granted access.
FREEVOL (source)	Aggregate backup	Aggregate backup fails.
FREEVOL (target)	Aggregate backup	FREEVOL processing gives up its use of this SDSP data set to the aggregate backup request. If no other SDSP data sets are available, FREEVOL processing skips the data sets.
AUDIT MEDIACONTROLS	Aggregate backup	Aggregate backup fails.
Migration cleanup	Aggregate backup	The SDSP data sets are released from migration cleanup. Migration cleanup skips processing from this SDSP data set.
Aggregate backup	Aggregate backup	The first aggregate backup request to reach the SDSP gets access to the SDSP. Other aggregate backup requests will retry until successful.

The functions of aggregate backup, aggregate recover, FREEVOL, and AUDIT should be run at times when they will not be contending for the use of the same SDSPs.

Multitasking Considerations for SDSP Data Sets

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 done by functions with a higher usage priority for the SDSP data sets.

It seems obvious that one SDSP data set is required for each concurrent migration task. However, there are some DFSMShsm activities that have a higher usage priority for SDSP data sets. These processing activities are:

- Recall
- Aggregate backup

- FREEVOL
- AUDIT MEDIACONTROLS
- Automatic secondary space management

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. The small data set that is in need of migration must find a different, available SDSP data set or it is skipped and left unprocessed until your next migration window.

Additionally, if all SDSP data sets should become full (as a result of migrations to them) the filled SDSP data sets are not candidates for further migration. Full SDSP data sets are not seen by migration processing and, as a result, any small user data sets are migrated as large data sets to ML1 volumes.

Example: The following three-part example (see Figures 21, 22, and 23) shows how SDSP data sets can become unavailable for use as level 0 to level 1 migration targets. In Part 1, three concurrent migration tasks are moving three small user data sets from level 0 user volumes to ML1 volumes (with SDSP data sets defined on the ML1 volumes).

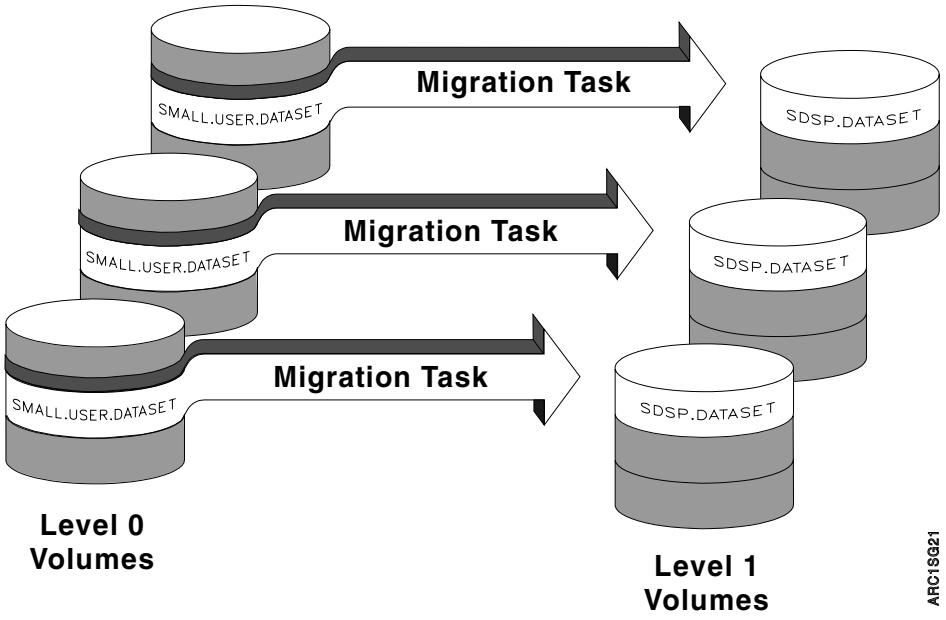


Figure 21. Part 1—Ideal Multitasking Migration. Three migration tasks migrate three small user data sets to three ML1 volumes on which SDSP data sets are defined.

In Part 2, a recall of a small user data set from an SDSP data set during level 0 to level 1 migration has effectively eliminated one concurrent migration task. The small-user data set, whose migration was preempted by a recall, is skipped and will not be migrated until the next migration window.

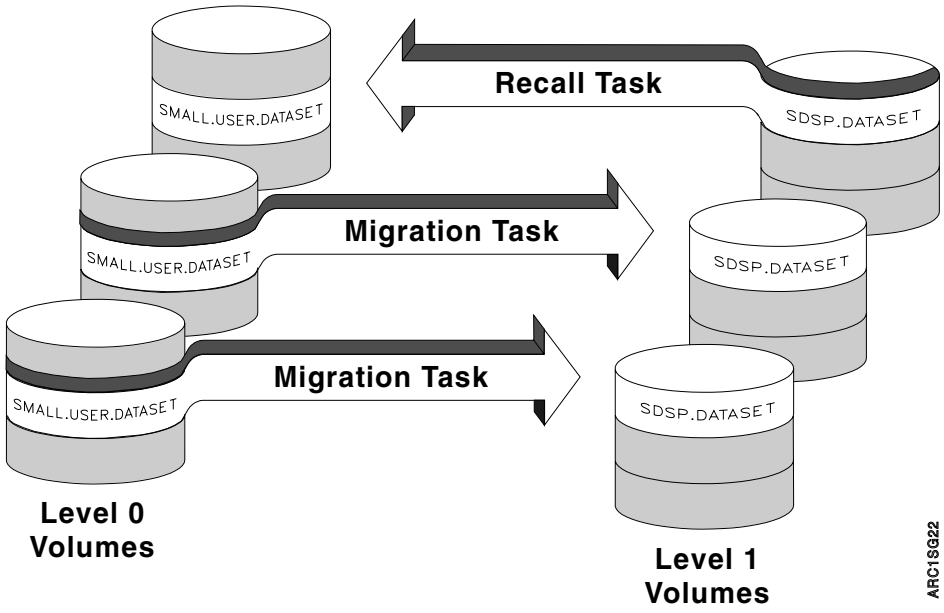


Figure 22. Part 2—Recall Processing Has a Higher Priority than Migration. One migration task does not process the small data sets because recall processing has a higher usage priority for the SDSP than migration processing.

In Part 3, all SDSP data sets have become full. They are no longer seen as candidates for level 0 to level 1 migration destinations and the small-user data sets migrate as large data sets.

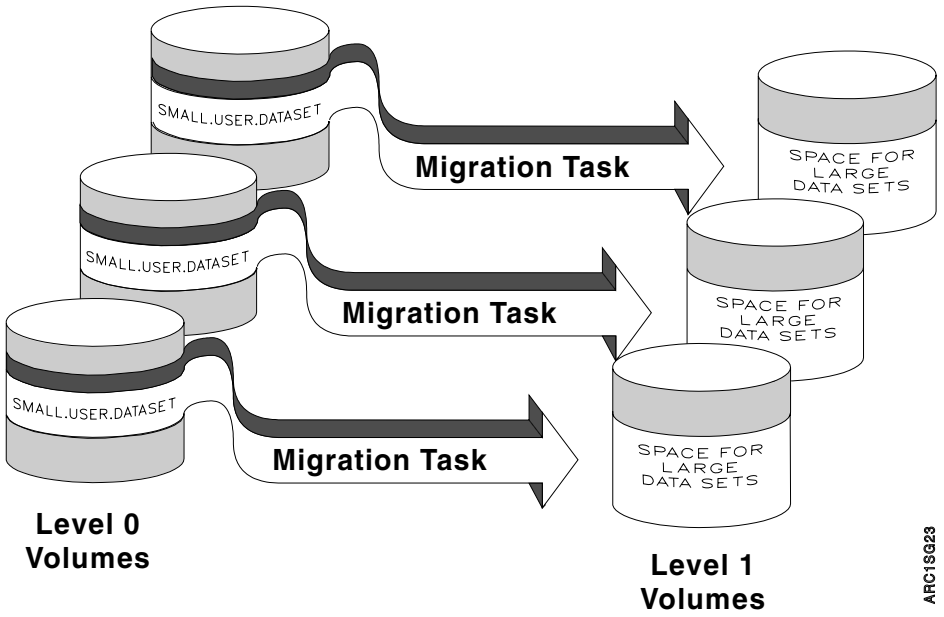


Figure 23. Part 3—All SDSP Data Sets Are Full. The small user data sets migrate as large data sets to the ML1 volumes.

In addition to planning the timing of activities whose SDSP usage priority is high, you must ensure that you do not allow automatic secondary space management to run at the same time that automatic primary space management is running. Because SDSP data sets are opened and allocated for the duration of automatic secondary space management, they cannot concurrently be targets for primary space management processing.

The point here is that other activity can affect your migrations and you need to plan and monitor those activities that can cause your small user data set migrations to be skipped. Therefore, you should define ample SDSP data sets to handle your worst case scenario (remember that one ML1 volume can have at most a single SDSP data set) and run automatic secondary space management at a different time than you run automatic primary space management.

Related reading: For additional information about the SDSP parameter of the SETSYS command, see the *z/OS DFSMShsm Storage Administration Reference*.

Preventing the Processing of Data Sets during Space Management

Any of the following parameters or values specified with the following commands can prevent an otherwise eligible volume or data set from being processed by space management:

- The DEBUG parameter of the SETSYS command.
- The NOMIGRATION parameter of the SETMIG command.
- The EXITON and EXITOFF parameters of the SETSYS command, which specify the data set migration installation exit or the data set deletion installation exit. DFSMShsm supplies sample exits that you can tailor to fit your needs.

Any of the following parameters or values specified with the following parameters can cause space management to stop processing or to fail to start processing:

- The MIGRATION(AUTO) parameter of the HOLD command. This stops or prevents automatic volume space management and automatic secondary space management from starting.
- The MIGRATION parameter of the HOLD command. This stops or prevents automatic and command volume space management, automatic secondary space management, and command data set migration from starting.
- The PRIMARYSPMGMTSTART parameter of the SETSYS command when it does not have a starting time specified. The ending time can also prevent space management from processing additional volumes.
- The SECONDARYSPMGMTSTART parameter of the SETSYS command when it does not have a starting time specified. The ending time can also prevent space management from processing additional data sets.
- The EMERGENCY parameter of the SETSYS command. This specifies that an emergency mode is in effect, and no DFSMShsm processing related to moving user data is performed.

  Note: This parameter does not affect the placement of recall requests onto the common recall queue. This enables this host to continue to place recalls onto the common recall queue that are eligible to be processed by other active hosts.
- The STOP command. This causes an orderly shutdown of DFSMShsm.
- The REQUEST parameter of the SETSYS command. The operator can deny permission to start or restart automatic space management.

Deferred Generation Data Sets

Deferred generation data sets are generation data sets (GDSs) for which a failure occurred before the data set was properly associated with the generation data group (GDG). These data sets are temporary in the sense that at some time they may become associated with the GDG. DFSMShsm treats deferred generation data sets as permanent non-GDS data sets.

Security Considerations for Automatic Space Management

DFSMShsm bypasses data set security checking during automatic space management, except after automatic recall. After automatic recall, security checking is performed by OPEN. This is to ensure that the user opening the recalled data set is authorized to use the data set.

DFSMShsm also bypasses security checking when it processes operator commands entered at the system console or commands issued by a DFSMShsm-authorized user.

DFSMShsm checks authority for a requester to access data sets when a requester who is not DFSMShsm-authorized issues an HMIGRATE, HDELETE, or HRECALL command. Security checking is not done when DFSMShsm-authorized users issue the user commands. If users are not authorized to manipulate data, DFSMShsm does not permit them to migrate data sets, delete migrated data sets, or explicitly recall data sets. DFSMShsm always does security checking for the scratch intercept when deleting migrated data sets. Table 11 on page 133 shows the RACF authority required to perform each space management function.

Table 11. RACF Authority Required for Space Management Functions

DFSMShsm Function	RACF Resource Access Authority Required
Migrate a data set	Update
Recall a data set using HRECALL (TSO users only)	Execute
Delete a migrated data set	Alter

Space Management of Data Sets with Discrete Data Set Profiles

Discrete data set profiles provide a one-to-one relationship between a specific data set on a specific volume and its security authorization requirement. When you use only discrete data set profiles, you should use the RACFIND parameter of the SETSYS command. For a description of discrete data set profiles, see "Discrete Data Set Profiles" on page 574.

Note: Supported multivolume non-VSAM data sets cannot be protected by discrete profiles. Use generic profile protection for these data sets.

Migrating RACF-Protected Data Sets

When an existing data set migrates to a DASD migration volume, the RACF indicator in the data set VTOC entry of the migration copy is set if the RACFIND parameter of the SETSYS command is in effect.

Notes:

1. To maintain data set integrity on your system's small data set packing data sets, protect them with a discrete or generic profile.
2. When you migrate a data set, the *volser* in the discrete profile is not updated. Attempts to access the data set without recalling it may cause the ICH408I INSUFFICIENT ACCESS AUTHORITY message to be issued, depending on access intent and access allowed. The message will contain the user, group, data set name, volume(MIGRAT), access intent, and access allowed.

   This message is issued because the catalog lookup before the RACHECK returns a volume serial number of MIGRAT. Therefore, MIGRAT is specified as the volume in the RACHECK. If RACF cannot find a discrete profile for the data set on *volser* MIGRAT, then a check is made for a generic profile. If a generic profile does not exist or does not give the requested access, the error message is issued.

   This error message will not be issued if specific generic profiles are used as opposed to discrete profiles.

Deleting RACF-Indicated Migrated Data Sets

When DFSMShsm deletes a migrated data set, it deletes the RACF discrete data set profile for the following components:

- Non-VSAM data sets.
- VSAM clusters eligible for volume migration. These clusters can contain up to eight specific components:
  - Base cluster
  - Base data object
  - Base index object
  - One base path
  - One alternate index (AIX)

  - AIX data object
  - AIX index object
  - One AIX path
- VSAM clusters with RACF discrete profile protection only on the base cluster.

If you want to delete migrated VSAM clusters that are not eligible for level migration and whose RACF protection is not on the base cluster, perform the following actions:

1. Recall the data set.
2. Use an Access Method Services command to delete the cluster and any associated RACF profiles.

Recalling RACF-Indicated Data Sets

If the RACF indicator was on when DFSMShsm migrated the data set, the RACF indicator is set on when DFSMShsm recalls the data set. When DFSMShsm recalls a non-VSAM data set, DFSMShsm changes the volume serial number and device type information, if necessary, in the RACF discrete data set profile to indicate the volume where DFSMShsm has recalled the data set. When DFSMShsm recalls a VSAM data set, it does not update the RACF discrete data set profile, because the profile contains the volume serial number of the catalog, which cannot change during recall or recovery.

To recall a RACF-indicated data set that has no RACF profile, perform the following actions:

1. Create a generic profile or recreate the RACF discrete profile. To create the discrete profile, use the RACF ADDSD command and specify the NOSET parameter.
2. Recall the data set without specifying where the data set should reside.

Password Protection of Data Sets on DASD

First, you must remember that password protection applies only to non-SMS-managed data sets.

Password protection in MVS is implemented differently for non-VSAM and VSAM data sets. The passwords for all non-VSAM data sets are stored in one system password data set. The password for each VSAM data set is stored in a catalog record for the specific VSAM data set. Therefore, VSAM passwords are maintained during DFSMShsm processing as a part of regular catalog record creation and updating.

If you are recalling a non-VSAM data set, the required password is the read-level password currently in the system password data set. If you are recalling a VSAM data set, the password is the one that existed when DFSMShsm migrated the data set. If you are deleting a migrated non-VSAM data set, the current password is the write-level password.

Be aware of the following conditions when you attempt to change password protection while data sets are migrated or backed up:

- You cannot completely remove password protection from a migrated non-VSAM data set because the password-protection indicator is stored in control information in the migrated data. Also, the data set name of the migrated data set is not the user data set name. Therefore, DFSMShsm cannot remove a password for a migrated non-VSAM data set from the system password data set because no password exists. Even if the password were deleted while the data

set was migrated, the VTOC entry after recall would indicate that your data set is password-protected. You would need to add a password to the system password data set before you access the recalled, non-VSAM data set.

- You can change the password of a migrated non-VSAM data set by using the IEHPROGM utility. However, to change the password of a migrated VSAM data set, you must use Access Method Services. If you use Access Method Services, a migrated VSAM data set is recalled.
- If you delete a password-protected VSAM data set while it is migrated, all password protection is deleted. If you delete a password-protected non-VSAM data set while it is migrated, DFSMShsm scratches the data set and deletes the catalog entry. However, the password still exists in the system password data set, and you should delete it.
- If you try to add password protection to a migrated non-VSAM data set, the password is added to the system password data set. However, the migrated data set does not have protection, and when DFSMShsm recalls the data set, the VTOC entry does not indicate password protection.
- If you try to add password protection to a migrated VSAM data set, you must use Access Method Services. For a migrated VSAM data set, DFSMShsm recalls the data set.

Note: If DFSMShsm tries to create a catalog entry in a catalog having update protection, the operator receives a password prompt for an unauthorized DFSMShsm function from an unauthorized DFSMShsm user.

Access Method Services Considerations

Access Method Services commands allow you to specify either FILE or INFILE to refer to DD statements that specify a data set name, a volume serial, or both, to be used for processing the commands. If a DD statement referred to by either FILE or INFILE specifies both a data set name and a non-SMS-managed volume, and if the data set has been migrated by DFSMShsm, the Access Method Services command might not be performed successfully. If an error occurs for this condition, perform one of the following two actions before you reissue the Access Method Services command:

- If you need not specify the volume serial number on the DD statement, specify the DD statement without specifying the volume serial number and unit name. This specification causes the data set to be allocated when the job step is initiated, and the migrated data set is recalled for the allocation request.
- If you *must* specify the volume serial number on the DD statement [such as when you specify AMP = (AMORG) for VSAM data sets], recall the data set either:
  - Explicitly by issuing a DFSMShsm RECALL or HRECALL command
  - Implicitly by referring to the data set without specifying the volume serial number in a step that precedes the step where the Access Method Services command is being requested

GUARANTEED SPACE Attribute Considerations

The storage class attribute of GUARANTEED SPACE allows the volumes on which a data set resides to be chosen explicitly when that data set is created. If GUARANTEED SPACE is used and any of the volumes specified are not part of one of the storage groups to which the storage class maps, the creation of the data set fails. The use of GUARANTEED SPACE requires some special considerations if DFSMShsm is to manage data sets that are assigned to such a storage class.

When DFSMShsm is the data mover, it uses the Access Method Services functions of EXPORT and IMPORT to migrate and recall VSAM data sets. The IMPORT function redefines a VSAM data set by using a copy of the catalog information saved in the EXPORT copy of the data set. This define request explicitly specifies the volumes from which the data set was exported. The explicit specification of the volume affects the way data sets are imported with the GUARANTEED SPACE attribute. Likewise, for non-VSAM data sets, VTOC data set services uses the volume serial number passed by DFSMShsm if the storage class is GUARANTEED SPACE.

Results of Recalling GUARANTEED SPACE VSAM Data Sets

When DFSMShsm is the data mover, data sets recalled with the GUARANTEED SPACE attribute are normally placed on the volume from which they migrated. Consider the effects on a migrated, multiple-volume VSAM data set. A subsequent recall preserves the volume-to-component relationship that originally existed as long as the data set remains SMS-managed. For a data set in a GUARANTEED SPACE storage class, the relationship includes returning the data set to the same volumes from which it migrated. Thus, if a data set was originally defined with the data component on one volume and the index component on another, a migrate/recall sequence preserves the relationship between the components and the volumes. Similarly, if a key range data set has three key ranges on three separate volumes, the relationship is preserved.

If a volume is specified on the RECALL command, that volume is passed to IMPORT. When the IMPORT function redefines the VSAM data set, the specified volume overrides the volume or volumes from which the data set has been exported. If the specified volume is eligible, the data set is imported to that single volume. If there is not enough space, the recall fails. If the volume is not eligible, another volume may be selected. If the data set resides on multiple volumes when it is migrated, it is recalled to a single volume.

Possible Problems

In an environment in which the storage groups and storage classes do not change, the preceding results are acceptable for DFSMShsm to manage the data sets. However, you can expect problems in an environment that is subject to storage group reconfiguration or to changes in the mapping of data sets to storage classes. These problems may discourage the use of DFSMShsm space management for these data sets and may require special contingency procedures for availability management. The problems that occur relate to the following conditions:

- Changes in the storage group
- Changes in the storage class to data set mapping
- Conversions of data sets to GUARANTEED SPACE

Changes in the Storage Group

Consider the consequence of making a specific volume request in a storage group configuration that has changed. Assume that a data set with a GUARANTEED SPACE request has migrated. Some time later, one or more of the volumes on which the data set resided are removed from the storage groups that contain GUARANTEED SPACE data sets. A subsequent recall of the data set fails because at least one of the requested volumes is no longer part of a storage group for a GUARANTEED SPACE request. The result is the same for both single-volume data sets and multiple-volume VSAM data sets if the specific volume required is unavailable in the necessary storage group.

If a storage group is changed to remove volumes that data sets migrated from, you must perform one of the following actions:

- Change the ACS storage class routine so that a storage class with GUARANTEED SPACE is not assigned to the data set.
- Change the volume configuration again so that the storage group contains the volumes from which the data set migrated.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This is applicable only if the volumes are already SMS-managed.
- Use the RECALL command with the FORCENONSMS and the VOLUME parameters. This RECALL command causes the data set to be recalled to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

  Note: If the data set was multiple-volume when it migrated, it is recalled to a single volume.

Changes in the Storage-Class-to-Data Set Mapping

Consider the consequence of making a specific volume request in a storage class mapping that has changed. Assume that a data set is defined and the ACS routines have assigned it to a storage class that does not have the GUARANTEED SPACE attribute or does not assign it to any storage class. The ACS routine is then changed to assign the data set to a storage class with the GUARANTEED SPACE attribute. Also, the original storage class is changed to include the GUARANTEED SPACE attribute. Therefore, the recall might fail because the requested volumes are not in the storage groups that receive GUARANTEED SPACE requests.

If an SMS-managed data set's original storage class is changed to specify the GUARANTEED SPACE attribute before the data set is recalled, you must perform one of the following actions:

- Change the ACS storage class routine so that a storage class without GUARANTEED SPACE is assigned to the data set.
- Change the storage class of the data set to a storage class without GUARANTEED SPACE, without recalling the data set, by using the IDCAMS ALTER command.
- Change the volume configuration in the appropriate storage groups to contain the volumes from which the data set migrated. The volumes may need to be converted to SMS management first. It may be difficult to determine all of the volumes involved. DFSMShsm does not save all of the original volume serial numbers, only that of the volume on which the base data component resided.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This is applicable only if the volumes are already SMS-managed.
- Use the RECALL command with the FORCENONSMS and the VOLUME parameters. This RECALL command causes the data set to be recalled to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

  Note: If the data set was multiple-volume when it migrated, it is recalled to a single volume.

Conversions of Data Sets to GUARANTEED SPACE

If the data set was non-SMS-managed when it migrated and is assigned a storage class with GUARANTEED SPACE during recall, it is assigned to a volume in a storage group that contains GUARANTEED SPACE data sets.

If the data set was SMS-managed when it migrated but did not have GUARANTEED SPACE, the volumes from which it migrated might not be in a storage group that receives GUARANTEED SPACE requests. To avoid errors, you must perform one of the following actions before the recall:

- Change the target storage class with GUARANTEED SPACE to map to a storage group that contains the original volumes.
- Add the original volumes to a storage group mapped to by the target storage class with GUARANTEED SPACE.

DFSMShsm Retrieval of Management Classes and Storage Groups

If you are changing the definitions of management classes, you should know when the new definitions will take effect. DFSMShsm retrieves a list of all currently defined management classes before the following functions begin:

- Automatic volume space management, which includes primary space management and interval migration
- Automatic secondary space management
- Command volume space management

If one of the volume functions is in progress or has just completed, and another volume function is about to begin, DFSMShsm uses the list of management classes that exists from the prior processing. When secondary space management begins, a list of management classes is retrieved from the active SMS configuration and is not shared with the other functions in the preceding list.

When DFSMShsm begins automatic volume space management, it retrieves the storage group definitions to determine which SMS-managed volumes are eligible for space management.

A storage group must meet the following criteria for DFSMShsm to consider it for processing:

- It must be type POOL.
- It must have a status of either ENABLED, QUIESCED NEW, or DISABLED NEW.
- It must have AM=Y, AM=I, or AM=P.

Deletion of Dynamic I/O Device Considerations

Because DFSMShsm serializes the UCB of each ML1 volume as soon as the volume is selected for use, you will not be able to delete the device the ML1 volume resides on until processing is complete. The serialization of the UCBs occurs during:

- Automatic volume space management
- Automatic secondary space management
- Command volume migration
- Command level 1 to level 2 migration

Space Management Considerations for Sequential Extended-Format Data Sets

DFSMShsm does migration and recall processing of single-stripe and multiple-stripe sequential extended-format data sets, using DFSMSdss as the data mover. The data sets must be SMS-managed.

Volume Space Management Processing

Because sequential extended-format data sets span volumes, DFSMShsm migrates a data set only when the volume that contains the first stripe is processed. Migration is skipped for this data set when volumes that contain subsequent stripes are processed.

DFSMShsm expires an sequential extended-format data set only when the first stripe (volume) is being processed, regardless of whether the data mover is DFSMSdss or DFSMShsm. Expiration is skipped when subsequent stripes (volumes) are being processed.

You can command migrate single-stripe or multiple-stripe sequential extended-format data sets.

Automatic Primary Space Management Processing

DFSMShsm releases any allocated unused space only for a single-stripe sequential extended-format data set; allocated unused space will not be released for a multiple-stripe sequential extended-format data set.

Extent reduction will not be performed on either single-stripe or multiple-stripe sequential extended-format data sets.

If a sequential extended-format data set is small enough and meets all criteria—regardless of whether it is single-stripe or multiple-stripe—it can be migrated to a small-data-set-packing (SDSP) data set.

DFSMShsm space management for sequential extended-format data sets will not occur for any of the following reasons:

- A sequential extended-format data set is not migrated if it is very large (greater than 64K tracks) and cannot fit on one migration DASD volume. It can be migrated to tape volumes.
- A multistripe sequential extended-format data set is not migrated if SETSYS USERDATASETSERIALIZATION is not specified for DFSMShsm.
- A multistripe sequential extended-format data set is not migrated if it is protected by a RACF discrete profile.

Recall Processing

DFSMShsm recalls a sequential extended-format data set (indicated in the MCD record) using DFSMSdss as the data mover.

DFSMSdss recalls a sequential extended-format data set in the following ways:

- A sequential extended-format data set that was multiple-stripe when it was migrated could be recalled as a multiple-stripe or single-stripe sequential extended-format data set, depending on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.

- Conversely, a sequential extended-format data set that was single-stripe when it was migrated could be recalled as a multiple-stripe or single-stripe sequential extended-format data set, depending on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.
- A data set that was not sequential extended-format when it was migrated is always recalled as a data set that is not sequential extended-format. DFSMShsm does not convert a data set that is not sequential extended-format to a sequential extended-format data set through recall processing.
- A sequential extended-format data set can be recalled as a data set that is not sequential extended-format on a non-SMS-managed volume by specifying FORCENONSMS on the RECALL command.
- A data set that was sequential extended-format when migrated is recalled as a non sequential extended-format but still SMS-managed data set if no SMS volumes supporting sequential extended-format data sets are available.

Recall fails if the ACS filter routines are modified to direct a data set that was sequential extended-format when it was migrated to a non-SMS-managed volume.

Note: If reblocking is required for a sequential extended-format data set, the reblocking is performed.

Chapter 5. Space Management Procedures

This section contains information about:

- Interrupting and restarting automatic primary or secondary space management, or interval migration
- Defining new automatic primary or secondary space management process windows
- Allowing Another Host to Take Over Secondary Space Management Functions

Interrupting and Restarting Automatic Primary or Secondary Space Management, or Interval Migration

Automatic primary or secondary space management, or interval migration, can be interrupted during the course of their processing either intentionally or inadvertently. If you are not using the secondary host promotion function, when DFSMShsm restarts, it resumes its work near the point where it left off.

If you are in a multiple DFSMShsm-host environment, and are using the secondary host promotion function, a secondary host can take over the unique functions that were being performed by a primary host that has failed. Secondary host promotion also allows another host to take over the secondary space management functions from either a primary or secondary host that has failed.

This section describes:

- When automatic primary or secondary space management, or interval migration, needs to be restarted
- Automatic restart of automatic primary or secondary space management
- Automatic restart of interval migration
- Manual restart of automatic primary or secondary space management
- How to define a new automatic primary or secondary space management processing window

When Automatic Primary or Secondary Space Management or Interval Migration Needs to be Restarted

Table 12 presents a list of interruptions to the automatic primary or secondary space management functions or to interval migration. It also presents how to restart these functions.

Table 12. Restarting Automatic Primary or Secondary Space Management

Cause of Interruption	How to Restart
System Outage	IPL system, then start DFSMShsm (S DFSMSHSM)
Stop DFSMShsm (F DFSMSHSM,STOP)	Restart DFSMShsm (S DFSMSHSM)
Hold automatic volume space management HOLD MIGRATION[(AUTO)]	Release automatic volume space management RELEASE MIGRATION[(AUTO)]
Hold automatic secondary space management HOLD MIGRATION[(AUTO)]	Release automatic secondary space management RELEASE MIGRATION[(AUTO)]

© Copyright IBM Corp. 1984, 2004

Table 12. Restarting Automatic Primary or Secondary Space Management (continued)

Cause of Interruption	How to Restart
Cancel DFSMShsm	Start (S DFSMSHSM)
SETSYS EMERGENCY	SETSYS NOEMERGENCY
Migration held because output media is not available	ADDVOL, then RELEASE MIGRATION[(AUTO)]

Automatic Restart of Automatic Primary or Secondary Space Management

If you are not using secondary host promotion, and there is an interruption caused by a system outage, a cancel of DFSMShsm, or a stop of DFSMShsm, the start of DFSMShsm causes DFSMShsm to attempt to restart automatic primary or secondary space management. Automatic primary or secondary space management automatically restarts if all of the following conditions exist:

- The time of the DFSMShsm restart is in the automatic primary or secondary space management processing window and the planned start time has not been changed
- Automatic primary or secondary space management is not held
- DFSMShsm is not in EMERGENCY mode
- The operator gives permission if the SETSYS REQUEST command asks permission to restart

Otherwise, if the planned start times have not been changed, you can manually restart automatic primary or secondary space management.

If you are in a multiple DFSMShsm-host environment and are using secondary host promotion, and the primary DFSMShsm host or the DFSMShsm host that runs secondary space management fails or goes into emergency mode, another DFSMShsm host is promoted to take over the unique functions of the failed DFSMShsm host. The unique functions performed by the primary DFSMShsm host include:

- Hourly space check, needed for interval migration and recall placement for non-SMS data
- Automatic CDS backup
- Automatic movement of backup versions from ML1 to tape
- Automatic backup of migrated data sets
- Expiration of dump copies
- Deletion of excess dump VTOC copy data sets

When the original primary DFSMShsm host or secondary space management DFSMShsm host restarts, and is not in emergency mode, it takes over those unique functions once more. However, if the promoted host is in the process of performing some of the functions it took over for the original host, the promoted host continues with those functions to completion. The original host does not attempt to perform its unique functions until the beginning of the next window.

By default, if you manually shut down the primary DFSMShsm host or secondary space management DFSMShsm host, secondary host promotion is not invoked, with the exception of the STOP DUMP command. You can override the default and

cause secondary host promotion to be invoked when you manually shut down a primary DFSMShsm host or secondary space management DFSMShsm host by using the STOP PROMOTE command.

Automatic Restart of Interval Migration

If the interruption was a system outage, a cancel of DFSMShsm, or a stop of DFSMShsm, the start of DFSMShsm causes DFSMShsm to check space on primary volumes eligible for automatic space management. If any such volume is over its threshold, DFSMShsm then restarts interval migration if all of the following conditions exist:

- Automatic primary or secondary space management is not held.
- DFSMShsm is not in emergency mode.
- The operator gives permission if the SETSYS REQUEST command asks permission to restart.

Manual Restart of Automatic Primary or Secondary Space Management

You can restart automatic primary or secondary space management from where it was interrupted if it did not complete and if the current time is still within the processing window. The planned start time is when automatic primary or secondary space management last planned to start from the beginning.

When the Current Time Is in the Processing Window

If automatic primary or secondary space management does not restart automatically and the current time is still in the processing window, use one of the following commands to restart automatic primary or secondary space management:

- If automatic primary or secondary space management was held, issue the RELEASE MIGRATION(AUTO) command.
- If DFSMShsm was in emergency mode, issue the SETSYS NOEMERGENCY command.

When the Current Time Is Not in the Start Window

Automatic Primary Space Management: If the current time is after the ending time, redefine an expanded processing window using the SETSYS PRIMARYSPMGMTSTART(*old-planned,new-end*) command. In the command, *old-planned* is the previous planned start time; *new-end* is a value greater than the current time. This command causes automatic primary space management to restart. If you enter a value for a planned start time that is earlier than the previous planned start time, DFSMShsm also considers the request to be for an expansion of the existing processing window.

If you enter a value for the planned start time that is later than the previous planned start time, DFSMShsm considers the request to be for a new processing window, and the restart is not possible. When automatic primary space management next runs, it will start from the beginning.

Automatic Secondary Space Management: If the current time is after the ending time, redefine an expanded processing window using the SETSYS SECONDARYSPMGMTSTART(*old-planned,new-end*) command. In the command, *old-planned* is the previous planned start time; *new-end* is a value greater than the current time. This command causes automatic secondary space management to restart.

If you enter a value for the planned start time that is different than the previous planned start time, DFSMShsm considers the request to be for a new processing window, and a resumption from the point of interruption is not done. When automatic secondary space management next runs, it will start from the beginning.

Where Automatic Primary or Secondary Space Management Restarts

Automatic Primary Space Management: DFSMShsm records which volumes were processed during automatic primary space management, so it will not attempt to process them again when it is restarted.

Automatic Secondary Space Management: If your installation is not using secondary host promotion, and automatic secondary space management functions restart (without changing the *planned-start time*), they resume their work at approximately the point of interruption. They complete the work that would have run if the interruption had not occurred, or they continue until the ending time has been reached. If your installation is using secondary host promotion, and the promotion occurs during a secondary space management window, and the window is not past the latest time to start (as defined on the primary host), then the promoted host takes over from approximately where the original host left off.

Note: Automatic secondary space management consists of two level functions, migration level cleanup and level 1 to level 2 migration. A restart occurs in whichever was interrupted.

Defining a New Automatic Primary or Secondary Space Management Processing Window

As you increase the number of volumes that DFSMShsm space manages, you may want to change the automatic primary or secondary space management processing window or you may want to add more tasks. This section describes how to define a new processing window. For more information about adding more tasks, see "Specifying the Maximum Automatic Primary Space Management Tasks" on page 35 and "Specifying the Maximum Automatic Secondary Space Management Tasks" on page 35.

You can define a new processing window whether automatic primary or secondary space management is running or not.

When Automatic Primary or Secondary Space Management Is Running

You can change the automatic primary or secondary space management processing window at any time, even when automatic primary or secondary space management is currently running on a particular DFSMShsm host. Automatic primary or secondary space management will run when the new planned start time occurs if the currently running automatic primary or secondary space management completes before the planned start time of your newly defined processing window. If the currently running automatic primary or secondary space management runs past the new planned start time, automatic primary or secondary space management will not start automatically until the next time the planned start time occurs.

When Automatic Primary or Secondary Space Management Is Not Running

If you define a planned start time that is later than the time that automatic primary or secondary space management last completed, automatic primary or secondary space management starts at the planned start time.

If you define a planned start time that is earlier than the time that automatic primary or secondary space management last completed, automatic primary or secondary space management does not start until the next time the planned start time occurs on a scheduled day.

Allowing Another Host to Take Over Secondary Space Management Functions

In a sysplex, you can enable another host to take over secondary space management functions for either a primary or a secondary host that has failed. This function, called secondary host promotion, ensures continuous availability of DFSMShsm functions.

Related reading: For additional information about secondary host promotion key definitions and how to enable secondary host promotion from the SETSYS command, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Part 3. How to Control Availability Management

Availability management is the function of the DFSMShsm program that you use to ensure that your customers can retrieve usable copies of their data sets should their online copies become lost or damaged. You can make the copies available by:

- Making daily incremental backup copies of changed data sets.
- Making periodic dump copies of the DFSMShsm-managed and ML1 volumes.
- Making aggregate backup copies of data sets that your operation will need if your installation is damaged.
- Making fast replication backup versions for sets of storage groups.

Functions of Availability Management: Availability management makes it possible for you to:

- Automatically make backup copies of individual changed data sets on DFSMShsm-managed volumes. This is known as incremental backup. You can specify how frequently to back up data sets on a data set basis for SMS-managed data sets and on a system-wide basis for non-SMS-managed volumes. You can specify on a system-wide basis how often backup runs.
- Automatically make dump copies of DFSMShsm-managed level 0 volumes and ML1 volumes on a specified schedule for day and time. You can dump different groups of volumes on different days with different periods for the number of days between dumps.
- Allow your customers to issue commands to recover their own data sets.
- Allow your customer to issue the HBACKDS command from a batch environment. See “Inline Backup” on page 415.
- By command, back up user data sets.
- By command, back up data sets of an application to tape so they can be taken to another computer site for recovery.
- By command, recover user data sets or data sets of an application to their original system environment at another computer site.
- By command, restore a volume from a dump copy and update the restored volume from later incremental backup versions.
- By command, recover a specific data set from either a dump copy or an incremental backup version.
- By command, create a fast replication backup of a copy pool.
- By command, recover volumes that have a fast replication backup version.

Backup Function: Backup is the process of copying a data set from a level 0 or an ML1 volume to daily backup volume. This copy is called a backup version. The purpose of backup is to have copies of data sets in case something happens to the original data sets. The difference between dump and backup is that the dump function backs up the entire allocated space on a volume, whereas the DFSMShsm backup function backs up individual data sets.

DFSMShsm can create backup versions of data sets either automatically or by command. DFSMShsm automatically creates backup versions of data sets on specified days beginning at a specified time of day. The data sets must meet eligibility criteria and must be on DFSMShsm-managed volumes that have been designated for automatic backup.

© Copyright IBM Corp. 1984, 2004

Fast Replication Function: Fast replication uses volume-level fast replication to create backup versions for sets of storage groups. You define a set of storage groups through the SMS 'copy pool' construct. The DFSMShsm FRBACKUP command creates a fast replication backup version for each volume in every storage group 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 small time frame. The term fast replication refers to the FlashCopy function supported by IBM TotalStorage Enterprise Storage Server (ESS) disk and the SnapShot function supported by IBM RAMAC Virtual Array (RVA) disk.

Dump Function: Dump is the process of copying all data from a DASD volume to dump tape volumes. Full-volume dump is an extension of DFSMShsm's availability management that invokes DFSMSdss through the DFSMSdss application interface. Full-volume dump backs up the entire allocated space of DFSMShsm-managed DASD volumes and ML1 volumes either automatically or by command. Non-DFSMShsm-managed volumes are dumped only by command. The purpose of the full-volume dump is to expedite the recovery process when an entire volume is lost or damaged and to supplement the incremental recovery process.

The full-volume-dump process can be one volume in and one dump copy out. It can also be one volume in and multiple dump copies out. Each dump copy is a complete image of the dumped volume, not just an incremental backup of selected data sets. Each successive full-volume dump of a volume, regardless of the number of dump copies, is a generation. Each dump copy in a generation is associated with a different dump class, which specifies how the dump copy is to be managed. DFSMShsm allows from one to five dump copies to be made concurrently for any one full-volume dump. DFSMSdss discontinues writing to dump copies on which errors occur. DFSMSdss creates the multiple copies and continues its full-volume-dump process as long as one output copy is good. DFSMShsm discards the contents of only the bad copies and issues an appropriate message. If all copies fail, the full-volume dump is failed.

DFSMShsm keeps 100 generations of dump copies for any given volume unless all the copies in a generation expire. When all dump copies in any generation for a volume reach their expiration dates, DFSMShsm deletes that generation from its records. When the 100 generations for a particular volume have been reached and the next full-volume dump for that volume is performed, the control records for the oldest generation are discarded, regardless of the retention periods of the individual copies.

There are special considerations when you are dumping a volume containing data sets with RLS attributes. For more information, refer to *z/OS DFSMSdss Storage Administration Guide*.

Types of Backup Volumes: You can define the following types of tape or DASD backup volumes to DFSMShsm:

- **Daily backup volumes:** Daily backup volumes are assigned to a specific day in the backup cycle and contain the backup versions created on that day. DFSMShsm creates these backup versions during the backup of a level 0 volume or during the backup of migrated data sets. Daily backup volumes can also contain backup versions created by the BACKDS command. However, the daily backup volume used to contain these manually backed up versions does not necessarily correspond to the day in the backup cycle when DFSMShsm created

the backup version. Instead, the daily backup volume corresponds to the day in the backup cycle that DFSMShsm moved the backup version from its temporary location on an ML1 volume to a daily backup volume.

- **Fast replication target volumes:** Fast replication target volumes contain the fast replication backup copies of DFSMShsm-managed volumes. Fast replication target volumes are defined with the SMS copy pool backup storage group type.
- **Spill backup volumes:** Spill backup volumes contain older backup versions of data sets. These backup versions are moved to spill backup volumes when DFSMShsm needs more space on DASD daily backup volumes. Tape spill volumes also receive all valid backup versions of data sets when a tape backup volume is recycled.
- **Unassigned backup volumes:** DFSMShsm uses unassigned volumes whenever it needs more daily or spill backup volumes. There are two types of unassigned volumes: unassigned daily backup volumes and unassigned backup volumes. When DFSMShsm selects an unassigned volume during backup processing, DFSMShsm assigns it as a daily or spill backup volume.
  - Unassigned daily backup volumes are those volumes defined to DFSMShsm as daily backup volumes but not associated with any day in the backup cycle. DFSMShsm can use these volumes only as daily backup volumes, but DFSMShsm determines which day in the backup cycle the volume should be assigned to when it first uses the volume.
  - Unassigned backup volumes are those volumes defined to DFSMShsm as backup volumes but not specified as daily or spill backup volumes. When it first uses unassigned backup volumes, DFSMShsm assigns the volumes as daily or spill backup volumes. If DFSMShsm uses a volume as a daily backup volume, DFSMShsm determines which day in the backup cycle the volume should be assigned to.

  When DFSMShsm needs another daily backup volume, it chooses an unassigned daily backup volume before it chooses an unassigned backup volume.

Recovery and Restore of Data Sets: Recovery and restore are processes that are requested only by command, not automatically, for backed up data sets. Recovery is the process of retrieving a full-volume dump or a backup version of a data set or a volume. Restore signifies that DFSMSdss is used to retrieve dumped data from dump volumes. You can use restore or recover processing to:

- Recover a data set that has been lost or damaged.
- Recover an earlier version of the data set without deleting the current version.
- Restore a volume from a full-volume dump and update the volume from later incremental backup versions.
- Restore a data set from a dump copy.
- Restore a volume from a full-volume dump.
- Recover a volume from DFSMShsm backup versions.
- Recover a single volume or a pool of volumes from the DFSMShsm-managed fast replication backup versions.

DFSMShsm volume recovery can use incremental backups or full-volume dumps or both. One DFSMShsm RECOVER command can be used to request both a volume restore and an incremental volume recovery.

DFSMShsm can recover data sets from a DFSMShsm incremental backup version or from a DFSMShsm dump copy. DFSMShsm automatically chooses the more recent copy of the data set if you allow users to perform data set restores (see "Controlling Restoring of Individual Data Sets from Dump Tapes" on page 157).

There are special considerations for full volume restores of volumes that were dumped using the BACKVOL DUMP command and whose data sets contain RLS attributes.

Related reading: For additional information about these special considerations, see the *z/OS DFSMSdss Storage Administration Guide*.

Organization of This Part: Because DFSMShsm availability management for SMS-managed storage is controlled at a different level than is non-SMS-managed storage, our descriptions of managing the two different groups of storage are separated.

This part is divided into the following chapters.

Task	See Page
Chapter 6, "Availability Management of SMS-Managed Storage"	151
Chapter 7, "Availability Management of Non-SMS-Managed Storage"	261
Chapter 8, "Aggregate Backup and Recovery Support (ABARS)"	281
Chapter 9, "Making Disaster Backup Copies of DFSMShsm-Owned Tape Volumes"	401
Chapter 10, "Other Considerations for Availability Management"	413
Chapter 11, "Availability Management Procedures"	437

Chapter 6. Availability Management of SMS-Managed Storage

As described in "Overview of DFSMShsm Functions" on page 5, DFSMShsm manages SMS-managed storage on a data set basis. That is, the specification for how to treat each data set on a volume is contained in the management class for that data set. The ACS routine can determine the storage group to which a data set is recovered.

This chapter contains the following information:

- What the example system shows
- The tasks necessary to do availability management for SMS-managed storage
- Dump tasks

What the Example System Shows

In the example system (see Figure 4 on page 17) all of the availability management functions are demonstrated. The automatic and command functions described in this chapter are also performed for non-SMS-managed storage. The example system shows the following three functions:

Related reading: For additional information about the controls for non-SMS-managed storage, see Chapter 7, "Availability Management of Non-SMS-Managed Storage," on page 261.

Automatic Incremental Backup

On every day except Sunday in the example system, all changed data sets except those in certain management classes are backed up. Backup starts between 3 a.m. and 4 a.m. with no new volumes started after 5 a.m. Backup is to tape. See Figure 31 on page 177 for the complete set of commands that DFSMShsm uses to process the incremental backup.

Host 1 backs up the control data sets at the beginning of backup (see Chapter 12, "Maintaining DFSMShsm Control Data Sets," on page 447). It also moves data sets that have been backed up by command from ML1 volumes to backup volumes and backs up migrated data sets that are in need of backup (see "Determining Eligible Data Sets for Backup" on page 53).

Automatic Full-Volume Dump

In the example system a full-volume dump is performed once a week on Sunday, starting between 3 a.m. and 4 a.m. and starting no new volumes after 5 a.m. All DFSMShsm-managed volumes are dumped every week. In addition, volumes in storage groups 1 and 2 are dumped to an extra copy for off-site storage every 14 days; volumes in storage groups 3 and 4 are dumped to an extra copy for off-site storage every 28 days. See Figure 25 on page 162 for the complete set of commands that DFSMShsm uses to process a full-volume dump.

Command Availability Management

In the example system, commands cause availability management functions to occur creating the following conditions:

- One data set to be backed up.

© Copyright IBM Corp. 1984, 2004

- All the changed data sets on a volume to be backed up.
- A volume to be dumped.
- A backed-up data set to be recovered.
- A volume to be restored from a dump and forward recovered from later incremental backup versions. *Forward recovery* is a process of updating a restored volume by applying later changes as indicated by the catalog and the incremental backup versions.
- A volume to be restored from a dump copy.
- A volume to be recovered from backup versions.
- A specific data set to be restored from a dump volume.
- All expired backup versions to be deleted.
- A fast replication backup version to be recovered from a copy pool.

Tasks for Automatic Availability Management

Now that the functions that the example system can accomplish have been described, the means to make the example system accomplish these functions are described. The tasks for controlling automatic availability management of SMS-managed storage require adding DFSMShsm commands to the ARCCMD*xx* member and specifying attributes in the SMS storage classes and management classes. It is assumed that the storage classes and management classes have already been defined. The attribute descriptions explain the attributes to be added to the previously defined storage groups and management classes. Similarly, the descriptions of DFSMShsm commands relate to commands to be added to the ARCCMD*xx* member of SYS1.PARMLIB.

Two groups of tasks are performed for availability management: dump tasks and backup tasks. Each of these groups of tasks is described separately.

Dump Tasks

The dump tasks that are described consist of:
- Specifying which volumes to dump and the dump classes to use
- Specifying when automatic dump processing starts
- Specifying the DFSMSdss DASD I/O buffering technique to use for dump
- Specifying the maximum number of dump tasks
- Specifying the days on which dump occurs
- Specifying the characteristics of dump classes
- Defining dump volumes to DFSMShsm

Specifying Which Volumes to Dump and the Dump Classes to Use

DFSMShsm must be told what volumes to dump and what dump classes to use for the volumes to be dumped. This specification comes from the storage group.

The AUTO DUMP attribute specifies whether DFSMShsm is to perform automatic dumps on the volumes assigned to the storage group. In the example system, the AUTO DUMP attribute for all storage groups is AUTO DUMP = Y.

The DUMP CLASS attribute determines two controls for DFSMShsm: (1) the dump classes to which the volumes associated with the storage group are dumped and (2) how many dump copies are made when the dump occurs. The number of dump classes specified for any storage group determines the *maximum* number of copies that are made for any one dump. You can specify up to five dump classes

for each storage group. Parameters in the dump class determine whether a dump copy is made for that particular dump class on that particular day.

The space management example (Figure 5 on page 24) shows how host 1 can be limited to doing space management only for storage group 1. Dump for any of the storage groups is not being limited to a single host. However, should you want to limit the processing to a single host, you can use the DUMP SYSTEM/SYSPLEX NAME attributes in the same way as the MIGRATE SYSTEM/SYSPLEX NAME attribute. Figure 24 shows the dump management attributes that have been added to the storage groups in the example system.

Related reading: For additional information about storage groups and their function in SMS, see the *z/OS DFSMSdfp Storage Administration Reference*.

```
NAME ===> SG1
  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> TWOWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG2
  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> TWOWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG3
  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> FOURWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG4
  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> FOURWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>
```

Figure 24. Storage Group Dump Attributes for Availability Management

Specifying When Automatic Dump Processing Starts

In the example system, automatic dump processing is started between 3 a.m. and 4 a.m. and no new volumes start after 5 a.m. The AUTODUMPSTART parameter of the SETSYS command controls the start of automatic dump processing. In the example system, the following SETSYS command is added to the ARCCMD*xx* member in each of the hosts:

```
SETSYS AUTODUMPSTART(0300 0400 0500)
```

Before automatic dump can start, you must have specified that backup be enabled in that particular host.

Related reading: For additional information about how and when to run automatic backup, see "Specifying Whether to Perform Volume Backup and the Kind of Device to Use" on page 173.

Specifying the DFSMSdss DASD I/O Buffering Technique to Use for Dump

With a single START I/O instruction, DFSMSdss can read 1, 2, or 5 tracks at a time, or a complete cylinder from the DASD being dumped. The DUMPIO parameter of the SETSYS command identifies which DFSMSdss DASD I/O buffering technique to use and the number of tracks to read.

The command is:

```
SETSYS DUMPIO(n,m)
```

where *n* indicates the DFSMSdss DASD I/O buffering technique for physical volume dump (using the BACKVOL command with the DUMP parameter or during automatic dump), and *m* indicates the value used for DFSMSdss logical dump (DFSMSdss specified as data mover on the SETSYS command).

The values used for *n* and *m* are:

Value	Meaning
1	DFSMSdss reads 1 track at a time
2	DFSMSdss reads 2 tracks at a time
3	DFSMSdss reads 5 tracks at a time
4	DFSMSdss reads 1 cylinder at a time

Note: If you specify DUMPIO without a value for *m*, then *m* defaults to the value specified for *n*.

If you do not specify the DUMPIO parameter on any SETSYS command, the DFSMShsm default for *n* is 1 and for *m* is 4.

In the example system, a buffer of 5 tracks is used, so the SETSYS command that is added to the ARCCMD*xx* member in each of the hosts is:

```
SETSYS DUMPIO(3,1)
```

Specifying Concurrent Copy for Volume Dumps

The VOLUMEDUMP parameter of the SETSYS command must be set if you want to use concurrent copy for volume dumps. To instruct DFSMShsm to request that DFSMSdss use concurrent copy, issue the following command:

```
SETSYS VOLUMEDUMP(CC)
```

If you are presently using concurrent copy for volume dumps and want to disable it, issue the following command:

```
SETSYS VOLUMEDUMP(NOCC)
```

The default of VOLUMEDUMP is NOCC.

Specifying the Maximum Number of Dump Tasks

DFSMShsm can run up to 32 dump tasks concurrently in each host. Because DFSMShsm can make up to five dump copies concurrently for each dump task, the number of dump tasks does not limit the number of dump copies. In deciding how many dump tasks to specify, consider how many tape units are available to receive the dump copies.

The MAXDUMPTASKS parameter of the SETSYS command controls how many dump tasks run concurrently. In the example system the use of three concurrent dump tasks is demonstrated. The following SETSYS command is added to the ARCCMD*xx* member in each host in the example system:

```
SETSYS MAXDUMPTASKS(3)
```

Because it has been specified that each storage group have two dump classes, six tape units must be available for each host on the days when both dump classes are scheduled.

If any dump class involved specifies a STACK value of more than one, DFSMShsm writes multiple dump copies sequentially as each tape is mounted for that dump class. That can reduce the number of tape drives (the number of dump tasks) needed concurrently, but the reduced parallelism can lengthen the total time needed for the automatic dump window specified by SETSYS AUTODUMPSTART.

Specifying the Day on Which a Dump Occurs

The DUMPCYCLE parameter of the DEFINE command specifies when the host does dump processing. You can also specify the day to start the cycle so that the cycle does not change with each reinitialization of DFSMShsm. In the example system, the following command is added to the ARCCMD*xx* member in each host:

```
DEFINE DUMPCYCLE(NNNNNNYNNNNNNYNNNNNNYNNNNNNY -
CYCLESTARTDATE(06/01/03))
```

The command establishes a 28-day cycle that begins on a Monday. Automatic dump processing occurs on Sunday. DFSMShsm uses the dump cycle to determine the days on which automatic dump occurs. DFSMShsm performs automatic dump only on days designated with a *Y* in the dump cycle. The dump cycle that you define applies to *all dump classes that you define.* Thus, the preceding example dump cycle applies to all of the dump classes that are defined in the following examples.

DFSMShsm stores the date you specify with CYCLESTARTDATE as the date the dump cycle began. If you do not specify a CYCLESTARTDATE parameter with the DEFINE command but a cycle start date is already stored and the dump cycle length has not changed, DFSMShsm uses the cycle start date that is already stored. If no cycle start date is stored and you do not specify one, DFSMShsm uses the day that the DEFINE DUMPCYCLE command is issued as the cycle start date.

Rule: 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.

Notes:

1. When you redefine the dump class, be careful if you use the DAY parameter to define the days that a given dump class is active. The day that is specified with

the DAY parameter should 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.

2. Dump cleanups run on *N* days in the dump cycle. Cleanup functions include expiration of expired dump copies and deletion of excess dump VTOC copy data sets.

Specifying the Characteristics of Dump Classes

The dump classes to which a volume is dumped determine how the particular dump copies are made and used. You can control the following factors with the dump class:

- Whether the dump tapes are automatically reused when the dumped data is no longer valid
- Whether the dump tapes can be used to restore individual data sets
- The day of the dump cycle on which data will be dumped to the dump class
- What to do with the newly created dump tapes
- How often to dump data to the dump class
- Whether to reset the data-set-changed indicator
- When the data on the dump volumes becomes invalid
- The kind of tape unit to use for the dump tapes
- The number of generations for which copies of the VTOC of dumped volumes will be kept
- The expiration date to use in the tape header label
- The maximum number of dump copies to be stacked on a dump tape assigned to this dump class

The DUMPCLASS parameter of the DEFINE command provides the control for the dump class.

There are three dump classes in the example system: ONEWEEK, TWOWEEK, and FOURWEEK. The copies in the ONEWEEK class are kept at the computer site and are used to restore complete volumes or individual data sets as necessary. The copies in the TWOWEEK and FOURWEEK classes are sent to another site to be available for returning the business to production should a disaster occur at the computer site. Because the DUMPCLASS parameter has many subparameters, the effect of each subparameter is described and the DUMPCLASS parameter is built as you proceed.

Controlling Automatic Reuse of Dump Tapes

The AUTOREUSE subparameter of the DUMPCLASS parameter controls whether dump tapes that have a retention period specified for them in the dump class are reused when the data expires. AUTOREUSE specifies that the tapes are returned to the DFSMShsm tape pool when the dump copy is invalidated. NOAUTOREUSE specifies that the tapes are marked as unavailable for selection by DFSMShsm when the dump copy is invalidated.

Because the volumes in dump class ONEWEEK are retained at the computer site, they can be reused when the data on them is invalidated. Therefore, begin building the DEFINE DUMPCLASS command to be added to the ARCCMD*xx* member in each host for this dump class as:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE)
```

Because the volumes in dump classes TWOWEEK and FOURWEEK are sent to another site, they cannot be automatically returned to use by the system. Therefore, add the following DEFINE DUMPCLASS commands to the ARCCMD*xx* member in each host for these dump classes:

```
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE)
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE)
```

The AUTOREUSE subparameter is really of use only when you specify that empty tapes be returned to a DFSMShsm pool of empty tapes that are known to DFSMShsm. If tapes are returned to the scratch pool and if you are not selecting specific tapes for subsequent output tape selection, this subparameter is of no use because it makes tapes reusable only if they are known to DFSMShsm.

In the example system, scratch tapes are used and empty tapes are released to the scratch pool. Therefore, although the specification of AUTOREUSE for dump class ONEWEEK was shown, it has no effect because volumes are not being defined to DFSMShsm. However, the reason for specifying NOAUTOREUSE in dump classes TWOWEEK and FOURWEEK is valid. NOAUTOREUSE must be specified for tapes that are taken away from the computer system site.

Controlling Restoring of Individual Data Sets from Dump Tapes

You can control whether DFSMShsm can use copies of data sets on dump tapes to recover individual data sets. If you allow restoring of individual data sets from dump copies in a particular dump class, DFSMShsm selects the latest copy of the data set, whether it resides on a dump volume or a backup volume. If you do not allow restoring of individual data sets from dump copies in a particular dump class, DFSMShsm cannot select the data set from a dump copy in that class.

Because it is intended to use the later of the copies on either the dump volumes in dump class ONEWEEK or the backup volumes, change the DUMPCLASS parameter to the following in the ARCCMD*xx* member in each host:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE DATASETRESTORE)
```

Because the volumes in dump classes TWOWEEK and FOURWEEK will be sent to another site, you do not want to choose them for restoring data sets. Therefore, change the DUMPCLASS parameter for these two classes in the ARCCMD*xx* member in each host by issuing the following commands:

```
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE)
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE)
```

Specifying the Day to Dump Data to This Dump Class

You can specify that you want data dumped to a particular dump class on a particular day in the dump cycle with the DAY subparameter. If you specify the DAY subparameter, the dump class is used only on that day. The day specified with the DAY subparameter must be a Y day in the dump cycle that you have defined with the DEFINE DUMPCYCLE command.

Because you want dump class ONEWEEK to be used four times during the dump cycle and you want dump class TWOWEEK to be used two times, do not specify the DAY subparameter for these two classes. The DAY subparameter does not

allow you to use a dump class more than one time in a dump cycle. However, as dump class FOURWEEK is used once every 28 days, you can change its DEFINE DUMPCLASS command as follows:

```
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28))
```

Specifying How Often to Dump Data to the Dump Classes

You can control how often volumes are dumped to a dump class by using the FREQUENCY subparameter. This subparameter of the DUMPCLASS parameter works with the DUMPCYCLE parameter to determine when volumes are dumped. For a volume to be dumped, the dump must occur on a Y day in the dump cycle and the number of days specified with the FREQUENCY subparameter must have passed. Use the FREQUENCY subparameter, rather than the DAY subparameter, when you want to use the dump class more than one time in the dump cycle.

In the example system, you want the volumes being dumped in dump class ONEWEEK to be dumped every seven days and the volumes in dump class TWOWEEK to be dumped every 14 days. (You also want the volumes being dumped to dump class FOURWEEK to be dumped every 28 days, but have taken care of this requirement with the DAYS subparameter.) To control dumping to dump classes ONEWEEK and TWOWEEK, you can change the DEFINE DUMPCLASS commands as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
DATASETRESTORE FREQUENCY(7))
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14))
```

Specifying What to Do with the Dump Volumes

As each DASD volume is dumped, DFSMShsm notifies the operator that the volume is dumped and to which tape volumes that it is dumped. By using the DISPOSITION subparameter of the DUMPCLASS parameter, you can add to this message instructions about what to do with the dump volumes.

Because you are using the volumes from dump class ONEWEEK at the computer site, you have no particular instructions for the operator concerning these volumes. Therefore, do not add the DISPOSITION subparameter to the DEFINE DUMPCLASS command for this dump class. However, because you are sending the volumes from dump classes TWOWEEK and FOURWEEK to another site, you can change the DEFINE DUMPCLASS command for these dump classes as follows:

```
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14) -
DISPOSITION('SEND OFFSITE BIWEEKLY'))
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28) -
DISPOSITION('SEND OFFSITE MONTHLY'))
```

Specifying Whether to Reset the Data-Set-Changed Indicator

You can specify whether to reset the data-set-changed indicator on changed data sets when dump processing is performed to a specific dump class. If the dump operation resets the data-set-changed indicator, the data sets are not backed up by the next incremental backup operation unless the data set is changed after the dump occurs. If volumes are being dumped to multiple dump classes and you indicate in any of the dump classes that the indicator is to be reset, the indicator is

reset whenever volumes are dumped to that dump class. If the dump copy is the most-recent backup copy of a data set and the data-set-changed indicator is reset, the most-recent backup copy can be lost if the dump copy expires automatically.

The RESET subparameter of the DUMPCLASS parameter controls whether the data-set-changed indicator is reset. If you intend to use incremental backup copies to update volumes restored from dump copies, you should use the NORESET option of this subparameter. You are going to use the incremental backup copies to update restored volumes; therefore, you can change the DEFINE DUMPCLASS command as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
DATASETRESTORE FREQUENCY(7) NORESET)
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14) -
DISPOSITION('SEND OFFSITE BIWEEKLY') -
NORESET)
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28) -
DISPOSITION('SEND OFFSITE MONTHLY') -
NORESET)
```

Specifying When the Data on Dump Volumes Becomes Invalid

The data on dump volumes is of value until the data on the original volume has changed significantly. In the meantime, you are making new dump copies periodically. After some period, you want to indicate to DFSMShsm that the old dump copies are no longer of value. You can do this with the RETENTIONPERIOD subparameter of the DUMPCLASS parameter.

Use the RETENTIONPERIOD subparameter to specify a number of days for which DFSMShsm considers data on the dump volumes as valid. After the period specified by RETENTIONPERIOD has elapsed, the volumes become eligible to be reused, either automatically or manually.

Because you are making dump copies weekly for use at your site, consider all data on volumes dumped to dump class ONEWEEK to be invalid after three weeks. Similarly, consider data on volumes dumped to dump classes TWOWEEK and FOURWEEK to be invalid after three cycles. If you want to retain the dump copies until the full 100 generations have been made, you can specify NOLIMIT as the value for the retention period. Change the DEFINE DUMPCLASS commands as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
DATASETRESTORE FREQUENCY(7) NORESET -
RETENTIONPERIOD(21))
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14) -
DISPOSITION('SEND OFFSITE BIWEEKLY') -
NORESET RETENTIONPERIOD(42))
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28) -
DISPOSITION('SEND OFFSITE MONTHLY') -
NORESET RETENTIONPERIOD(84))
```

Specifying a Tape Expiration Date

Some tape management programs require a tape expiration date in the tape header label. If you specify EXPIRATION or EXPIRATIONINCLUDE as tape security parameters, DFSMShsm places a tape expiration date of 99365 in the tape header label. You use the TAPEEXPIRATIONDATE subparameter to cause DFSMShsm to

use a unique expiration date for the volumes rather than 99365. The example system does not use a tape expiration date.

Specifying the Kind of Tape Units to Use for Dump Volumes

The UNIT subparameter of the DUMPCLASS parameter limits the kind of tape unit to use for dump volumes when the dump function is not using a system-managed library.

Related reading: For additional information about the UNIT subparameter of the DUMPCLASS parameter, see the *z/OS DFSMShsm Storage Administration Reference*.

Specifying the Maximum Number of Dump Copies to Stack on a Dump Volume

The STACK subparameter of the DUMPCLASS parameter directs DFSMShsm to store one or more dump copies (each in its own file) on a dump tape, while that tape is initially mounted after selection during automatic dump.

Although the STACK value is intended to help you make better use of the newer, higher-capacity tape cartridges, DFSMShsm does not limit its use to certain unit types, and does not reduce the amount of stacking to prevent dump copies from spanning to additional dump tapes. However, if a dump class specifies a STACK value of more than one, and that value results in a given sequence of dump copies spanning to one or more additional dump tapes, DFSMShsm writes a message alerting you to that fact; you may want to reduce the STACK value.

Since the number of volumes to be dumped is not likely to be an integral multiple of the STACK value, the last dump tape assigned to a dump class with STACK value of M is likely to contain fewer than M dump copies.

DFSMShsm does not try to extend existing dump volumes during initial selection, but selects only empty volumes as targets of full-volume dumps. Therefore, after a dump tape is dismounted, it can be taken offsite or whatever action is specified by the dump class DISPOSITION parameter.

All the dump copies stacked on a dump tape expire on the same day, but some copies may become invalid (rolled off) before that time.

Specifying the Number of VTOC Copies to Keep

As DFSMShsm makes a generation of dump copies, it makes a copy of the VTOC of the level 0 volume being dumped for that generation and associates it with the dump copies that are made. DFSMShsm uses the VTOC copy when a dump copy is used to restore a data set. You can specify how many copies of the VTOC are kept. DFSMShsm does not create a VTOC copy when an ML1 volume is dumped.

The example system demonstrates how VTOC copies are kept. It has been stated that storage groups 1 and 2 are using dump classes ONEWEEK and TWOWEEK and that storage groups 3 and 4 are using ONEWEEK and FOURWEEK. Every seven days a new generation is made for all volumes with a copy for dump class ONEWEEK. In addition, every 14 days the volumes in storage groups 1 and 2 are dumped to dump class TWOWEEK so that for those generations, two copies of the volumes in storage groups 1 and 2 are made. However, only one VTOC copy is made for each source volume in a generation even though two copies of the volume are made. Table 13 on page 161 shows the copies in the various generations for one dump cycle.

In this example, the generation-numbering scheme is not the one used by DFSMShsm but is a simplified scheme for instructional purposes.

Table 13. Generations of Example Dumps

Day	Generation	ONEWEEK	TWOWEEK	Discard VTOC Copy
7	1	X		None
14	2	X	X	None
21	3	X		1
28	4	X	X	None

Now, assuming that you have specified that you want two copies of the VTOC saved for each of dump classes ONEWEEK and TWOWEEK, complete the following actions:

1. When generation 1 is made, save one VTOC copy for dump class ONEWEEK.
2. When generation 2 is made, save one VTOC copy for both dump classes ONEWEEK and TWOWEEK.
3. When generation 3 is made, save one VTOC copy for dump class ONEWEEK and discard the VTOC copy from generation 1 because it exceeds the two VTOCs specified for dump class ONEWEEK.
4. When generation 4 is made, make one VTOC copy for dump classes ONEWEEK, TWOWEEK, and FOURWEEK; however, the VTOC copy from generation 2 cannot be deleted even though it exceeds the two VTOCs specified for dump class ONEWEEK, because it is still needed to satisfy the two VTOCs specified for dump class TWOWEEK. At this time, you have VTOC copies from generations 2, 3, and 4. The copies for generations 3 and 4 satisfy the two generations for dump class ONEWEEK. The copies for generations 2 and 4 satisfy the two generations for dump class TWOWEEK.

The VTOCCOPIES subparameter of the DUMPCLASS parameter controls the number of VTOC copies kept for the dump class. The dump VTOC copy data sets are written to ML1 volumes. The space required for these data sets varies with the size of the original VTOC, but it is less than that of the original VTOC because only part of each data set VTOC entry is written in the dump VTOC copy.

When no VTOC copy exists for a particular set of dump copies, DFSMShsm can no longer use its automatic search capability to restore data sets from that dump (although it can restore entire volumes). Lack of a VTOC copy also prevents DFSMShsm from providing listings of the data sets contained in that dump. *Authorized users* can restore data sets from a specified dump volume even though the VTOC copy does not exist.

In the example system, save two VTOC copies for dump classes ONEWEEK and TWOWEEK and no VTOC copy for dump class FOURWEEK.The DUMPCLASS parameters for the DEFINE commands are completed as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
DATASETRESTORE FREQUENCY(7) NORESET -
RETENTIONPERIOD(21) UNIT(ACL) -
VTOCCOPIES(2))
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14) -
DISPOSITION('SEND OFFSITE BIWEEKLY') -
NORESET RETENTIONPERIOD(42) -
UNIT(ACL) VTOCCOPIES(2))
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28) -
DISPOSITION('SEND OFFSITE MONTHLY') -
NORESET RETENTIONPERIOD(84) -
UNIT(ACL) VTOCCOPIES(0))
```

Defining Dump Volumes to DFSMShsm

If you are using SELECTVOLUME(SPECIFIC) and TAPEDELETION(HSMTAPE), you will want to identify your dump volumes to DFSMShsm. Use the ADDVOL command to identify the volume as a dump volume and to specify the kind of unit for the volume and the dump class to which the volume belongs. A sample ADDVOL command for adding a dump volume might be:

```
ADDVOL DMP001 UNIT(3490) DUMP(DUMPCLASS(ONEWEEK))
```

Figure 25 lists the complete set of DFSMShsm commands that the example system uses to control automatic dump processing for SMS-managed storage.

```
/* DEFINING WHEN TO START AUTOMATIC DUMP */
  SETSYS AUTODUMPSTART(0300 0400 0500)
  /* SPECIFYING THE DFDSS BUFFERING TECHNIQUE */
  SETSYS DUMPIO(3,1)

  /* SPECIFYING THE MAXIMUM NUMBER OF DUMP TASKS */
  SETSYS MAXDUMPTASKS(3)

  /* DEFINING THE DUMP CYCLE */
  DEFINE DUMPCYCLE(NNNNNNYNNNNNNYNNNNNNYNNNNNNY -
    CYCLESTARTDATE(95/01/06))

  /* DEFINING THE DUMP CLASSES */
  DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
    DATASETRESTORE FREQUENCY(7) NORESET -
    RETENTIONPERIOD(21) UNIT(ACL) -
    VTOCCOPIES(2))

  DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
    NODATASETRESTORE DISPOSITION('SEND OFFSITE BIWEEKLY') -
    FREQUENCY(14) NORESET RETENTIONPERIOD(42) -
    UNIT(ACL) VTOCCOPIES(2))

  DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
    NODATASETRESTORE DAY(28) -
    DISPOSITION('SEND OFFSITE MONTHLY') NORESET -
    RETENTIONPERIOD(84) UNIT(ACL) -
    VTOCCOPIES(0))

  /* SPECIFYING CONCURRENT COPY */
  SETSYS VOLUMEDUMP(CC)
```

Figure 25. DFSMShsm Automatic Dump Commands for SMS-Managed Storage

Backup Tasks

The following backup tasks are described in this section:

- Specifying which storage group has automatic backup performed
- Specifying how to manage groups of data sets
- Specifying whether to use concurrent copy
- Specifying frequency of backup
- Guaranteeing frequency of backup
- Specifying the maximum number of backup versions to keep
- Specifying how long to keep backup versions
- Specifying who can back up data sets by command
- Specifying whether to perform automatic backup
- Specifying the start time for automatic backup
- Specifying whether to perform volume backups and the kind of device to use
- Specifying the first qualifier for names of backup versions
- Specifying the maximum number of volume backup tasks in each DFSMShsm host
- Specifying the data set backup by command environment
- Specifying whether to back up RACF discrete data set profiles
- Specifying whether DFSMShsm performs spill processing
- Specifying whether to back up only changed data sets
- Specifying how DFSMShsm should respond to a data set in use
- Specifying the days on which automatic backup occurs
- Defining backup volumes to DFSMShsm
- Specifying duplexing for backup tapes

There is also a discussion of the example system management classes in "Example Management Classes" on page 169.

Specifying Which Storage Group Has Automatic Backup Performed

The AUTO BACKUP attribute of the storage group specifies whether DFSMShsm performs automatic incremental backup on the volumes assigned to the storage group. For the example system, the AUTO BACKUP attribute for all storage groups is AUTO BACKUP = Y.

As with the space management example, backup for any of the storage groups is not limited to a single host. However, should you want to limit the processing to a single host, you can use the BACKUP SYSTEM/SYSPLEX NAME attribute in the same way as the MIGRATE SYSTEM/SYSPLEX NAME attribute was used. A sysplex name can also be used as the BACKUP SYSTEM/SYSPLEX NAME.

Figure 26 on page 164 shows the backup attributes added to the storage groups for the example system.

```
  NAME ===> SG1
  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG2
  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG3
  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG4
  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME===>
```

Figure 26. Storage Group Attributes for Backup

Specifying How to Manage Groups of Data Sets

During backup processing, DFSMShsm processes individual SMS-managed data sets to comply with the attributes specified in the management classes to which the data sets are associated. Management class backup attributes let you specify:

- Whether to use concurrent copy
- How often the data sets can be backed up
- How to guarantee frequency of backup
- How many backup versions (maximum) to keep
- How long to keep backup versions
- Who can back up data sets by command
- Whether to perform automatic backup

Related reading: For additional information about management classes and their function in SMS, see the *z/OS DFSMSdfp Storage Administration Reference*.

Specifying Whether to Use Concurrent Copy

Concurrent copy is a system-managed backup process. DFSMS/MVS provides a BACKUP COPY TECHNIQUE attribute for the SMS management class that informs DFSMShsm whether or not to use the concurrent copy facility when backing up data sets during automatic, volume, or command data set backup processing. The BACKUP COPY TECHNIQUE attribute has three parameters:

- CONCURRENT REQUIRED
- CONCURRENT PREFERRED
- STANDARD

These parameters function as follows:

- CONCURRENT REQUIRED (R) — specifies that the concurrent copy facility is required. If a concurrent copy session cannot be established, the data sets will not be backed up.
- CONCURRENT PREFERRED (P) — specifies that the concurrent copy facility is preferred. If a concurrent copy session can be established, the data sets are backed up using the concurrent copy facility. Otherwise, data sets are backed up by DFSMSdss logical dump without concurrent copy.
- STANDARD (S) — specifies that the concurrent copy facility is not to be used, even though it may be available for the data sets. Normal backup processing is used.

When DFSMShsm is performing either automatic, volume, or command data set backup processing, the management class associated with a data set specifies that a concurrent copy BACKUP COPY TECHNIQUE is to be used, and DFSMSdss is the data mover, then DFSMShsm invokes DFSMSdss using the CONCURRENT parameter.

Notes:

1. BACKUP COPY TECHNIQUE applies only to data sets residing on level 0 volumes. Migrated data sets are backed up using DFSMSdss logic dump.
2. Concurrent copy does not apply to catalogs and partitioned data sets.

Specifying Frequency of Backup

The backup attribute BACKUP FREQUENCY governs how frequently the data sets can be backed up automatically. Backup frequency is a management class attribute. This attribute specifies the number of days that must have elapsed since the last backup before the data set can be backed up again automatically, even though the data set may have changed and the volume it resides on is undergoing backup processing. For example, the data set will not be backed up because the frequency specification has not been met if the following conditions are true:

- The management class has a backup frequency of 2
- The data set was backed up yesterday
- The data set has been changed
- The volume is being processed for backup

However, if the backup frequency is 1 or if the data set had been backed up the day before yesterday, the data set is backed up.

The default for BACKUP FREQUENCY is 1.

Note: Normally, DFSMShsm processes volumes only if they were not already processed within the previous 14 hours. A nightly backup starts over at the beginning. To have DFSMShsm start the backup with volumes that were not backed up in the previous cycle, set the backup to a larger number of hours. For a patch command example, refer to a discussion on the patches to be added to the ARCCMD*xx* parmlib member in the *z/OS DFSMShsm Implementation and Customization Guide*.

Guaranteeing Frequency of Backup

The maximum number of days between backups can be guaranteed by the storage group attribute GUARANTEED BACKUP FREQUENCY. This attribute 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 has elapsed, whether or not the data set has been changed. The backup version is created either during DFSMShsm's automatic backup or command volume backup incremental processing.

Note: There is no interaction between GUARANTEED BACKUP FREQUENCY and command backup of a single data set or of a total volume.

If the data set has not changed and the backup version is created because the GUARANTEED BACKUP FREQUENCY has been met, then that backup version replaces the most recent backup version (replacement will not be performed for (H)BACKDS and BACKVOL TOTAL commands). If the data set has changed, backup versions are kept as described in the "Specifying the Maximum Number of Backup Versions to Keep" on page 166 section that follows.

GUARANTEED BACKUP FREQUENCY (which is a maximum) takes precedence over the management class attribute BACKUP FREQUENCY (which is a minimum), such as in the following: 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 or not the data set has been changed. 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 number of tape mounts needed to recover a volume when a few data sets are not being changed. Volume dumps copy all data, even if nearly all of it is already backed up by incremental backup. Having all the data needed for a recovery on a small number of tapes can reduce the number of tape mounts needed to recover a volume entirely from incremental backups.

If the data set's storage group attribute of AUTO BACKUP is NO, then 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 have not been changed.

Specifying the Maximum Number of Backup Versions to Keep

Each time that DFSMShsm makes a backup of a data set, the copy is known as a version. (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.) The number of backup versions determines how far back you can go to recover a data set. You can specify the number of backup versions to keep by using the two NUMBER OF BACKUP VERSIONS attributes in the management class.

The NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) attribute controls the maximum number of backup versions you accumulate for data sets that are currently on either a level 0 volume or a migration volume. You will most likely want to keep multiple backup versions of these data sets. After you have as many versions of a data set as you specify with this attribute, each new backup version that is made of that data set causes the oldest backup version that exists to be discarded. The default for NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) is 2.

The NUMBER OF BACKUP VERSIONS (DATA SET DELETED) attribute controls the maximum number of backup versions you keep for a data set that has been deleted. DFSMShsm does not automatically delete excess backup versions for deleted data sets at the time the data set is deleted. You must issue a DFSMShsm EXPIREBV command that finds and deletes the excess backup versions of deleted data sets. The default for NUMBER OF BACKUP VERSIONS (DATA SET DELETED) is 1.

Note: When you delete a data set, control of the number of backup versions to be kept changes from the NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) attribute to the NUMBER OF BACKUP VERSIONS (DATA SET DELETED) attribute.

Related reading: For additional information about how record length affects the number of backup versions, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Relationship of Frequency and Maximum Number of Backup Versions

Proper use of the FREQUENCY and NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) attributes can give added assurance that data sets can be recovered. Assume that you specify a 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.

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 will contain two days' worth of daily changes, and this may not provide enough granularity for your needs.

You can also control the number of days that a user has in which to detect errors by increasing the number of backup versions retained.

Specifying How Long to Keep Backup Versions

You can control how long to keep extra backup versions for data sets and how long to keep the only backup version of data sets that have been deleted. (DFSMShsm does not delete the only backup version of a data set that exists unless you specifically delete the backup version.) The attributes that control the number of days to keep extra backup versions are overridden by the attributes that control the number of versions to keep. That is, if you have more backup versions than are specified by the appropriate NUMBER OF BACKUP VERSIONS attribute, the excess versions are deleted by the EXPIREBV command even though they may not have reached the age specified by the RETAIN DAYS attributes.

The RETAIN DAYS EXTRA BACKUP VERSIONS attribute controls the number of days that you keep backup versions that are older than the most recent one. You can specify values from 1 to 9999 or NOLIMIT for this attribute. If you use the NOLIMIT value, you can delete extra backup versions based on the following criteria:

- The value you assign to the NUMBER OF BACKUP VERSIONS attribute.
- The issuance of a DFSMShsm command to delete specific backup versions.
- The value assigned to the RETAIN DAYS ONLY BACKUP VERSION attribute.

Note: All remaining backup versions are deleted after the retention period specified if you assign a value other than NOLIMIT.

The RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED) attribute controls how long the last backup version is retained after the data set has been deleted. You can specify values from 1 to 9999 or NOLIMIT. If you use the NOLIMIT value, the last remaining backup version can be deleted only by the value of the NUMBER OF BACKUP VERSIONS attribute being zero, or by a DFSMShsm command to delete the specific backup version. The default for RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED) is 60. The number of days is counted from the time that DFSMShsm detects that the data set has been deleted. DFSMShsm detects that the data set has been deleted when the EXPIREBV command is processed.

When RETAIN DAYS ONLY BACKUP VERSION is set to other than NOLIMIT (1 to 9999), and the RETAIN DAYS EXTRA BACKUP VERSION is set to NOLIMIT,

the resulting effect is to keep the number of backup versions specified by the NUMBER OF BACKUP VERSIONS (DATA SET DELETE) attribute for the amount of time specified by the RETAIN DAYS ONLY BACKUP VERSION attribute. After the retention period, all remaining backup versions are deleted. If NOLIMIT is specified for both RETAIN DAYS EXTRA BACKUP VERSION and RETAIN DAYS ONLY BACKUP VERSION, the backup versions are not deleted by EXPIREBV.

Relationship of the NUMBER OF BACKUP VERSIONS and RETAIN DAYS Attributes

Consider a management class that specifies:

```
NUMBER OF BACKUP VERSIONS (DATA SET EXISTS)        = 4
NUMBER OF BACKUP VERSIONS (DATA SET DELETED)       = 2
RETAIN DAYS EXTRA BACKUP VERSIONS                  = 20
RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED) = 60
```

Now consider a single data set associated with this management class:

- As long as no backup version becomes more than 20 days old and the data set exists, no more than four backup versions are retained because of the action of NUMBER OF BACKUP VERSIONS (DATA SET EXISTS). Backup versions in excess of four are deleted even though they may not yet be 20 days old.
- If backup versions are no longer made (for example, because the data set is not being changed), the next processing of the EXPIREBV command deletes any backup version that is more than 20 days old except for the most recent backup version. (Remember, DFSMShsm *always* requires you to take specific action to delete the last backup version of an existing data set.) The deletion of the oldest backup version is controlled by RETAIN DAYS EXTRA BACKUP VERSIONS. EXPIREBV treats the newest EXTRA backup copy as if it had been created on the same day as the newest BACKUP VERSION. Therefore, the newest EXTRA backup copy is not deleted until the number of days specified in RETAIN DAYS EXTRA BACKUP VERSIONS has passed, starting from the creation date of the newest backup copy.

Assume that the data set is deleted while four backup versions exist that are less than 20 days old:

- The next processing of the EXPIREBV command deletes the two oldest backup versions because of the action of NUMBER OF BACKUP VERSIONS (DATA SET DELETED). Deleting the data set caused a change in the controlling attribute.
- Since there are only two backup versions left, the oldest backup version is also the newest EXTRA backup version. EXPIREBV treats the newest EXTRA backup copy as if it had been created on the same day as the newest BACKUP VERSION. Therefore, the newest EXTRA backup copy is not deleted until the number of days specified in RETAIN DAYS EXTRA BACKUP VERSIONS has passed, starting from the creation date of the newest backup copy.
- When 60 days have passed since a DFSMShsm EXPIREBV command has detected that the data set was deleted, the next processing of the EXPIREBV command deletes the last backup version through the action of RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED).

Specifying Who Can Back Up Data Sets by Command

The ADMIN OR USER COMMAND BACKUP attribute controls who can issue commands to back up data sets associated with the management class. You can specify values of BOTH, ADMIN, or NONE. If you specify BOTH, both DFSMShsm-authorized and non-DFSMShsm-authorized users can issue commands

to back up data sets associated with the management class. If you specify ADMIN, only DFSMShsm-authorized users can issue commands to back up data sets associated with the management class. If you specify NONE, data sets associated with that management class cannot be backed up either automatically or by command. The default for the ADMIN OR USER COMMAND BACKUP attribute is BOTH.

Specifying Whether to Perform Automatic Backup

The AUTO BACKUP attribute controls whether data sets that belong to the management class can be backed up by automatic backup processing. The values you can specify are Y (yes) and N (no). The default for AUTO BACKUP is Y. However, if the ADMIN OR USER COMMAND BACKUP attribute has the value of NONE, data sets associated with that management class are not backed up regardless of the value of the AUTO BACKUP attribute.

Example Management Classes

In the example system there are four management classes: standard, GDS, large, and no migration.

Standard Management Class

For data sets using the standard management class (STANDMC), the following actions occur:

- The data sets are automatically backed up no more frequently than every three days.
- Three versions of the data sets are retained while the data set exists.
- One version of the data set is retained for 60 days after the data set is deleted.
- Extra versions are retained for 20 days after they are created.
- Any users can issue commands to back up data sets.
- Automatic backup processing backs up the data sets.
- Concurrent copy is not used.

Figure 27 shows the management class attributes for STANDMC.

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

SCDS NAME:              SYS1.SMS.SCDSA
MANAGEMENT CLASS NAME:  STANDMC

TO DEFINE MANAGEMENT CLASS, SPECIFY:

  BACKUP ATTRIBUTES
    BACKUP FREQUENCY              ===> 3        (0 to 9999 or blank)
    NUMBER OF BACKUP VERS         ===> 3        (1 to 13 or blank)
      (DATA SET EXISTS)
    NUMBER OF BACKUP VERS         ===> 1        (0 to 13 or blank)
      (DATA SET DELETED)
    RETAIN DAYS ONLY BACKUP VER   ===> 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=Conc Preferred,
                                                 R=Conc Required or S=Standard)
```

Figure 27. Backup Attributes for STANDMC

GDS Management Class

Data sets in generation data groups use the GDS management class (GDSMC) for the following results:

- Changed data sets are automatically backed up every day.
- One backup version is retained while the data set exists.
- One backup version is retained for 14 days after the data set is deleted.
- No extra backup versions ever exist.
- Only DFSMShsm-authorized users can issue commands to back up data sets.
- Automatic backup processing backs up data sets.
- Concurrent copy is not used.

Figure 28 shows the management class attributes for GDSMC.

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

SCDS NAME:              SYS1.SMS.SCDSA
MANAGEMENT CLASS NAME:  GDSMC

TO DEFINE MANAGEMENT CLASS, SPECIFY:

  BACKUP ATTRIBUTES
    BACKUP FREQUENCY              ===> 1        (0 to 9999 or blank)
    NUMBER OF BACKUP VERS         ===> 1        (1 to 13 or blank)
      (DATA SET EXISTS)
    NUMBER OF BACKUP VERS         ===> 1        (0 to 13 or blank)
      (DATA SET DELETED)
    RETAIN DAYS ONLY BACKUP VER   ===> 14       (1 to 9999, NOLIMIT or blank
      (DATA SET DELETED)
    RETAIN DAYS EXTRA BACKUP VERS ===>          (1 to 9999, NOLIMIT or blank
    ADMIN OR USER COMMAND BACKUP  ===> ADMIN    (BOTH, ADMIN, or NONE)
    AUTO BACKUP                   ===> Y        (Y or N)
    BACKUP COPY TECHNIQUE         ===> S        (P=Conc Preferred,
                                                 R=Conc Required or S=Standard)
```

Figure 28. Backup Attributes for GDSMC

Large Management Class

Data sets that use this management class (LARGEMC) receive the following processing:

- The data sets are automatically backed up every other day (except Sunday).
- Four backup versions are retained while the data set exists.
- One version is retained for 60 days after the data set is deleted.
- Backup versions other than the most current are retained for 20 days after they are created.
- Only DFSMShsm-authorized users can issue commands to back up the data sets.
- Automatic backup processing backs up the data sets.
- Concurrent copy is not used.

Figure 29 shows the management class attributes for LARGEMC.

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

SCDS NAME:              SYS1.SMS.SCDSA
MANAGEMENT CLASS NAME:  LARGEMC

TO DEFINE MANAGEMENT CLASS, SPECIFY:

  BACKUP ATTRIBUTES
    BACKUP FREQUENCY              ===> 2        (0 to 9999 or blank)
    NUMBER OF BACKUP VERS         ===> 4        (1 to 13 or blank)
      (DATA SET EXISTS)
    NUMBER OF BACKUP VERS         ===> 1        (0 to 13 or blank)
      (DATA SET DELETED)
    RETAIN DAYS ONLY BACKUP VER   ===> 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  ===> ADMIN    (BOTH, ADMIN, or NONE)
    AUTO BACKUP                   ===> Y        (Y or N)
    BACKUP COPY TECHNIQUE         ===> S        (P=Conc Preferred,
                                                 R=Conc Required or S=Standard)
```

Figure 29. Backup Attributes for LARGEMC

No Migration Management Class

For data sets that have value as long as they are used but lose value quickly when they are not used, the no migration (NOMIGMC) management class has been defined. Data sets that use this management class receive the following processing:

- The data sets are not backed up automatically.
- Both DFSMShsm-authorized and non-DFSMShsm-authorized users can issue commands to back up data sets.
- If a data set is backed up by command, one backup version is retained for 30 days after the data set is deleted.
- Concurrent copy is not used.

Figure 30 shows the attributes for NOMIGMC.

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

SCDS NAME:              SYS1.SMS.SCDSA
MANAGEMENT CLASS NAME:  NOMIGMC

TO DEFINE MANAGEMENT CLASS, SPECIFY:

  BACKUP ATTRIBUTES
    BACKUP FREQUENCY              ===> 1        (0 to 9999 or blank)
    NUMBER OF BACKUP VERS         ===> 1        (1 to 13 or blank)
      (DATA SET EXISTS)
    NUMBER OF BACKUP VERS         ===> 1        (0 to 13 or blank)
      (DATA SET DELETED)
    RETAIN DAYS ONLY BACKUP VER   ===> 30       (1 to 9999, NOLIMIT or blank
      (DATA SET DELETED)
    RETAIN DAYS EXTRA BACKUP VERS ===>          (1 to 9999, NOLIMIT or blank
    ADMIN OR USER COMMAND BACKUP  ===> BOTH     (BOTH, ADMIN, or NONE)
    AUTO BACKUP                   ===> N        (Y or N)
    BACKUP COPY TECHNIQUE         ===> S        (P=Conc Preferred,
                                                 R=Conc Required or S=Standard)
```

Figure 30. Backup Attributes for NOMIGMC

Specifying the Start Time for Automatic Backup

In the example system automatic backup processing starts at 3 a.m., and no later than 4 a.m.. New volumes are not started after 5 a.m. However, at the beginning of automatic backup, the primary host backs up the control data sets.The enqueue on the control data sets and reserve of volumes prevents automatic space management, automatic backup, and automatic dump from running simultaneously while the control data sets are being backed up. Whichever process starts first (space management, dump, or backup) delays the others from starting.

The AUTOBACKUPSTART parameter of the SETSYS command controls the time of day to start automatic backup processing. The SETSYS command that is added to the ARCCMDxx member in hosts 1, 2, and 3 for the example system is:

```
SETSYS  AUTOBACKUPSTART(0300  0400  0500)
```

When DFSMShsm backs up the control data sets, it issues an exclusive enqueue on the control data sets and, in a multiple DFSMS-host environment, reserves the volumes that contain the journal and the control data sets. The enqueue on data sets and reserve of volumes prevents automatic space management, automatic backup, and automatic dump from running simultaneously while the control data sets are being backed up. Whichever process starts first (space management, dump,

or backup) delays the other from starting. You should consider this delay when you plan the start of automatic space management, the start of automatic backup, and the start of automatic dump.

The use of the concurrent copy function with DFSMSdss will significantly reduce the duration of complex-wide serialization of the DFSMShsm CDS resources when performing CDS version backup. This reduction increases the availability of DFSMShsm functions. To achieve maximum DFSMShsm availability, you should place the CDS data sets on SMS-managed devices that support the concurrent copy function, and you should use DFSMSdss to back up the CDSs. If DFSMSdss backs up to tape, each CDS resource data set is copied to an individual tape drive (in parallel).

Specifying Whether to Perform Volume Backup and the Kind of Device to Use

DFSMShsm can use either tape or DASD as target volumes for backup. In the example system, all backup is to tape outside of a system-managed tape library. The BACKUP parameter of the SETSYS command controls whether a host is allowed to do backup and dump processing and limits the kind of device to use as target volumes. In the example system, the following SETSYS command is added to the ARCCMD*xx* member in each host:

```
SETSYS BACKUP(TAPE(ACL))
```

The BACKUP parameter specifies to DFSMShsm that the host is allowed to perform backup and dump processing. The TAPE(ACL) subparameter specifies that backup versions are to be written to tapes mounted on the devices associated with the ACL esoteric name.

The SETSYS command has a DENSITY parameter that is related to mounting a scratch backup tape for the backup unit. However, when the tape unit specified is a cartridge-type device, DFSMShsm ignores the DENSITY parameter.

To prevent the host from performing backup, you would use the NOBACKUP parameter, which prevents both automatic and command backup processing from occurring. The NOBACKUP parameter also prevents automatic and command dump from occurring.

Related reading: For information about how to direct volume backup to a tape library, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Specifying the First Qualifier for Names of Backup Versions

The backup prefix is used in the ACS filters that prevent DFSMShsm-owned data sets from being allocated on SMS-managed volumes. In addition, the use of the prefix allows you to use RACF generic profile protection for all your backup versions.

When DFSMShsm backs up data sets, it leaves the original data set name in the catalog. For use in its own records of the data sets, DFSMShsm generates its own names for the backup versions. The data set name that DFSMShsm generates is of the form:

bprefix*.BACK.T*ssmmhh.user1.user2.Xyddd

The BACKUPPREFIX parameter of the SETSYS command specifies the *bprefix* (high-level qualifier) of the generated name. If you do not specify a backup prefix, DFSMShsm uses the UID you specified in the startup procedure. In the example system the following command is added to the ARCCMD*xx* member in each host:

```
SETSYS BACKUPPREFIX(HSM)
```

Related reading:

- For additional information about the ACS filters, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the data set name, see "Generating Names of Backup Versions of Data Sets" on page 189.

Specifying the Maximum Number of Volume Backup Tasks in Each Host

DFSMShsm can have up to 15 volume backup tasks running concurrently in each host. The MAXBACKUPTASKS parameter of the SETSYS command controls the number of concurrent volume backup tasks that will run. To decide how many concurrent volume backup tasks DFSMShsm should run, consider how many tape units are available if you use tape backup volumes or how many backup DASD units are available. If you are using the duplex tape option for backups, you will need two tape units for every backup task. Also, consider the system work load when volume backup is running. The default for this parameter is 2.

Note: In a multiple DFSMShsm-host environment with concurrent multitasking, it is beneficial to run all tasks on a single host. For example, having eight tasks running automatic backup from one host gives you better performance than splitting the tasks and running four on one host and four on another host.

In the example system, the maximum backup tasks are specified as:

```
SETSYS MAXBACKUPTASKS(3)
```

Specifying the Data Set Backup by Command Environment

The data set backup by command function has a number of options that can control your data set backup operation. These options include the number of tasks, the output device type, and tape management.

Related reading: For additional information about how to tailor the command data set backup environment, see "Using the Data Set Backup by Command Capabilities" on page 209.

Specifying Whether to Back Up RACF Discrete Data Set Profiles

If you are using RACF discrete profiles rather than generic profiles, you can have DFSMShsm back up the profile when it backs up the data set. The PROFILEBACKUP parameter of the SETSYS command controls whether discrete profiles are backed up. Because the example system is in a RACF always-call environment and uses generic profiles, choose the NOPROFILEBACKUP option of this parameter. In the example system, the following command is added to the ARCCMD*xx* member in each host:

```
SETSYS NOPROFILEBACKUP
```

Specifying Whether DFSMShsm Performs Spill Processing

When DFSMShsm performs backup to DASD volumes, the DASD volumes can become full. If this occurs, DFSMShsm can perform the spill process, which moves older backup versions of data sets to other volumes, known as *spill volumes*. Spill volumes are usually tape volumes.

Because the example system uses tape for backup volumes, no spill processing is possible. Therefore, the following command is added to the ARCCMD*xx* member in each host:

```
SETSYS NOSPILL
```

Specifying Whether to Back Up Only Changed Data Sets

When DFSMShsm is first installed, you might have many data sets on your volumes with the data-set-changed indicator in the data set VTOC entry off (such as data sets for which the bit has been turned off by some other program product). Because DFSMShsm uses the data-set-changed indicator to decide if a data set needs to be backed up for incremental backup, these data sets would not be backed up. DFSMShsm provides an INCREMENTALBACKUP parameter for the SETSYS command that allows you to specify whether to back up such data sets. If you specify the ORIGINAL subparameter of the INCREMENTALBACKUP parameter, DFSMShsm searches its records for each data set that does not have the data-set-changed indicator turned on. If DFSMShsm has a backup copy of the data set, the data set is not backed up again. If DFSMShsm does not have a backup copy of the data set, a backup version is made.

Of course, searching the records each time a data set is found with the data-set-changed indicator turned off takes a lot of time. You may want to use the ORIGINAL subparameter for the first few backup cycles, then change to the CHANGEDONLY subparameter. When you use the CHANGEDONLY subparameter, DFSMShsm backs up only those data sets that have the data-set-changed indicator turned on.

In the example system, it is presumed that all old data sets have been backed up or have their data-set-changed indicators set to on, so the following command is added to the ARCCMD*xx* member in each host:

```
SETSYS INCREMENTALBACKUP(CHANGEDONLY)
```

Specifying How DFSMShsm Should Respond to a Data Set in Use

To ensure integrity and consistency of the data in a backup version, DFSMShsm tries to serialize on system resources to indicate whether a data set is in use. This prevents the data set from changing while DFSMShsm is backing it up. However, accepting the philosophy that something is better than nothing, you may still want backup copies of data sets that are in use at the time of backup, so long as you can identify (if you should need to recover from such a backup) that a given backup was created while the data set was possibly being changed.

If DFSMShsm makes an unserialized backup, after first trying and failing to serialize on system resources, the data-set-changed indicator remains on; the data set is still eligible to be backed up. If DFSMShsm is instructed *not* to serialize on system resources on the first attempt to back up a data set, the data-set-changed indicator is turned off. The instruction for no serialization is an option of the ARCHBACK user macro and the ARCBDEXT installation exit. However, the ARCINBAK inline backup program always specifies this option, because it expects the JCL to perform its own serialization.

DFSMShsm recognizes two situations involving backup and a data set being in use. First, there are data sets that you do not expect to be open for update, but that may 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 such a retry is required or merely preferable. The backups that the INUSE subparameter allows may be placed temporarily on ML1 DASD.

In the example system it is presumed that you want DFSMShsm to attempt such 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 is added to the example system:

```
SETSYS BACKUP(INUSE(RETRY(Y) DELAY(10) +
                    SERIALIZATION(PREFERRED)))
```

For the second situation, there are data sets (typically a database) that you expect to be open for update most of the time.You can use the data set backup installation exit ARCBDEXT 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 given data set.

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

Specifying the Days on Which Automatic Backup Occurs

The BACKUP parameter of the DEFINE command specifies a backup cycle that specifies the days on which the host does automatic backup processing. You can also specify the day to start the cycle so that the cycle does not change with each re-initialization of DFSMShsm. In the example system, the following command is added to the ARCCMD*xx* member in each host:

```
DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(1995/01/06))
```

The command establishes a 7-day cycle that begins on a Monday. Backup processing is done on every day except Sunday.

DFSMShsm stores the date you specify with CYCLESTARTDATE as the date the backup cycle began. If you do not specify a CYCLESTARTDATE parameter with the DEFINE command but a cycle start date is already stored and the backup cycle length has not changed, DFSMShsm uses the cycle start date that is already stored. If no cycle start date is stored and you do not specify one, DFSMShsm uses the day that the DEFINE BACKUP command is issued as the cycle start date.

Rule: The date 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.

If you are backing up only to DASD, you should also specify how many DASD backup volumes you want DFSMShsm to use for each day in the backup cycle.

Related reading: For more information about defining the desired number of DASD backup volumes, see *z/OS DFSMShsm Storage Administration Reference*.

Defining Backup Volumes to DFSMShsm

If you are using SELECTVOLUME(SPECIFIC) and TAPEDELETION(HSMTAPE), you may want to identify your backup volumes to DFSMShsm. Use the ADDVOL command to identify the volume as a backup volume and to specify the kind of unit for the volume and whether the volume is a daily backup volume or a spill volume. The following provides a sample ADDVOL command for adding a DASD backup volume:

```
ADDVOL BAK001 UNIT(3390) BACKUP(DAILY(5))
```

This command makes BAK001 a daily backup volume that is used on day five of the backup cycle.

Figure 31 shows the complete set of DFSMShsm commands that the example system uses to control automatic backup processing for SMS-managed storage.

```
/* SPECIFYING THE TIME AUTOMATIC BACKUP IS TO START */
     /* (PROCESSING UNITS 1, 2, AND 3)  */
     SETSYS AUTOBACKUPSTART(0300 0400 0500)
     /* SPECIFYING THE TIME AUTOMATIC BACKUP IS TO START */


     /* SPECIFYING TO DIRECT BACKUP TO TAPE, AND TO RETRY */
     /* BACKUP WHEN A DATA SET IS IN USE.  */
     SETSYS BACKUP(TAPE(ACL) INUSE(RETRY(Y) DELAY(10) +
                             SERIALIZATION(PREFERRED)))

     /* SPECIFYING FIRST QUALIFIER FOR NAMES OF BACKED UP */
     /* DATA SETS   */
     SETSYS BACKUPPREFIX(HSM)

     /* SPECIFYING MAXIMUM NUMBER OF BACKUP TASKS */
     SETSYS MAXBACKUPTASKS(3)

     /* SPECIFYING BACKUP OF RACF DISCRETE DATA SET */
     /*PROFILES   */
     SETSYS NOPROFILEBACKUP

     /* SPECIFYING SPILL PROCESSING */
     SETSYS NOSPILL

     /* SPECIFYING BACKUP OF ONLY CHANGED DATA SETS */
     SETSYS INCREMENTALBACKUP(CHANGEDONLY)

     /* SPECIFYING DAYS FOR AUTOMATIC BACKUP TO OCCUR */
     DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(1995/01/06))
```

Figure 31. DFSMShsm Automatic Backup Commands for SMS-Managed Storage

Specifying Duplexing for Backup Tapes

The duplex tape option creates two identical tapes in parallel, with one tape designated as the original and one the alternate. To create duplex backup tapes, specify:

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

If duplexing is not desired for your backup tapes, specify:

```
SETSYS DUPLEX(BACKUP(N))
```

The DFSMShsm default is no duplexing.

Automatic Availability Management Operation

Previously, it has been pointed out how the commands to control DFSMShsm are structured for the example system. In the text that follows, the example system is used to describe what DFSMShsm does in the process of automatic availability management. Dump and backup are shown as two separate processes that contribute to total availability management.

The example system uses cartridge-type tape units in single-file mode for backup volumes. Also, the example system always selects scratch volumes because there are no volumes identified to DFSMShsm as dump or backup tape volumes and all partial tapes are marked full when they are demounted. The example system defers requests for tape mounts until OPEN processing for the data set occurs. These parameter selections minimize interference with other DFSMShsm functions when a tape mount is requested.

Dump Processing

Figure 2 on page 13 shows the example system for dump processing. For this process, the dump volumes are mounted on the cartridge-type devices designated as 3490 with the automatic cartridge loader feature. Because dump processing moves all of the data sets on the volumes that are processed, any data set names on the volumes in each group are not shown.

Assume that it is 3 a.m. on 2003/02/02. This is day 28 of the dump cycle that was established with the DEFINE command. Unless a specific action is taken to prevent it, all hosts that are active begin automatic dump processing for the following reasons:

- BACKUP in the SETSYS command is specified
- AUTODUMPSTART(0300 0400 0500) in the SETSYS command is specified
- It is a Y day in the dump cycle
- NOREQUEST in the SETSYS command is specified

At the beginning of automatic dump in each of the hosts when SMS is active, DFSMShsm obtains a list of the volumes associated with storage groups that the host is allowed to process and that have the AUTO DUMP ===>Y attribute, are type POOL, and have a status of ENABLED, QUIESCED NEW, or DISABLED NEW. Volumes in storage groups that have a host specified by the DUMP SYSTEM/SYSPLEX NAME have a host affinity. If SMS is not installed and active, DFSMShsm processes only non-SMS-managed DASD volumes.

These DFSMShsm managed volumes are only considered candidates if:

- The volume has not been processed in any host by automatic dump in the last 14 hours
- The frequency requirement specified for the dump class to which the volume belongs has been met

DFSMShsm first determines whether there are volumes with dumps that can be stacked. Those volumes associated with the same set of eligible dump classes, of which at least one has a STACK value of more than one, are considered "stackable" together. Within those volumes, the order is:

1. Affinity SMS volumes
2. Non-SMS and non-affinity SMS volumes

A given dump tape contains dump copies from only one of the two categories. In the second category, dumps of SMS-managed and non-SMS-managed volumes can be contained on the same dump tape.

After all volumes eligible for stacking have been dumped, the remaining volumes are dumped, with DFSMShsm selecting the order of processing.

After DFSMShsm has attempted to process all the candidate volumes eligible for automatic dump, it retries the selection of any of these volumes found in use by another DFSMShsm function.

Dump processing is performed in three phases in the primary host and in one phase in the nonprimary hosts. These phases are:

- Delete expired dump copies (primary host only)
- Dump DFSMShsm-managed volumes for which automatic dump is requested
- Delete and uncatalog excess dump VTOC copy data sets (primary host only)

If a secondary host is made eligible to take over for a primary host that has failed, it can perform the expiration of dump copies, and deletion of excess dump VTOC copy data sets.

Deleting Expired Dump Copies

In the first phase of automatic dump on the primary host, the expired dump copies are deleted. Dump copies become eligible to be automatically deleted when they expire or when the maximum number of dump generations is reached and a new generation of dump copies is to be made. Dump copies are automatically deleted after the number of days specified in the RETENTIONPERIOD subparameter of the DUMPCLASS parameter of the DEFINE command has elapsed. This phase of auto dump causes the status of the dump tapes containing the deleted copies to be changed from expired to available if they remain under DFSMShsm control.

DFSMShsm does not automatically delete the last and only remaining copy of a source volume. The dump volumes that make up this dump copy remain in EXPIRED status. To determine which dump tapes are currently in EXPIRED status, issue the LIST DUMPVOLUME command with the SELECT(EXPIRED) option. To delete the dump volumes in this copy, issue the DELVOL command with the LASTCOPY option for one of the dump volumes. All volumes in the copy are then deleted.

Defining What to Do with Tapes Marked Unavailable during DUMP Processing

When dump processing encounters an I/O error on the output tape, DFSMShsm marks the tape unavailable. The tape is left under DFSMShsm's control to prevent re-release of the unavailable tape. This prevents DFSMShsm from selecting the tape for scratch, and prevents other system functions from encountering the unavailable tape before reinitialization.

Use the following procedure to remove the tape from DFSMShsm's control:

- Issue the LIST DUMPVOLUME SELECT(UNAVAILABLE) command to get a list of the tapes marked unavailable.
- Run the relabel utility that is provided by your tape management system, such as EDGINERS from DFSMSrmm™. If no tape management system is available, use the IEHINITT utility.
- Issue DELVOL *volser* DUMP(PURGE) for each volume.

Dumping DFSMShsm-Managed Volumes

Before beginning this phase, each host obtains from SMS a list of the volumes in storage groups that have the AUTODUMP attribute = Y. It also obtains a list of the dump classes to which the volumes in each storage group are to be dumped. DFSMShsm then starts automatic dump tasks in each host, until each host has as many dump tasks as are specified by the MAXDUMPTASKS parameter of the SETSYS command.

The following process is followed first for volumes with dumps that can be stacked, then for the remaining volumes: If the maximum number of dump tasks is reached and there are still more volumes to process, DFSMShsm waits for any already-active volume dump task to complete dumping its current volume. When a volume dump task completes dumping its current volume, it begins dumping the next DFSMShsm-managed volume with the automatic dump attribute. This process continues until all DFSMShsm-managed volumes have been dumped, or the quiesce time is passed.

When processing each volume, DFSMShsm checks the FREQUENCY and DAY subparameters specified in the dump classes to which the volume is assigned. If the DAY corresponds with the day of the dump cycle, and the time since the volume was last dumped is not less than the FREQUENCY specified, the volume is dumped to that dump class. The determination of frequency of dumps includes dumps made by command as well as automatic dumps.

As an example of the effect of the DAY and FREQUENCY subparameters, it has been stated that you are beginning this description of processing for day 28 in a specific dump cycle. Consider volume SG2002 in Figure 2 on page 13. Storage Group 2 specifies that the volumes be dumped to dump classes ONEWEEK and TWOWEEK. Now assume that someone had issued a command to dump SG2002 to dump class ONEWEEK on day 16 of the dump cycle. SG2002 would not have been dumped on day 21, as was specified by ONEWEEK for that day, because less than seven days had elapsed since it was last dumped. It is now day 28. Dump class ONEWEEK specifies that you can dump the volume to it every seven days, so that SG2002 is dumped to dump class ONEWEEK.

Now consider volume SG4001. Storage Group 4 is dumped to dump classes ONEWEEK and FOURWEEK. Dump class FOURWEEK specifies that volumes can

be dumped to it only on day 28 in the dump cycle. Therefore, SG4001 is dumped to dump class FOURWEEK because it is day 28 in the dump cycle.

If the maximum number of dump generations has been reached and another dump is to be performed, the oldest generation is deleted. If the RETENTIONPERIOD has the attribute of NOLIMIT, a dump copy is automatically deleted only when it is a member of a dump generation that exceeds the maximum of 100.

Volume Serialization

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 being dumped should be kept at an absolute minimum during the full-volume dump process, or errors will occur.

Related reading: For additional information about DFSMSdss full-volume dump, see the *z/OS DFSMSdss Storage Administration Guide* and *z/OS DFSMSdss Storage Administration Reference*.

Names of Dump Copy Data Sets

When DFSMShsm initially allocates a dump volume for output during a volume dump operation, it generates a unique data set name and supplies this name on the dynamic allocation request. (If dump copies are being stacked on a dump volume, each dump copy is in its own file and has its own unique data set name. For each dump copy, DFSMShsm stores the file sequence number and beginning file block ID in the BCDS records for use during restore.) Because DFSMSdss writes a full-volume dump copy as one data set, this data set name is the same for all tape volumes that are a part of the same dump copy. This name is supplied by DFSMShsm on the dynamic allocation of a dump copy during a restore operation.

The format of the dump copy data set name is:

```
bprefix.DMP.dclass.Vvolser.Dyyddd.Tssmmhh
```

where:

- *bprefix* is the prefix you specify with the SETSYS BACKUPPREFIX command
- DMP, V, D, and T are constants
- *dclass* is the dump class name
- *volser* is the source volume serial number
- *yy* is the last two digits of the year of the dump date stamp
- *ddd* is the Julian day of the dump date stamp
- *ssmmhh* is the time in seconds, minutes, and hours

Related reading: For additional information about how the DFSMShsm unique data set name is used, see "Determining the Dump Copy Data Set Name" on page 427.

Write Dump VTOC Copy Data Set

If you request it, DFSMShsm makes a dump VTOC copy data set for each dump generation that is made for a volume. When DFSMShsm makes a dump VTOC copy data set, it also catalogs the data set. Only a single dump VTOC copy is made for each generation, no matter how many copies are in the generation.

The dump VTOC copy data sets are physical sequential data sets written on ML1 volumes. They contain an entry for each data set VTOC entry that is in the source volume's VTOC when the volume is dumped. These data sets are created just before DFSMShsm invokes DFSMSdss. VTOC copy data sets are not made when DFSMShsm dumps a DFSMShsm-owned volume.

The data set name of the dump VTOC copy is:

bprefix.DUMPVTOC.T*ssmmhh*.V*volser*.D*yyddd*

where:

- *bprefix* is the prefix you specified with the BACKUPPREFIX parameter of the SETSYS command
- DUMPVTOC, T, V, and D are constants
- *ssmmhh* is the time in seconds, minutes, and hours
- *volser* is the source volume serial number
- *yyddd* is the year and day

With the following sample job, you can selectively copy all dump VTOC copy data sets to tape. These tapes can be taken off site as a group for use during the recovery/restore process.

```
//DFDSSDUMP   JOB 'ACCOUNT #','NAME',MSGLEVEL=(1,1)
//            EXEC PGM=ADRDSSU
//SYSPRINT    DD SYSOUT=A
//TAPE2       DD UNIT=3490,VOL=SER=DPVTOC,LABEL=(,BLP)
//SYSIN       DD *
  DUMP DATASET (INCLUDE(bprefix.DUMPVTOC.**))
       OUTDDNAME (TAPE2)
/*
```

bprefix is the prefix you specified with the BACKUPPREFIX parameter of the SETSYS command.

Selecting Tape Dump Volumes

For information about selecting tape dump volumes, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Deleting and Uncataloging Excess Dump VTOC Copy Data Sets

When a volume dump of a level 0 volume is directed to a dump class that has a VTOCCOPIES value greater than zero, a VTOC copy data set is created. Only one VTOC copy data set is created for each dump generation. If more dump copies of a given volume associated with the same dump class are kept than the specified limit of VTOCCOPIES, the VTOC copies created for the older dumps are excess. If the VTOC copy is in excess for all other dump copies in the same generation, the VTOC copy is deleted and uncataloged during this phase of automatic dump.

Automatic Backup Processing

Assume that it is 3 a.m. on 2003/01/06. Because this is a Monday, it is a Y day in the backup cycle. The combination of the Y day in the backup cycle, the BACKUP parameter of the SETSYS command, and the AUTOBACKUPSTART parameter of the SETSYS command cause automatic backup to start in host 1.

Automatic backup is performed in four phases:
1. Backing up the control data sets
2. Moving backup versions
3. Backing up migrated data sets
4. Backing up DFSMShsm-managed volumes

In multiple DFSMShsm-host configurations, only the primary host performs the first three phases. The first three phases are performed as successive phases of a single task.

If the primary host fails in a multiple DFSMShsm-host environment where the secondary host promotion function is being used, the promoted host can take over the first three phases for the failed primary host.

At the beginning of backup in each of the hosts, DFSMShsm requests a list of all defined management classes. If SMS does not return the list, DFSMShsm considers SMS inactive and issues a message that backup cannot be performed for SMS-managed volumes. If SMS is active, DFSMShsm obtains a list of the volumes associated with storage groups that the host is allowed to process, and that have the following characteristics: the AUTO BACKUP ===> Y attribute, are type POOL, and have a status of ENABLED, QUIESCED NEW, or DISABLED NEW. Volumes in storage groups that have a host specified by the BACKUP SYSTEM/SYSPLEX NAME have a host affinity. If SMS is not installed and active, DFSMShsm processes only non-SMS-managed DASD volumes. For purposes of the illustration, it is assumed that SMS is installed and active.

DFSMShsm selects the following order of processing as a method of controlling input path contention, subject to prioritization for SMS volumes restricted to processing by one host:
1. Affinity SMS volumes
2. Retry of in-use affinity SMS volumes
3. Non-SMS and nonaffinity SMS volumes

After DFSMShsm has attempted to process all the candidate volumes eligible for automatic backup, DFSMShsm retries the selection of any of these volumes found in-use by another DFSMShsm function.

Related reading:

- For additional information about secondary host promotion, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about retrieving a list of management classes and a list exception, see "DFSMShsm Retrieval of Management Classes" on page 433.

Phase 1: Backing Up the Control Data Sets

The first phase of automatic backup is the backing up of the DFSMShsm control data sets (CDSs). This process is called CDS version backup, and it is controlled by the CDSVERSIONBACKUP parameter of the SETSYS command.

Before backing up the control data sets and the journal data set, DFSMShsm issues an exclusive enqueue against the control data sets. In a multiple DFSMShsm-host environment, if the control data sets are *not* being accessed with RLS, DFSMShsm reserves the volumes containing the journal and the control data sets. If the control data sets *are* being accessed using RLS, DFSMShsm does an exclusive enqueue.

The enqueue prevents all data set or volume-type functions from accessing the CDSs while they are being backed up. CDS backup may have to be delayed for other DFSMShsm tasks to release their shared enqueues on this resource before it can start backing up the individual CDSs. Therefore, you should carefully consider the best time to start backup of the CDSs.

The use of the concurrent copy function with DFSMSdss will significantly reduce the duration of complex-wide serialization of the DFSMShsm CDS resources when performing CDS version backup. This reduction increases the availability of DFSMShsm functions. To achieve maximum DFSMShsm availability, you should place the CDS data sets on SMS-managed devices that support the concurrent copy function, and you should use DFSMSdss to back up the CDSs. If DFSMSdss backs up to tape, each CDS data set (each cluster if multi-cluster CDSs are used) is copied to an individual tape drive (in parallel).

If CDS backup is invoked while the CDSs are accessed in RLS mode, DFSMSdss must be the datamover. If the backup is directed to tape, the PARALLEL parameter of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY) must be specified. Automatic backup overrides the settings if they are not correct and issues error message ARC0793I. CDS backup fails.

If a long-running function starts before automatic backup, backup of the CDSs is delayed until the long-running task completes the unit of work, volume, or data set that it enqueues the CDS backup resource on.

Using the CDSVERSIONBACKUP parameter, you can specify how many versions of the CDS backups be kept active. Each time CDS version backup runs, DFSMShsm assigns a new version number to the newly-created backup data sets and uses that version number as the final qualifier of the CDS backup data set names.

When backing up CDSs to tape, DFSMShsm always creates and catalogs the new version and expires the oldest version, so that only the latest versions are available. When backing up the CDSs to DASD, the oldest existing version is reused to record the new backup, and when the backup is complete, that oldest version is renamed and recataloged as the newest version.

DFSMShsm optionally invokes the CDS backup installation exit (ARCCBEXT) that allows you to access or validate the backup versions after DFSMShsm creates them.

When the CDS backup is complete, DFSMShsm releases the volumes containing the journal data set and each control data set, and removes the exclusive enqueue against the control data sets, allowing other DFSMShsm functions to proceed.

Related reading: For additional information about the ARCCBEXT installation exit, see *z/OS DFSMS Installation Exits*.

Phase 2: Moving Backup Versions

Before moving backup versions, DFSMShsm checks to see if any internal TAPECOPY processing is needed for backup volumes, and if so, schedules internal TAPECOPY MWEs. TAPECOPY processing could be needed because of duplexing errors.

Backup versions that are created by data set backup commands may be stored temporarily on ML1 volumes. After DFSMShsm backs up the control data sets, the primary host moves any backup versions created by data set backup commands that are temporarily residing 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. Automatic backup is not scheduled to run if you have specified SETSYS AUTOBACKUPSTART(0) or have specified N for this day in the backup cycle.

Phase 3: Backing Up Migrated Data Sets

A migrated data set might not have a backup version because it has migrated after it was created or changed but before it was backed up. Such a condition could occur if a MIGRATE command were issued for the data set shortly after it was created or changed. The migration could be to either an ML1 volume or a DASD ML2 volume.

A data set needs to be backed up when all of the following 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 DFSMShsm-managed volume 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.

In backing up migrated data sets, DFSMShsm reads the required records in the BCDS. As each record is read, DFSMShsm passes control to the data set backup installation exit (ARCBDEXT) if the installation exit is enabled. If ARCBDEXT does not prevent backup, DFSMShsm checks (1) the BACKUP FREQUENCY attribute from the management class associated with the data set when the data set migrated, and (2) the recorded last backup date. If the number of days since the last backup date equals or exceeds the BACKUP FREQUENCY attribute value, DFSMShsm backs up the data set. If DFSMShsm cannot determine a last backup date and no backup version exists, it backs up the data set. If the number of days since the last backup date is less than the BACKUP FREQUENCY attribute value, DFSMShsm retains the information in the BCDS and does not back up the data set.

Related reading: For additional information about the ARCBDEXT installation exit, see *z/OS DFSMS Installation Exits*.

Phase 4: Backing Up SMS-Managed Volumes from Storage Groups with the Automatic Backup Attribute

Before backup processing begins for SMS-managed volumes, DFSMShsm requests from SMS a list of management classes defined to the configuration in which DFSMShsm is running. 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 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 up to the maximum 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 have been backed up or the quiesce time has passed.

Backup processing examines each data set on the volume individually. Backup processing for each data set:

- Checks for unmatched data sets 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

Processing Unmatched Data Sets

If the catalog entry, the data set VTOC entry, or both, are missing, the data set is an unmatched data set. For an unmatched data set, the space management and backup exit (ARCSAEXT) is invoked if the exit is enabled. If the exit indicates that further eligibility testing be performed for this data set, DFSMShsm issues a message that this data set is unmatched. If the exit indicates that further eligibility testing not be performed, DFSMShsm issues no message. In either case, processing continues on the next data set.

Deleting Temporary Data Sets

If for some reason (such as a system outage) SMS-managed temporary data sets are not deleted at end of job, DFSMShsm attempts to delete them during backup. In a JES3 environment, an SMS-managed temporary data set must be two days old before it can be deleted. These data sets must be as old or older than the integrity age, which is normally zero days, but can be one or two days.

Related reading: For additional information about data set integrity age, see "Integrity Age for Space Management" on page 40.

Create a VTOC Copy Data Set Entry for the Volume

As DFSMShsm examines each data set, it creates a VTOC copy data set entry from the information contained in the data set's VTOC entry. The VTOC copy data set is used in recovering a complete volume. The VTOC copy data set is stored on an ML1 volume.

For SMS-managed volumes, DFSMShsm records all the data sets on the volume except unmatched data sets and temporary data sets.

DFSMShsm keeps the two most recent versions of the VTOC copy data sets for the volumes it backs up. If an excess VTOC copy results when DFSMShsm backs up a volume, it deletes the oldest VTOC copy data set.

The VTOC copy data set is used to guide volume recovery. When a VTOC copy data set is written, DFSMShsm catalogs the data set.

For each volume that is backed up, DFSMShsm generates a unique name for the backup VTOC copy data set.The data set name of the VTOC copy is:

bprefix.VTOC.T*ssmmhh*.V*volser*.D*yyddd*

where:

- *bprefix* is the prefix you specified with the BACKUPPREFIX parameter of the SETSYS command, if entered. Otherwise, *bprefix* is the DFSMShsm default UID.
- VTOC, T, V, and D are constants.
- *ssmmhh* is the time in seconds, minutes, and hours.
- *volser* is the source volume serial number.
- *yyddd* is the year and day.

With the following sample job, you can selectively copy all backup VTOC copy data sets to tape. These tapes can be taken off site as a group for use during the recovery/restore process.

```
//DFDSSDMP   JOB 'ACCOUT #','NAME',MSGLEVEL=(1,1)
//           EXEC PGM=ADRDSSU
//SYSPRINT   DD SYSOUT=A
//TAPE1      DD UNIT=3490,VOL=SER=BUVTOC,LABEL=(,BLP)
//SYSIN      DD *
  DUMP DATASET (INCLUDE(bprefix.VTOC.**)) OUTDDNAME (TAPE1)
/*
```

bprefix is the prefix you specified with the BACKUPPREFIX parameter of the SETSYS command, if entered. Otherwise, *bprefix* is the DFSMShsm default UID.

Backing Up Eligible Data Sets

DFSMShsm loops through the entries for the data sets on the volume and backs up all those data sets that are eligible. An eligible data set is one that:

- Has the management class ADMIN OR USER COMMAND BACKUP attribute value of ADMIN or BOTH
- Has the management class AUTO BACKUP attribute value of Y
- Has eligibility, as determined by the data-set-changed indicator, as follows:
  - If the data-set-changed indicator is off and SETSYS INCREMENTALBACKUP(ORIGINAL) is specified, DFSMShsm backs up the data set if no prior backup version exists.
  - If the data-set-changed indicator is on and DFSMShsm can determine a last backup date, DFSMShsm backs up the data set if the number of days since the last backup date equals or exceeds the value specified with the management class BACKUP FREQUENCY attribute.
  - If the data-set-changed indicator is on but DFSMShsm cannot determine a last backup date (indicating a new data set), DFSMShsm backs up the data set.
  - If the data-set-changed indicator is on and the data set resides on a volume in a storage group that is defined with the GUARANTEED BACKUP FREQUENCY attribute, DFSMShsm backs up the data set if the number of

days since the last backup date equals or exceeds the value that is specified by GUARANTEED BACKUP FREQUENCY days.

If the data set is eligible for backup and the exit is enabled, DFSMShsm passes control to the data set backup exit (ARCBDEXT).

For those data sets that are backed up, DFSMShsm invalidates the oldest excess versions that exceed the limit specified with the NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) management class attribute.

Related reading:

- For additional information about the ARCBDEXT installation exit, see *z/OS DFSMS Installation Exits*.
- For additional information about data mover support for DFSMShsm availability management for backup data sets, see the table called "Availability Management—Data Mover Support" in chapter three of the *z/OS DFSMShsm Implementation and Customization Guide*.

Resetting the Data-Set-Changed Indicator and Last Backup Date

After each data set is backed up, if the backup was serialized, DFSMShsm turns off the data-set-changed indicator in the data set's VTOC entry on the source volume. The data-set-changed indicator is also turned off for data sets that are not serialized, if DFSMShsm has been instructed not to serialize on the first attempt to back up.

Note: With multivolume data sets, the data-set-changed indicator is reset only on the first volume and is left on for all subsequent volumes.

No serialization is an option of the ARCHBACK user macro and the ARCBDEXT installation exit. However, the ARCINBAK inline backup program always specifies this option, because it expects the JCL to perform its own serialization. In addition, DFSMShsm writes the current date in the catalog entry and in the last backup date field in its own MCB record.

Backing up VSAM Data Sets

DFSMShsm backup functions that exist for non-VSAM data sets also exist for VSAM data sets cataloged in the integrated catalog facility catalog. DFSMShsm backs up VSAM data sets on level 0 volumes by using the Access Method Services EXPORT command or DFSMSdss as a data mover.

The backup of a VSAM data set is performed only on a complete VSAM sphere. The base cluster and all alternate index clusters defined on the base cluster are backed up as part of the data set and are not specified separately. Paths are included automatically in the data set backup.

Note: When a VSAM compressed data set is backed up, DFSMShsm suspends any software compaction requested and records in the BCDS that the data set is striped, compressed, or both.

DFSMShsm can back up multiple-volume, SMS-managed, VSAM data sets when it processes the first volume of the base data component. Normally, DFSMShsm prevents update access to any VSAM data set that is currently being backed up. This is done to prevent possible inconsistency of the backup copy.

Tip: You can override this protection for VSAM data sets with the SETSYS INUSE RETRY command or the ARCBDEXT exit.

Some installations, however, would prefer that their jobs not fail when they attempt to open a VSAM data set that is currently being backed up. There are two methods to accomplish this, one which includes the risk of having a corrupted backup copy, and one which excludes that risk:

- One method is to use the DFSMShsm data set backup installation exit, ARCBDEXT. This exit can be coded to change the way DFSMShsm performs serialization on VSAM data sets, allowing batch jobs to open the data sets while the data sets are being backed up. This technique, however, exposes installations to the possibility of creating unserialized backup copies.
- The other method is to add an IEFBR14 step to any batch job that accesses any VSAM data set for updating. The only function of the added step is to access the data set with a DISP of MOD or OLD. This step should be placed immediately prior to the step that accesses the data set for updating. The IEFBR14 step attempts to get an exclusive ENQ on the data set, which causes the job to wait until DFSMShsm has finished processing the data set before the subsequent step accesses the data set. This method may require more time to implement than the ARCBDEXT solution, but it does eliminate the possibility of having corrupted backup copies.

Related reading:

- For additional information about the ARCBDEXT installation exit, see *z/OS DFSMS Installation Exits*.
- For additional information and an example of how to use the ARCBDEXT installation exit in the SYS1.SAMPLIB file, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Generating Names of Backup Versions of Data Sets

For each data set that is backed up, DFSMShsm generates a name for the backup version. The name also applies to the name of the backup version of the integrated catalog facility catalog and to DFSMShsm backup profiles. The name is in the form:

bprefix.BACK.T*ssmmhh.user1.user2.Xyddd*

where:

- *bprefix* is the prefix you specify with the SETSYS BACKUPPREFIX command.
- BACK and T are constants.

  Note: If two data sets with the same first two high-level qualifiers are backed up at the same second on the same day, DFSMShsm assigns a letter other than T to one of these data sets.
- *ssmmhh* is the time in seconds, minutes, and hours.
- *user1* and *user2* are the first two qualifiers of the data set name.
- *Xyddd* is the year and day of the year. You can replace *X* with a letter that represents the decade, as follows:

G = 1970	B = 2020
H = 1980	C = 2030
I = 1990	D = 2040
J = 2000	E = 2050
A = 2010	F = 2060

If, for example, J3032 replaces *Xyddd*, then J3032 represents February 1, 2003.

If you need to access a single-file format tape, you must specify a valid data set name for the single file that comprises up to 255 tape volumes. The backed up data set name for each set of cartridge-type tape volumes in single-file format is:

bprefix.BACKTAPE.DATASET

The tape data set name for each copy of a cartridge-type tape volume created by the TAPECOPY command or the duplex tape option is:

bprefix.COPY.BACKTAPE.DATASET

Only the last 17 characters (.BACKTAPE.DATASET) are recorded in the tape label. To access the single-file format data set, you must supply any valid first qualifier along with the .BACKTAPE.DATASET qualifiers.

Backing Up RACF Discrete Profiles for Cataloged Data Sets

When DFSMShsm backs up a cataloged RACF-indicated data set, it optionally creates a RACF profile, called the backup profile. The original data set's profile is a model for the backup profile, but the volume serial number is MIGRAT, which means the backup profile does not correspond to an existing data set. This eliminates the need for further updates to the backup profile when DFSMShsm moves the backup versions from an ML1 volume or backup volume to a daily or spill backup volume. After the backup profile is created, it does not have to be changed. DFSMShsm uses the backup profile for authorization checking during recovery if the discrete profile for the original data set no longer exists or if the authorization check against the original profile fails.

DFSMShsm also uses the backup profile for authorization checking during HALTERDS and HBDELETE processing if the authorization check against the original profile fails.

You use the PROFILEBACKUP or NOPROFILEBACKUP parameter of the SETSYS command to specify whether DFSMShsm creates backup profiles. If you specify PROFILEBACKUP, the backup prefix defined to DFSMShsm or the default backup prefix of HSM must be a user ID or group defined to RACF.

DFSMShsm maintains only one backup profile for each cataloged, RACF-indicated, VSAM or non-VSAM data set, no matter how many backup versions of the data set are maintained. The backup profile is always associated with the latest backup version of a data set that was RACF-indicated and cataloged at the time of backup. If you delete the latest backup version of a cataloged, RACF-indicated data set through the BDELETE or HBDELETE command, the backup profile is transferred to the next latest backup version of the data set that was made from a RACF-indicated, cataloged data set. DFSMShsm does not create backup profiles when it backs up uncataloged data sets.

If the DFSMShsm-started procedure is not associated with a RACF user ID, the user and group names in the backup profile will be "*".

DFSMShsm backs up cataloged, RACF-indicated data sets even if a backup profile is not created or transferred.

Profile Backup for Migrated Data Sets

If a migrated data set is being backed up, DFSMShsm uses the RACF profile of the data set as a model for creating the backup profile.

Deleting Backup Profiles

When DFSMShsm creates a new backup profile for a data set, it deletes all previously created backup profiles for the data set.

When backup versions of cataloged, RACF-indicated data sets become obsolete, DFSMShsm deletes their associated backup profiles even if you specified the NOPROFILEBACKUP parameter of the SETSYS command after the backup profiles were created.

Note: If you use RACF Release 1.6 or a subsequent release, you can distribute the location of backed up RACF profiles throughout the RACF data set.

Related reading: For additional information about distributing the location of backed up RACF profiles, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Parameters Used in Creating Backup Profiles

DFSMShsm's access control environment element (ACEE) is used when backup profiles are created. If the user ID associated with the DFSMShsm startup procedure has profile modeling active or is in a group for which profile modeling is active, any access list associated with the model profile is added to the backup profiles. If a profile is recovered that uses the backup profile as a model, the additional user IDs in the access list are attached to the recovered profile.

Selecting Daily Backup Volumes

When DFSMShsm selects a backup volume, it considers whether this is initial volume selection or subsequent volume selection.

If an error occurs during the selection process for a tape and the error is associated with a particular volume, DFSMShsm considers the tape backup volume unavailable for backup processing and selects another backup volume. Each time DFSMShsm is started, all nonfull backup volumes are considered available for backup processing.

The error handling and tape selection process differs when using the duplex tape option for your backup tapes.

Unless they are superseded by the BACKUPDEVICECATEGORY parameter of the BACKVOL or ADDVOL commands, the subparameters of the BACKUP parameter of the SETSYS command control the type of backup volume selected for output during backup processing:

- SETSYS BACKUP(ANY) specifies that any type of backup volume can be selected for output during backup processing.
- SETSYS BACKUP(DASD) specifies that only DASD backup volumes are selected for output during backup processing.
- SETSYS BACKUP(TAPE) specifies that only tape backup volumes are selected for output during backup processing. The automatic class selection (ACS) routines determine whether the backup function is directed to a system-managed tape library.
- SETSYS BACKUP(TAPE(*unittype*)) specifies that only tape backup volumes are selected for output during backup processing. The ACS routines determine whether the backup function is directed to a system-managed tape library.

When the backup function is not using a system-managed tape library, UNITTYPE specifies that only tape backup volumes capable of being mounted and written to on the specified type of unit are selected for output during backup processing.

DFSMShsm backs up users' data sets to daily backup volumes during volume backup. Daily backup volumes either are assigned by the storage administrator to a day in the backup cycle before DFSMShsm uses them or are assigned when DFSMShsm selects a volume added as a backup volume.

During backup of a level 0 volume, backup of migrated data sets, or movement of backup versions, DFSMShsm searches for a target volume as follows:

- It finds a volume already associated with the current day in the backup cycle that is neither full nor in use.
- It reuses a previously full volume.
- It assigns an additional volume.
- It waits for another task to finish using an acceptable volume.

DFSMShsm stops the search as soon as it finds an acceptable volume. If it does not find an acceptable volume, DFSMShsm stops processing the requested backup function.

DFSMShsm determines which daily backup volume to use by considering:

- The status of volumes previously used on this day in the backup cycle. This includes whether this is initial volume selection or subsequent volume selection.
- The backup attributes of the volume being backed up.
- The directions the storage administrator gives to DFSMShsm.

If an error occurs during the selection process and the error is associated with a particular volume, DFSMShsm considers the volume unavailable for backup processing and selects another daily backup volume. Each time DFSMShsm is started, the target volumes previously considered unavailable for backup processing are reinstated as available volumes.

When DFSMShsm initially selects a daily backup volume, it tries to find one that meets the following characteristics:

- The volume is already assigned to the current day in the backup cycle.
- The volume has the appropriate backup device category.
- The volume is not full.
- The volume is not being used by another task in this or another host.

After a volume backup task has obtained a daily backup volume, it continues to use that volume until the volume is no longer needed, the volume becomes full, or certain error conditions occur.

Now consider the processing that occurs when you specify the backup device categories of DASD, TAPE, or NONE.

Related reading:

- For additional information about the duplex tape option, see "The Duplex Tape Option" on page 402.

- For additional information about how to assign ACS routines and their affect on tape backup selection, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about tape backup selection, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Selecting DASD Daily Backup Volumes

For SMS-managed volumes, you specify a backup device category for the entire system with the BACKUP parameter of the SETSYS command. Suppose that you specified SETSYS BACKUP(DASD). In this environment, DFSMShsm uses DASD unless a command backup for a volume specifies another device category. DFSMShsm uses DASD daily backup volumes even if the storage administrator has specified the ADDVOL command to add tape daily backup volumes.

Selection of DASD daily backup volumes depends on the daily limit you specify for the backup cycle. The daily limit determines at what point volume backup stops assigning more DASD daily backup volumes to this day in the backup cycle and instead tries to reuse the full DASD daily backup volumes already assigned to this day in the backup cycle. The daily limit has little or no value when backup output is to tape.

Until the daily limit is reached, DFSMShsm simply attempts to associate another backup volume with the day in the cycle. DFSMShsm then reuses as many of these daily backup volumes as necessary during this day in the backup cycle as long as enough space is available on the DASD daily backup volumes. You use the BACKUP parameter of the DEFINE command to define the daily limit.

If you choose a daily limit that is too small, your system will have poorer performance because it spends too much time doing cleanup for the amount of space made usable. The daily limit is too small if cleanup does not make the daily backup volumes reusable.

If you choose a daily limit that is too large, too many volumes can be assigned to a particular day in the backup cycle; therefore, available space cannot be used because it is restricted to certain days in the backup cycle. The daily limit is too large if DFSMShsm uses only a small percentage of the daily backup volumes assigned to the day in the backup cycle on a day when a normal amount of data is backed up.

To reuse DASD daily backup volumes that were previously full, DFSMShsm uses spill processing or cleanup processing. You specify spill processing with the SPILL parameter of the SETSYS command; only cleanup processing is done if you specify the NOSPILL parameter.

DFSMShsm reuses DASD daily backup volumes when the following conditions occur:

- Another volume is needed.
- The number of volumes assigned to this day in the backup cycle has already reached the daily limit or there were no unassigned DASD daily backup volumes to assign to this day in the backup cycle.
- All volumes assigned to this day in the backup cycle are either being used or were marked full after being used earlier during this day in the backup cycle.

If the reuse process has tried today to use all volumes not currently in use, an additional volume is assigned from the set of unassigned DASD backup volumes,

even though the number of assigned volumes exceeds the daily limit. If no unassigned DASD backup volume exists but at least one acceptable volume is being used, DFSMShsm waits for the task to finish using the acceptable volume.

Related reading:

- For additional information about spill processing, see "Backup Volume Spill Process" on page 200.
- For additional information about cleanup processing, see "Cleanup Processing" on page 204.

Selecting Initial DASD Daily Backup Volumes

The processing that occurs during initial selection of the DASD daily backup volumes depends on whether the daily limit has not been reached or the daily limit has been reached or exceeded.

Daily Limit Not Reached: If all DASD daily backup volumes are full or in use before the daily limit is reached, DFSMShsm does the following processing:

1. It assigns another DASD daily backup volume. When DFSMShsm assigns another DASD daily backup volume, it tries to select volumes in the following order:
   a. Unassigned DASD daily backup volume. DFSMShsm then assigns it to the current day in the backup cycle.
   b. Unassigned DASD backup volume. DFSMShsm then assigns it as a daily backup volume and to the current day in the backup cycle.
2. It reuses a full DASD daily backup volume. When no unassigned DASD backup volumes are available, DFSMShsm either spills or cleans up the full, unallocated, DASD daily backup volumes. DFSMShsm uses the first volume that falls below the threshold of occupancy that you specified with the THRESHOLD parameter of the ADDVOL command.
3. It waits for a DASD daily backup volume that is in use to become available. If the spill or cleanup process does not result in a usable DASD daily backup volume and another volume backup task is currently using a DASD daily backup volume, this volume backup task waits for a DASD daily backup volume to become available.

   In a multiple DFSMShsm-host environment, before a volume backup task enters the wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMShsm tries to select another DASD daily backup volume. If the DASD daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMShsm tries to select the DASD daily backup volume again.
4. It stops backup processing. If DFSMShsm cannot find a DASD daily backup volume and no other volume backup task is using a DASD daily backup volume, the current volume backup task ends because no DASD daily backup volume is available.

Daily Limit Reached or Exceeded: If no daily backup volume is available and the number of daily backup volumes assigned to the current day in the backup cycle is greater than or equal to the daily limit, DFSMShsm does the following processing:

1. It reuses a full DASD daily backup volume. When the daily limit has been reached or exceeded, DFSMShsm either spills or cleans up the full, unallocated,

DASD daily backup volumes. DFSMShsm uses the first volume that falls below the threshold of occupancy that you specified with the THRESHOLD parameter of the ADDVOL command.

2. It assigns another DASD daily backup volume. If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMShsm assigns another daily backup volume. A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are being used or have been cleaned up today. DFSMShsm tries to select volumes in the following order:
   a. Unassigned DASD daily backup volume. DFSMShsm then assigns it to the current day in the backup cycle.
   b. Unassigned DASD backup volume. DFSMShsm then assigns it as a daily backup volume and to the current day in the backup cycle.
3. It waits for a DASD daily backup volume that is in use to become available. If the spill or cleanup process does not result in a usable DASD daily backup volume and another volume backup task is currently using a DASD daily backup volume, this volume backup task waits for a DASD daily backup volume to become available.

   In a multiple DFSMShsm-host environment, before a volume backup task enters the wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMShsm tries to select another DASD daily backup volume. If the DASD daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMShsm tries to select the DASD daily backup volume again.
4. It stops backup processing. If DFSMShsm cannot find a DASD daily backup volume and no other volume backup task is using a DASD daily backup volume, the current volume backup task ends because no DASD daily backup volume is available.

Selecting Subsequent DASD Daily Backup Volumes

DFSMShsm uses the same procedures during subsequent DASD daily backup volume selection as are described in "Selecting Initial DASD Daily Backup Volumes" on page 194. However, after a volume backup task has obtained a daily backup volume, it continues to use that volume until the volume is no longer needed, the volume becomes full, or certain error conditions occur.

If a DASD daily backup volume does not contain enough available space for the backup version of the data set DFSMShsm is backing up or contains enough total available space but the space is too fragmented, DFSMShsm does not back up the data set. The space is too fragmented if DFSMShsm cannot allocate the backup version in five or fewer extents. When the allocation of the backup copy of a data set fails because the DASD daily backup volume does not have enough space for the backup version, DFSMShsm does one of the following:

- If the percent of occupied space on the DASD daily backup volume is less than the specified threshold of occupancy, DFSMShsm fails backup of the data set. Volume processing continues with the next eligible data set on the source volume.
- If the percent of occupied space on the DASD daily backup volume is greater than or equal to the specified threshold of occupancy, DFSMShsm considers the DASD daily backup volume to be full. When the current DASD daily backup volume becomes full, DFSMShsm releases it, selects another daily backup volume, and tries to back up the data set again. If DFSMShsm cannot back up

the data set a second time because the second daily backup volume does not have enough space, DFSMShsm fails backup of the data set. However, DFSMShsm does not consider the second DASD daily backup volume to be full. Volume backup processing continues with the next eligible data set on the source volume.

Selecting Tape Daily Backup Volumes

In the example system, the SETSYS BACKUP(TAPE) command was used to specify that all backups are to tape. In this environment, DFSMShsm uses only tape daily backup volumes unless a volume backup command specifies backup to DASD.

If you want DFSMShsm to use tape as a daily backup volume and you have not specified restrictions limiting selection to tape, you must define at least one tape for each of the appropriate days in the backup cycle by using the ADDVOL command. You do not have to define tapes as daily backup volumes with the ADDVOL command if you have restricted daily backup volumes to tape by:

- Specifying BDC(TAPE) on a BACKVOL command
- Specifying BACKUP(TAPE) on a SETSYS command

If you use cartridge-type tape backup volumes, DFSMShsm writes data on the tape in a single-file format.

DFSMShsm considers the following factors when it selects a tape backup volume:

- Whether this is initial or subsequent volume selection
- For subsequent volume selection, the type of volume selection you specified with the SELECTVOLUME parameter of the SETSYS command: specific or scratch.

Related reading: For additional information about the selection of tape volumes that are in a system-managed tape library, refer to the *z/OS DFSMShsm Implementation and Customization Guide*.

Selecting Initial Tape Daily Backup Volumes

If you have no partially filled or empty tape daily backup volumes already assigned to the day in the backup cycle, DFSMShsm does the following processing:

1. DFSMShsm assigns another tape daily backup volume. When these volumes are assigned, DFSMShsm tries to select volumes in the following order:
   a. Unassigned tape daily backup volume. DFSMShsm then assigns it to the current day in the backup cycle.
   b. Unassigned tape backup volume. DFSMShsm then assigns it as a daily backup volume and to the current day in the backup cycle.

   If no unassigned backup volumes are available, DFSMShsm requests a scratch tape.
2. DFSMShsm waits for a tape daily backup volume that is in use to become available. If no scratch tape is available and another volume backup task is currently using a tape daily backup volume, this volume backup task waits for a tape daily backup volume to become available.

   In a multiple DFSMShsm-host environment, before a volume backup task enters the wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMShsm tries to select another tape daily backup volume. If the tape daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another

volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMShsm tries to select the tape daily backup volume again.

3. DFSMShsm stops backup processing. If DFSMShsm cannot find a tape daily backup volume and no other volume backup task is using a tape daily backup volume, the current volume backup task ends because no tape daily backup volume is available.

Selecting Subsequent Tape Daily Backup Volumes

After a volume backup task has obtained a tape daily backup volume, it continues to use that volume until the volume is no longer needed, the volume becomes full, or certain error conditions occur.

If end-of-volume is reached while DFSMShsm is writing on a tape daily backup volume, DFSMShsm considers the currently mounted tape daily backup volume to be full and selects another tape volume.

A backup version of a data set can span more than one tape backup volume. DFSMShsm fails the backup of a data set if the data set spans more than 40 tape backup volumes.

When DFSMShsm reaches end-of-volume while writing on a tape daily backup volume, the processing that occurs depends on whether you want DFSMShsm to select a volume from tapes already known to it, or to select a scratch tape volume.

If you specified the SELECTVOLUME(SPECIFIC) or SELECTVOLUME(BACKUP(SPECIFIC)) parameter of the SETSYS command, DFSMShsm selects a tape backup volume that meets the following conditions:

- The volume is added to DFSMShsm as a tape backup volume. Use the ADDVOL command to add the volume to DFSMShsm.
- The volume is empty.
- The volume is not currently being used.
- The volume has the same device type and density as the previous volume.

If DFSMShsm cannot find a volume that meets the criteria just specified or if you specified SETSYS SELECTVOLUME(SCRATCH) or SELECTVOLUME(BACKUP(SCRATCH)), DFSMShsm requests that a scratch tape be mounted on the already allocated unit. When the operator mounts a scratch tape, DFSMShsm assigns it as a tape daily backup volume and assigns it to the current day in the backup cycle.

After the operator mounts the volume, DFSMShsm continues backing up the current data set to that volume.

Certain error conditions, such as an error in writing on the tape, cause DFSMShsm to switch tape daily backup volumes. If an error occurs, DFSMShsm performs the following actions:

- Marks the tape daily backup volume as full
- Releases the tape volume and deallocates the unit
- Performs initial selection to select another tape daily backup volume

Selecting DASD and Tape Daily Backup Volumes

If you specified BACKUP(ANY) on the SETSYS command, DFSMShsm selects the first non-full, unallocated, daily backup volume assigned to the current day in the backup cycle, regardless of its device category. If you want DFSMShsm to use tape

as a daily backup volume and you have not specified restrictions limiting selection to tape, you must define at least one tape for each of the appropriate days in the backup cycle by using the ADDVOL command.

Selecting Initial Daily Backup Volumes (DASD or Tape)

DFSMShsm can select either DASD or tape as a daily backup volume. The process that occurs during initial DASD or tape daily backup volume selection depends on whether the daily limit has been reached or exceeded or has not been reached.

Daily Limit Reached or Exceeded: If no daily backup volume is available and the number of daily backup volumes assigned to the current day in the backup cycle is greater than or equal to the daily limit, DFSMShsm does the following processing:

1. It reuses a full DASD daily backup volume. When no unassigned backup volumes are available, DFSMShsm either spills or cleans up the full, unallocated, DASD daily backup volumes. If you specified SETSYS SPILL, DFSMShsm spills the volumes. If you specified SETSYS NOSPILL, DFSMShsm only cleans up the volumes. DFSMShsm uses the first volume that falls below the threshold of occupancy (which you specified with the THRESHOLD parameter of the ADDVOL command). A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are either being used or are full and have been cleaned up today.

   If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMShsm requests a scratch tape. After a scratch tape is mounted, DFSMShsm obtains the volume serial number and internally adds the volume as a daily backup volume assigned to the current day in the backup cycle.
2. It assigns another daily backup volume. If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMShsm assigns another daily backup volume. When DFSMShsm assigns another daily backup volume, it tries to select volumes in the following order:
   a. Unassigned daily backup volume. DFSMShsm then assigns it to the current day in the backup cycle. This daily backup volume can be tape or DASD.
   b. Unassigned backup volume. DFSMShsm then assigns it as a daily backup volume and to the current day in the backup cycle. This backup volume can be tape or DASD.

   If no unassigned backup volume exists, DFSMShsm requests a scratch tape. When no unassigned backup volumes are available, DFSMShsm either spills or cleans up the full, unallocated, DASD daily backup volumes. If you specified SETSYS SPILL, DFSMShsm spills the volumes. If you specified SETSYS NOSPILL, DFSMShsm only cleans up the volumes. DFSMShsm uses the first volume that falls below the threshold of occupancy (which you specified with the THRESHOLD parameter of the ADDVOL command). A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are either being used or are full and have been cleaned up today.

   If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMShsm requests a scratch tape. After a scratch tape is mounted, DFSMShsm obtains the volume serial number and internally adds the volume as a daily backup volume assigned to the current day in the backup cycle.
3. It waits for a daily backup volume that is in use to become available. If DFSMShsm cannot find a daily backup volume to use and another volume backup task is currently using an acceptable daily backup volume, this volume backup task waits for any type of daily backup volume to become available.

4. It stops backup processing. If DFSMShsm cannot find the type of daily backup volume it needs and no other volume backup task is using that type of backup volume, the current volume backup task ends because no daily backup volume is available.

Daily Limit Not Reached: If all daily backup volumes are full or in use before the daily limit for daily backup volumes is reached, DFSMShsm does the following:

1. It assigns another daily backup volume. When DFSMShsm assigns another daily backup volume, it tries to select volumes in the following order:
   a. Unassigned daily backup volume. DFSMShsm then assigns it to the current day in the backup cycle. This daily backup volume can be tape or DASD.
   b. Unassigned backup volume. DFSMShsm then assigns it as a daily backup volume and to the current day in the backup cycle. This backup volume can be tape or DASD.
2. It reuses a full DASD daily backup volume or requests a scratch tape. When no unassigned backup volumes are available, DFSMShsm either spills or cleans up the full, unallocated, DASD daily backup volumes. If you specified SETSYS SPILL, DFSMShsm spills the volumes. If you specified SETSYS NOSPILL, DFSMShsm only cleans up the volumes. DFSMShsm uses the first volume that falls below the threshold of occupancy (which you specified with the THRESHOLD parameter of the ADDVOL command). A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are either being used or are full and have been cleaned up today.

   If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMShsm requests a scratch tape. After a scratch tape is mounted, DFSMShsm obtains the volume serial number and internally adds the volume as a daily backup volume assigned to the current day in the backup cycle.
3. It waits for a daily backup volume that is in use to become available. If DFSMShsm cannot find a daily backup volume to use and another volume backup task is currently using an acceptable daily backup volume, this volume backup task waits for any type of daily backup volume to become available.

   In a multiple DFSMShsm-host environment, before a volume backup task enters a wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched and DFSMShsm tries to select another daily backup volume. If the daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMShsm tries to select the daily backup volume again.
4. It stops backup processing. If DFSMShsm cannot find the type of daily backup volume it needs and no other volume backup task is using that type of backup volume, the current volume backup task ends because no daily backup volume is available.

Selecting Subsequent Daily Backup Volumes (DASD or Tape)

DFSMShsm uses the same procedures during subsequent daily backup volume selection as those described in "Selecting Initial Daily Backup Volumes (DASD or Tape)" on page 198. However, after a volume backup task has obtained a daily backup volume, it continues to use that volume until the volume is no longer needed or becomes full, or until certain error conditions occur.

Selecting Subsequent DASD Daily Backup Volumes: If a DASD daily backup volume does not contain enough available space for the backup version of the data set DFSMShsm is backing up, or contains enough total available space for the

backup version but the space is too fragmented, DFSMShsm follows the procedures described in “Selecting Subsequent DASD Daily Backup Volumes” on page 195.

If you use both DASD and tape backup volumes, DFSMShsm can start backing up a volume to a DASD daily backup volume and continue with a tape daily backup volume. However, after DFSMShsm has selected a tape daily backup volume to back up a volume, it uses only tape backup volumes when additional backup volumes are required for the backup of the current volume.

Selecting Subsequent Tape Daily Backup Volumes: If end-of-volume is reached while DFSMShsm is writing on a tape daily backup volume, DFSMShsm follows the procedures described in “Selecting Subsequent Tape Daily Backup Volumes” on page 197.

Backup Volume Spill Process

Moving backup versions from DASD backup volumes to spill volumes is called *spill*. Spill is a process that applies *only* if you are using DASD devices to store your backup versions. If you use DASD as your daily backup volumes, your daily backup volumes eventually become filled with recent backup versions and older backup versions of the same data sets. The DASD daily backup volumes can also contain backup versions of deleted data sets and retired backup versions.

Note: Although both the spill and recycle process output to spill backup tapes, the spill process occurs only from DASD. Moving backup versions from tape backup volumes is part of recycle.

When adding DASD backup volumes to DFSMShsm with an ADDVOL command, you can specify a maximum percentage of the capacity of the volume above which the volume is considered to be full after spill or cleanup processing.This percentage is called the threshold of occupancy. When a DASD daily backup volume becomes full (an allocation has been failed for lack of space), it is *spilled* by the spill process. If the spill process does not reduce the occupancy of the volume below the threshold of occupancy, the volume is still considered to be full.

The spill process moves all but the latest backup version of a data set from a DASD daily backup volume to a spill backup volume. DFSMShsm owns the spill backup volume. During daily backup volume selection when all DASD daily backup volumes are full, DFSMShsm uses the spill process to make space available on the full DASD daily backup volumes. The spill process produces a DASD daily backup volume usable during volume backup processing. The spill process moves the following from daily backup volumes to spill backup volumes:

- All but the latest backup version of each data set
- All retired backup versions (see “Data Set Retirement” on page 95)
- All backup versions of a deleted data set

If a backup version is not valid, DFSMShsm does not move it. Instead, DFSMShsm scratches the invalid backup version.

DFSMShsm spills the full, unallocated, DASD daily backup volume that is assigned to the current day in the backup cycle and that has not been spilled in the longest period of time. If the volume spill process is successful, the DASD daily backup volume is used for the backup of the DFSMShsm-managed volumes. Otherwise, DFSMShsm spills the full, unallocated, DASD daily backup volume that has not been spilled in the next longest period of time. This process is repeated

until a usable DASD daily backup volume is produced or until every full, unallocated, DASD daily backup volume assigned to the current day in the backup cycle has been spilled on that day.

If cataloged and uncataloged data sets have the same name and each has backup versions, DFSMShsm does not move either the latest backup version of the cataloged data set or the latest backup version of the uncataloged data set.

When a DASD daily backup volume is spilled, it is also cleaned up. "Backup Volume Cleanup Process" on page 205 contains detailed information about cleanup processing. When the full DASD daily backup volume to be spilled has been allocated, DFSMShsm:

- Reads the volume table of contents (VTOC) of the DASD daily backup volume. As each data set VTOC entry is encountered, the name is checked to see if it is a backup version created by DFSMShsm. If it is, a check is made to see if this backup version is the latest backup version for the cataloged or uncataloged data set. If it is not the latest backup version, is a retired version, or is a version of a deleted data set and it is a valid backup version, DFSMShsm records that the backup version needs to be moved to a spill backup volume. If the data set on the backup volume either was not created by DFSMShsm or is a VTOC or VCAT copy, it is not processed.
- Selects a spill backup volume if any backup versions exist that should be moved. After DFSMShsm selects a spill backup volume, it begins to move the backup versions.
- Determines the percentage of allocated space on the daily backup volume after all the backup versions are moved to the spill backup volume. If the percentage of allocated space is less than or equal to the user-specified threshold of occupancy, the daily backup volume is no longer considered full and is used for the backup of the DFSMShsm-managed volumes. If the percentage of allocated space is not less than or equal to the threshold of occupancy, the daily backup volume is considered to still be full and is not selected as a target for backup versions. You can specify the threshold of occupancy by using the THRESHOLD parameter of the ADDVOL command.

Selecting Spill Backup Volumes

When DFSMShsm selects a spill backup volume, it considers whether this is the initial spill volume selection or a subsequent spill volume selection.

If an error occurs during the selection process for a tape and the error is associated with a particular volume, DFSMShsm considers the tape spill backup volume unavailable for spill backup processing and selects another spill backup volume. Each time DFSMShsm is started, all non-full backup volumes are considered available for backup processing.

The subparameters of the SPILL parameter of the SETSYS command control the type of spill backup volume selected for output during spill processing. DFSMShsm ignores the device category specified for backup when it performs spill processing:

- SETSYS SPILL(ANY) specifies that any type of spill backup volume can be selected for output during spill processing.
- SETSYS SPILL(DASD) specifies that only DASD spill backup volumes are selected for output during spill processing.

- SETSYS SPILL(TAPE) specifies that only tape spill volumes are selected for output during spill processing. The automatic class selection (ACS) routines determine whether the spill function is directed to a system-managed tape library.
- SETSYS SPILL(TAPE(*unittype*)) specifies that only tape spill volumes are selected for output during spill processing. The ACS routines determine whether the spill function is directed to a system-managed tape library.

  When the spill function is not using a system-managed tape library, UNITTYPE specifies that only tape spill volumes capable of being mounted and written to on the specified type of unit are selected for output during spill processing.

Related reading: For additional information about spill processing, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Initial Spill Backup Volume Selection (DASD and Tape)

For each full, DASD daily backup volume being spilled, DFSMShsm selects a spill backup volume to contain the backup versions to be moved. If you want DFSMShsm to use tape as spill backup volumes and you have not specified restrictions limiting selection to tape, you must define at least one tape as a spill volume by using the ADDVOL command. You need not define a tape as a spill volume if you restrict spill backup volumes to tape by specifying SPILL(TAPE) on a SETSYS command.

DFSMShsm selects the first non-full, unallocated spill backup volume to contain the backup versions to be moved. If a spill backup volume is not available, DFSMShsm selects an unassigned backup volume and assigns it as a spill backup volume.

Unassigned Backup Volume Exists: The type of unassigned backup volume selected depends on the type of volumes already assigned as spill backup volumes:

- If both tape and DASD spill backup volumes are defined, DFSMShsm selects any type of unassigned backup volume
- If only DASD spill backup volumes are defined, DFSMShsm selects only a DASD unassigned backup volume
- If only tape spill backup volumes are defined, DFSMShsm selects only a tape unassigned backup volume

No Unassigned Backup Volume Exists: If no unassigned backup volume exists, the actions taken depend on the type of spill backup volumes already defined:

- If both tape and DASD spill backup volumes are defined, DFSMShsm also uses the cleanup process in an attempt to produce a usable spill backup volume. However, if the cleanup process does not produce a usable spill backup volume, DFSMShsm requests a scratch tape. If a scratch tape is not available and another volume backup task is using a tape spill backup volume, this volume backup task waits for any spill backup volume to become available.
- If only DASD spill backup volumes are defined, DFSMShsm uses the cleanup process in an attempt to produce a usable spill backup volume. If the cleanup process is unsuccessful and another volume backup task is using a DASD spill backup volume, this volume backup task waits for a DASD spill backup volume to become available.

- If only tape spill backup volumes are defined, DFSMShsm requests a scratch tape. If a scratch tape is not available and another volume backup task is using a spill backup volume, this volume backup task waits for a tape spill backup volume to become available.

Wait for a Spill Backup Volume: In a multiple DFSMShsm-host environment, before a volume backup task enters a wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMShsm tries to select another spill backup volume. If the spill backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMShsm tries to select the spill backup volume again.

No Spill Backup Volume Available: If no spill backup volume is available and another volume backup task is not using a spill backup volume, this backup volume spill operation ends because no spill backup volumes are available.

Subsequent Spill Backup Volume Selection (DASD and Tape)

After a volume backup task has obtained a spill backup volume, the task continues to use that volume until the volume becomes full, the spill of the full DASD daily backup volume completes, or certain error conditions occur.

Subsequent DASD Spill Backup Volume Selection: If a DASD spill backup volume is being used and does not contain enough available space for the backup version currently being moved or contains enough total available space for the backup version but the space is too fragmented, DFSMShsm fails the movement of the backup version.DFSMShsm considers the space too fragmented if the allocation of the backup version cannot be done in five or fewer extents. When the new allocation for the backup version fails because of insufficient space for the backup version, DFSMShsm performs the procedures described in “Selecting Subsequent DASD Daily Backup Volumes” on page 195.

Note: In reading the procedures just referred to, substitute the term “spill” for the term “daily”.

Subsequent Tape Spill Backup Volume Selection: If end-of-volume is reached while DFSMShsm is writing on a tape spill backup volume, DFSMShsm performs the procedures described in “Selecting Subsequent Tape Daily Backup Volumes” on page 197.

Note: In reading the procedures just referred to, substitute the term “spill” for the term “daily”.

If DFSMShsm cannot find a tape volume meeting the preceding criteria or if you specified SETSYS SELECTVOLUME(SCRATCH), DFSMShsm requests that a scratch tape be mounted on the already allocated unit. When the operator mounts a scratch tape, DFSMShsm assigns it as a tape spill backup volume.

After the operator mounts the volume, DFSMShsm continues moving backup versions to that volume.

Certain error conditions, such as an error in writing on the tape, cause DFSMShsm to switch tape spill backup volumes.If an error occurs, DFSMShsm considers the tape spill backup volume to be full, releases the volume, and selects another tape

spill backup volume as described in "Initial Spill Backup Volume Selection (DASD and Tape)" on page 202, but only a tape volume is selected. After DFSMShsm selects a tape spill backup volume for the spilling of a full DASD daily backup volume, it uses only tape spill backup volumes when additional spill backup volumes are required during the current spilling of the full DASD daily backup volume.

Cleanup Processing

Cleanup processing is a subset of spill processing. When DFSMShsm does spill processing, it also does cleanup processing. DFSMShsm also does cleanup processing even when spill processing is not enabled.

DFSMShsm uses the cleanup process to scratch unwanted data sets on the DASD backup volumes.These data sets are:

- Out-of-date VTOC copy data sets created by DFSMShsm releases before Version 2 Release 3.0 and out-of-date VCAT copy data sets
- Unwanted backup versions

When DFSMShsm scratches a VTOC copy data set, it also uncatalogs the data set.

During volume backup processing, DFSMShsm creates a data set (the VTOC copy data set) that contains selected information from the VTOC of the volume that DFSMShsm is backing up. This data set drives the recovery of the volume. DFSMShsm maintains two copies of the VTOC copy data set. When DFSMShsm backs up a volume, the newly created VTOC copy data set becomes the latest VTOC copy data set, and the data set that was previously the latest VTOC copy data set becomes the next latest VTOC copy data set. As a result, these data sets become continually out of date as DFSMShsm keeps backing up the volume. If DFSMShsm encounters any of these out-of-date VTOC or VCAT copy data sets during volume cleanup processing, DFSMShsm scratches them and uncatalogs the VTOC copy data set.

Unwanted backup versions are those whose backup version record (MCC) has been deleted from the BCDS. The following are processes that can cause these unwanted backup versions:

- New backup versions are created and the MCC record for an old backup version is deleted, but the old version is not scratched.
- Use of the BDELETE command. The MCC record for the old backup version is deleted, but the old version is not scratched.

DFSMShsm does cleanup processing under three conditions:

- DFSMShsm selects a DASD daily or spill backup volume.
- No daily backup volume is available. That is, all daily backup volumes assigned to that day in the backup cycle are full, in use, or unavailable.
- No spill backup volume is available. That is, all spill backup volumes are full, in use, or unavailable.

DFSMShsm Selects a DASD Backup Volume: When DFSMShsm selects a non-full, unallocated, DASD daily or spill backup volume, DFSMShsm cleans it up if the volume has not been cleaned up on that day. DFSMShsm cleans up the volume even if the volume is not full.

No Daily Backup Volume Available: If you specified SETSYS NOSPILL, DFSMShsm cleans up the full, unallocated, DASD daily backup volumes. DFSMShsm starts with the full, unallocated, DASD daily backup volume that has not been cleaned up in the longest period of time. If the cleaned up DASD daily backup volume is usable, DFSMShsm backs up the level 0 volume to this DASD daily backup volume. Otherwise, DFSMShsm cleans up the full, unallocated, DASD daily backup volume that has not been cleaned up in the next longest period of time. This process continues until a usable DASD daily backup volume is produced or until every full, DASD daily backup volume has been cleaned up on that day. DFSMShsm does not clean up a DASD backup volume more than once a day.

No Spill Backup Volume Available: DFSMShsm cleans up the full, unallocated, DASD spill backup volume that has not been cleaned up in the longest period of time. If the cleaned up DASD spill backup volume is usable, DFSMShsm spills the full, DASD daily backup volume to this DASD spill backup volume. Otherwise, DFSMShsm cleans up the full, unallocated, DASD spill backup volume that has not been cleaned up in the next longest period of time. This process continues until a usable DASD spill backup volume is produced or until every full DASD spill backup volume has been cleaned up on that day. DFSMShsm does not clean up a DASD spill backup volume more than once a day.

Backup Volume Cleanup Process

After DFSMShsm allocates the DASD backup volume to be cleaned up, DFSMShsm does the following:

- It reads the VTOC. As each data set VTOC entry is encountered, DFSMShsm checks the name to see if it corresponds to that of a VTOC or VCAT copy data set. If it does, DFSMShsm checks to see if the data set is one of the current versions of a volume. If it is not current, DFSMShsm scratches the data set. If the name does not correspond to that of a VTOC or VCAT copy data set, DFSMShsm checks to see if it corresponds to that of a DFSMShsm-created backup version. If it does, DFSMShsm tries to read the backup version record (MCC) from the BCDS. If no backup version record exists, DFSMShsm scratches the backup version. If DFSMShsm does not recognize the data set name as that of a backup version, DFSMShsm does not process the data set.
- After DFSMShsm reads the VTOC and scratches all unnecessary backup versions, VTOC copy data sets, and VCAT copy data sets, it determines what the percentage of allocated space is on the DASD backup volume.If DFSMShsm is cleaning up a full DASD daily backup volume and the allocated space becomes less than or equal to the user-specified threshold of occupancy, DFSMShsm no longer considers the volume full. It uses the volume to back up a DFSMShsm-managed volume. If DFSMShsm is cleaning up a full DASD spill backup volume and the allocated space becomes less than or equal to the user-specified threshold of occupancy, DFSMShsm no longer considers that volume full. It uses the volume to spill a full DASD daily backup volume. You specify the threshold of occupancy with the THRESHOLD parameter of the ADDVOL command.

Note: When cleanup runs, the backup volume record (BVR) for the volume is held with an exclusive enqueue while the cleanup process runs. This enqueue can cause message ARC0712I to be issued during the cleanup process.

Releasing a Backup Volume

When a backup volume task finishes using a backup volume, it releases the backup volume by deallocating it. If the backup volume being released is not full and another volume backup task in the same host is waiting for the type of backup volume being released, the other volume backup task is notified that the backup volume is available. DFSMShsm determines the amount of unallocated space remaining on the volume if the backup volume is a DASD volume. If the released backup volume is a virtual volume and you have specified the UNLOAD parameter of the SETSYS command, DFSMShsm issues the MVS™ UNLOAD command to unload this volume.

Automatic Backup Example

Table 14 shows the conditions that pertain to a group of data sets at the beginning of automatic backup. See "Example Management Classes" on page 169 for a description of the management classes for these data sets. The names of the data sets are not precisely as the system would generate them. They have been shortened and modified for convenience in using them for illustrations.

Note: The management class name has been used as the high-order qualifier for the data set names for your convenience in comparing the conditions of the data sets with the attributes specified in the management classes. This is purely for educational purposes. You are not required to use the management class name as any qualifier in the data set name for SMS-managed data sets.

Table 14. Data Set Characteristics for Availability Management Example

Data Set Name	Number of Existing Backups	Last Backup (Days)	Data Set Changed Indicator	Data Set Was Backed Up
GDSMC.G3.V1	0	0	Yes	Yes
GDSMC.G3.V2	1	1	Yes	Yes
LARGEMC.G4.E1	4	3	Yes	Yes
LARGEMC.G4.E2	4	2	No	No
NOMIGMC.G4.G1	0	4	Yes	No
STANDMC.G4.H1	3	2	Yes	No

As DFSMShsm processes each of the data sets, the following occurs:

- Data set GDSMC.G3.V1 is backed up because the data-set-changed indicator is on and no backup version exists.
- Data set GDSMC.G3.V2 is backed up because the data-set-changed indicator is on and the data set has met the BACKUP FREQUENCY specified in management class GDSMC. However, because a backup version of the data set already exists and the new backup copy exceeds NUMBER OF BACKUPS (DATA SET EXISTS) specified in management class GDSMC, DFSMShsm invalidates the old backup version.
- Data set LARGEMC.G4.E1 is backed up for the same reason as for GDSMC.G3.V2. Note, however, that the days since last backup exceeds the value specified for BACKUP FREQUENCY in management class LARGEMC. This happened because the data set was last backed up on Friday. (Remember, the example system is running on a Monday.) Whether the data set changed on

Friday (after backup ran), Saturday, Sunday, or Monday, it could not be backed up until Monday. This data set also causes a roll-off of the oldest backup version.

- Data set LARGEMC.G4.E2 is not backed up because the data-set-changed indicator is set off.
- Data set NOMIGMC.G4.G1 is not backed up even though the data-set-changed indicator is on and the last backup meets the BACKUP FREQUENCY attribute in management class NOMIGMC. The data set is not backed up because the AUTO BACKUP attribute value is N in management class NOMIGMC.
- Data set STANDMC.G4.H1 is not backed up because, even though the data-set-changed indicator is on, the last backup days value does not meet the BACKUP FREQUENCY specified in management class STANDMC.

Command Availability Management

Previous sections have described how you can automatically cause DFSMShsm to dump and back up your data sets. However, you may want to dump a volume or back up data sets or volumes at a time other than when automatic availability management would cause the backup or dump. Also, when a data set or volume is damaged, either physically or logically, you want to access the dumped or backed up versions to replace the damaged data. Command availability management allows you to accomplish these functions. You can perform the following tasks using availability management commands:

- Dump a volume to either the dump classes specified in the command or the dump classes specified in the storage group
- Dump the volumes in one or more storage groups
- Back up an individual data set
- Use the data set backup by command capabilities
- Back up all eligible changed data sets on a volume
- Back up all eligible changed data sets in one or more storage groups
- Back up all supported data sets on a volume
- Back up all supported data sets in one or more storage groups
- Delete expired backup versions from the backup volumes
- Delete specific backup versions from the backup volumes
- Recover an individual data set from either a dump copy or an incremental backup version, whichever is the more recent
- Recover an individual data set from an incremental backup version
- Recover an individual data set from a dump copy
- Restore a complete volume from a dump copy and update the data sets from the latest incremental backup versions
- Restore a complete volume from a dump copy
- Recover a volume by using only incremental backup versions

Dumping a Volume

Issue one of the following commands to cause a command dump of one or more volumes:

```
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))
```

For all of the commands, DFSMShsm checks to ensure that SMS is installed and active in the system. If SMS is installed and active:

- For the first form of the command, the dump operates as explained in "Dump Processing" on page 178.
- For the second form of the command, dump processing is the same as is described in "Dump Processing" on page 178 except that the dump classes specified by the command are used instead of the dump classes specified by the storage group.
- For the third form of the command, dump processing is the same as is described in Dump Processing except that the dump classes specified by the command are used instead of the dump classes specified by the storage group, and the retention periods specified in the command are used instead of the retention periods specified by the dump classes.

Dumping the Volumes in One or More Storage Groups

To dump all the volumes in one or more storage groups, use the command:

```
BACKVOL STORAGEGROUP(sgn) DUMP
```

The processing for dumping each volume in a specified storage group is the same as explained in "Dumping a Volume" on page 207. As in that case, there is an additional available option for stacking dump copies as shown in the following example:

```
BACKVOL STORAGEGROUP(sgn) -
DUMP(DUMPCLASS(class,class,...,class) STACK(nn,nn,...,nn))
```

You can specify a stacking value, as follows:

- In the STACK parameter on the BACKVOL command
- If the STACK parameter is not specified on the BACKVOL command, in the definition of the dump class specified on the BACKVOL
- If the DUMPCLASS parameter is not specified, in the definition of a dump class associated with a specified storage group

If the stacking value is greater than 1, DFSMShsm stacks up to *nn* dump copies on the dump volumes of the dump classes used.

Backing Up an Individual Data Set

You can use one of the following data set backup commands to back up a single data set:

BACKDS *dsname*

HBACKDS *dsname*|*dsn_filter_name*

Note: You can also use the ARCINBAK program and the ARCHBACK macro to submit data set backup requests. DFSMShsm processes these requests according to descriptions in this section.

In processing these commands, DFSMShsm 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 HBACKDS 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 done.

The following process occurs when an individual data set is backed up to **DASD**:

1. Copy the data set to an ML1 volume. (If the data set is too large to fit on a single ML1 volume used for migration, it can be retried going to an ML1 OVERFLOW volume. If it does not fit there, then it fails.)
2. Build an MCM record in the BCDS to indicate that the backup copy must be moved to a backup volume.
3. Set the data-set-changed indicator to off and update the last backup date fields in the catalog entry and the BCDS MCB record.
4. Invalidate the oldest versions if the number of backup versions exceeds the value specified by the NUMBER OF BACKUPS (DATA SET EXISTS) management class attribute.

If the backup to DASD is unsuccessful, a message is issued.

The following process occurs when an individual data set is backed up to **tape**:

1. Copy the data set to a backup tape volume.
2. Set the data-set-changed indicator to off and update the last backup date fields in the catalog entry and the BCDS MCB record.
3. Invalidate the oldest versions if the number of backup versions exceeds the value specified by the NUMBER OF BACKUPS (DATA SET EXISTS) management class attribute.

If the backup to tape is unsuccessful, a message is issued.

DFSMShsm does not back up an SMS-managed data set if SMS is not installed and active in the system.

Using the Data Set Backup by Command Capabilities

The data set backup by command function provides the following capabilities:

- Up to 64 data sets per 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 demount continuously mounted backup tapes.
- Users can tailor the times when DFSMShsm demounts a tape.

Because the data set backup function allows users to target their backups directly to tape, this can possibly 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.

Specifying Data Set Backup Tasks

The data set backup commands allow a user to 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. It also allows you to 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** parameter allows you to 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. Defaults for this parameter are 3000 KB for the *maximum* size and 250 KB for the *standard* size. For a discussion about assigning values for *maximum* and *standard* sizes, see "Assigning DASDSELECTIONSIZE Values" on page 211.

The **DASD** parameter allows you to specify the number of concurrent data set backup tasks that can direct backups to ML1 DASD. The value *nn* for the TASKS subparameter ranges from 0 to 64 tasks. The command can take the following form:

```
SETSYS DSBACKUP(DASD(TASKS(nn)))
```

The **TAPE** parameter allows you to specify the maximum number of concurrent data set backup tasks that can direct backups to tape. The value *nn* for the TASKS subparameter ranges from 0 to 64 tasks. The command can take the following form:

```
SETSYS DSBACKUP(TAPE(TASKS(nn)))
```

Notes:

1. The DASDSELECTIONSIZE parameter allows you to balance the workload between DASD and TAPE tasks for all WAIT requests that do not specify output media. By setting appropriate values, you ensure that a sufficient number of backups go to each media type to keep them busy. Ultimately, you must weigh the efficient use of data set backup tasks against media use and the need to perform a second movement of backup versions from ML1 DASD to backup tapes during automatic backup processing.
2. The sum of both DASD and tape tasks may not exceed 64.
3. Because data set backups can be directed to tape, you may want to reassess the number of ML1 overflow volumes that you have defined to DFSMShsm. You may want to decrease that number.
4. A backup to tape is a one-time process, but a backup to DASD means that the data first goes to DASD before it moves to tape. To avoid this additional data movement, it is almost always more efficient to back up to tape. (An environment with small data sets that arrive infrequently may be the exception.)

Additional TAPE subparameters are MAXIDLETASKS and DEMOUNTDELAY. The MAXIDLETASKS option allows you to tell DFSMShsm the maximum number of idle tape tasks (0 to 64) that you will accept for each delay. The DEMOUNTDELAY option allows you to tell DFSMShsm how much time to wait before it deallocates idle tape drives.

Related reading: For additional information about the relationship between *maximum* and *standard* values associated with the DASDSELECTIONSIZE keyword and the SETSYS DSBACKUP command, see the *z/OS DFSMShsm Storage Administration Reference*.

Controlling Target Devices that Are Used by Data Set Backup

Because DFSMShsm can use both tape and DASD media types for data set backup requests, there is a need to balance existing workloads against available or allowed resources. In general, tapes take longer from initial selection to first write than DASD does, but tape may be faster in terms of data throughput. By tailoring the SETSYS DSBACKUP command to define the number of tasks that are available to the data set backup function, you can ensure that DFSMShsm efficiently directs your backup to either tape or DASD, depending on the workload.

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. 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*, then 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.

Note: If you specify the subparameter TASKS(0) for either DASD or tape, DFSMShsm never directs nontargeted requests to that media.

There are three distinct data set size ranges:

- A **small** data set is any data set with a size less than the value of *standard* in DASDSELECTIONSIZE*(maximum standard)*.
- A **medium** data set is any data set larger than a small data set but less than the value of *maximum* in DASDSELECTIONSIZE*(maximum standard)*.
- A **large** data set is any data set that is greater than a medium data set.

Notes:

1. There is a direct correlation between the number of defined tasks (tape and DASD) and the size of data sets that are directed to DASD. This correlation can result in double data movement during the automatic backup function. (See "Assigning DASDSELECTIONSIZE Values" for help with assigning size values to *maximum* and *standard*.) The automatic backup function can delay the start of the multitasked volume backup function; that is, volume backup will not begin until all backup versions on ML1 have been moved to tape. You can use the FREEVOL ML1BACKUPVERSIONS command to initiate movement of all backup versions off all ML1 volumes to daily backup volumes.
2. DFSMShsm starts tape tasks immediately on receipt of WAIT requests (medium and large) and NOWAIT requests, up to the TAPE (DEMOUNTDELAY(MAXIDLETASKS)) value.
3. When both tape and DASD tasks are allowed, NOWAIT requests are only processed by tape tasks.
4. DFSMShsm starts DASD tasks and tape tasks up to the allowable value when there is enough work for the tasks to do. New tasks are started that are based on the number of requests not currently being processed.

Assigning DASDSELECTIONSIZE Values

Small, medium, and large data set sizes are established when you specify the SETSYS DSBACKUP DASDSELECTIONSIZE*(maximum standard)* command. The values assigned to *maximum* and *standard* essentially help balance the workload between DASD and tape tasks. Default sizes for this parameter are 3000 KB for *maximum* and 250 KB for *standard* sizes.

The following considerations might help you decide what to specify for the *maximum* and *standard* values that are associated with the DASDSELECTIONSIZE parameter:

- Increasing either the *maximum* or *standard* value:
  - Increases the number of data sets that may be directed to DASD and decreases the number of data sets that may be directed to tape
  - Uses more ML1 DASD space
  - Increases the workload for autobackup because more backup versions must be moved from ML1 DASD to tape
- Decreasing the *maximum* or *standard* value:
  - Decreases the number of data sets that can be directed to DASD and increases the number of data sets that can be directed to tape
  - Uses less ML1 DASD space
  - Reduces the workload for autobackup since less backup versions must be moved from ML1 DASD to tape

Note: The default is DASDSELECTIONSIZE(3000 250), where 3000 is the maximum in KB, and 250 is the standard in KB. The categories in the table are based on these default values. If the you change the DASDSELECTIONSIZE(maximum standard) settings, then these categories change.

Table 15. Determining Data Set Sizes and Their Task Targets

Default Criteria	Large	Medium	Small
Data Set Size	3001 KB and up	251 – 3000 KB	0 – 250 KB
WAIT Requests	Favor tape	First available	Favor disk
NOWAIT Requests	Favor tape	Favor tape	Favor tape

Related reading: For additional information about how to determine the size of a data set and whether it will use tape or disk, see the *z/OS DFSMShsm Storage Administration Reference*.

Switching Data Set Backup Tapes

When you specify a value other than zero for SETSYS DSBACKUP(TAPE(DEMOUNTDELAY(MAXIDLETASKS(*n*)))), the data set backup function may allow output tapes to be continuously mounted for an indefinite amount of time. There may be times when you want or need to demount the tapes. You can use the SWITCHTAPES keyword and parameters to demount a continuously mounted tape. You may want to do so, for example, so that it or an alternate can be taken to another site for disaster backup, or so that you can install new tape devices. The SWITCHTAPES keyword is available on both the DEFINE and HOLD commands.

Note: If you specify the MARKFULL parameter of the DEFINE SWITCHTAPES(DSBACKUP) command, the tapes that are released from command data set backup are marked full. However, other partial tapes in the data set backup inventory (previous partial tapes) remain as selection candidates. These partial tapes may be immediately selected and mounted for output processing.

DEFINE SWITCHTAPES tells DFSMShsm that, at specific times or events, you want the data set backup function to automatically demount and deallocate all

currently mounted tapes and allocated drives. HOLD(BACKUP(DSCOMMAND)) tells DFSMShsm that you want to immediately quiesce data set backup activity to tape, demount all mounted tapes, and deallocate all tape devices. Therefore, if you plan ahead, you can use the DEFINE command to switch tapes. For unplanned activity, use the HOLD command to switch tapes.

Note: A tape can remain continuously mounted for a long period of time because both new tape technologies and a steady stream of data set backup requests allow an increased amount of data on the tape. It can also be mounted for a long time because the MAXIDLETASKS and DEMOUNTDELAY parameters on the SETSYS DSBACKUP command specify that tapes can be continuously mounted while inactive.

The DEFINE SWITCHTAPES(DSBACKUP) command allows you to choose among the following parameters:

TIME Indicates that you want to define a time when DFSMShsm initiates the demount and deallocation of all mounted tapes, performed at a certain time of day.

AUTOBACKUPEND Indicates that you want DFSMShsm to initiate the demount and deallocation of all mounted tapes at the end of the autobackup cycle.

PARTIALTAPE Indicates the method that DFSMShsm uses to mark an output tape as full. This is an optional keyword. MARKFULL, REUSE, and SETSYS are all valid options. SETSYS indicates that you want DFSMShsm to use the SETSYS PARTIALTAPE value.

The HOLD BACKUP(DSCOMMAND(SWITCHTAPES)) command tells DFSMShsm to perform SWITCHTAPES processing now; that is, to demount and to deallocate drives. DFSMShsm demounts active tapes at the end of the current data set, but immediately demounts idle tapes. The demounted tapes are marked full or not, according to the existing PARTIALTAPE parameter of the DEFINE SWITCHTAPES command. No subsequent RELEASE command is necessary. After these tapes are demounted and the drives are deallocated, data set backup continues with newly selected tapes, if requests are present.

Related reading: For additional information about the exact syntax of the DEFINE and HOLD commands, see the *z/OS DFSMShsm Storage Administration Reference*.

Taking a Backup Tape Away from a Data Set Backup Task

Depending on how you have defined your data set backup by command environment and the arrival rates of data set backup requests, backup output tapes may be mounted for extended periods of time. Consequently, DFSMShsm has a recover tape takeaway function. This function allows, for instance, the timely recovery of a data set when its backup was made in a previous job step.

For an explanation of backup tape contention possibilities and the scope of the recover tape-takeaway function, see Chapter 10, “Other Considerations for Availability Management,” on page 413.

Targeting Data Set Backups to Required Device Types

To target data set backups to a required target output device type, such as ML1 DASD or backup tape, use the following commands:

BACKDS *dsname* **TARGET(DASD)**

BACKDS *dsname* **TARGET(TAPE)**

TARGET DASD specifies that you want to back up the data set initially to ML1 DASD. TARGET TAPE specifies that you want to back up the data set directly to tape.

Users can specify the TARGET keyword on any of the data set backup commands. For example, the BACKDS *dsname* TARGET(DASD) NOWAIT command forces the backup copy to DASD. The TARGET keyword may override any device output or task that DFSMShsm might have selected on its own.

Using Concurrent Copy for Data Set Backup

The BACKDS command has the optional CC keyword, which allows you to specify concurrent copy for both SMS and non-SMS data sets.

BACKDS *dsname* **CC(PREFERRED)**

BACKDS *dsname* **CC(REQUIRED)**

PREFERRED indicates that you want to use concurrent copy as the backup method, if it is available. If concurrent copy is not available, the data set will be backed up as if this keyword were not specified.

REQUIRED indicates that concurrent copy must be used as the backup method and that the data set backup should fail if concurrent copy is not available.

You can also specify the STANDARD keyword, which indicates that you want to use a standard backup method (without using concurrent copy). The standard backup method may possibly override a management class option to use concurrent copy.

Note: For SMS data sets, DFSMShsm uses the management class values if you do not specify the CC parameter. For non-SMS data sets, DFSMShsm does not use concurrent copy if you do not specify the CC parameter.

There are two additional options: PHYSICALEND (PE) and LOGICALEND (LE).

- PHYSICALEND (the default) indicates that control returns to the application only after the backup has physically completed.
- LOGICALEND indicates that control returns to the application when the concurrent copy initialization completes. If PREFERRED was specified and concurrent copy is unavailable for the data set, the backup continues as if this keyword had not been specified. The BACKDS command fails if you specify both STANDARD and LOGICALEND keywords.

DFSMShsm uses the STGADMIN.ADR.DUMP.CNCURRNT facility class resource to limit concurrent copy use to those users who are authorized to use the CC option.

Authorizing the Use of Concurrent Copy: By default, all users can use the CC options. However, if RACF indicates a lack of authority, DFSMShsm fails the data set backup request if the concurrent copy request is specified as REQUIRED. If the concurrent copy request is not specified as REQUIRED and RACF indicates a lack of authority, DFSMShsm continues to back up the data set as if the concurrent copy keyword were not specified on the backup command.

Inline Backup of an Individual Data Set

Use the ARCINBAK program to back up individual data sets in a batch environment and to support the JCL back reference of data set names and relative GDG data set names. The processing for backing up individual data sets with ARCINBAK is the same as explained in "Backing Up an Individual Data Set" on page 208.

Note: You can use ARCINBAK on a system that is using a release prior to OS/390 V2R10 to submit jobs that use the new data set backup keywords. However, DFSMShsm processes the backup request as though there are no new specified keywords.

For information about how to use inline backup, see "Inline Backup" on page 415.

Backing Up All Data Sets on One or More Volumes

To back up all the data sets on a volume, use the command:

BACKVOL VOLUMES(*volser,...,volser*) TOTAL

The processing for backing up all the data sets on a volume by command is the same as that explained in "Phase 4: Backing Up SMS-Managed Volumes from Storage Groups with the Automatic Backup Attribute" on page 186 except:

- DFSMShsm does not retrieve a list of volumes to be backed up.
- DFSMShsm checks only the ADMIN OR USER COMMAND BACKUP attribute from the management class for each data set. The SMS-managed data set is eligible to be backed up unless the ADMIN or USER = NONE. All other values for this attribute and all other attributes in the management class are ignored.
- Non-SMS-managed data sets excluded by the ALTERDS or HALTERDS VERSIONS(0) command are not backed up.
- If you specify the BACKUPDEVICECATEGORY parameter of the BACKVOL command, the data sets are backed up to the device type you specify.

When the BACKVOL VOLUME command is issued for an SMS-managed volume, that volume is considered to be DFSMShsm-managed. DFSMShsm will maintain control records for the volume until the DELVOL command is issued against it.

Backing Up All Data Sets in One or More Storage Groups

To back up all the data sets in a storage group, use the command:

BACKVOL STORAGEGROUP(*sgn,...,sgn*) TOTAL

The processing for backing up all the data sets in a storage group by command is the same as is explained in "Phase 4: Backing Up SMS-Managed Volumes from Storage Groups with the Automatic Backup Attribute" on page 186, except for the following difference:

- DFSMShsm checks the AUTO BACKUP attribute from the management class for each data set. The SMS-managed data set is eligible to be backed up if AUTO BACKUP=YES. All other values for the command backup attribute and all other attributes in the management class are ignored.
- If you specify the BACKUPDEVICECATEGORY parameter of the BACKVOL command, the data sets are backed up to the device type you specify.
- When the BACKVOL STORAGEGROUP command is issued, the volumes processed are considered to be SMS-managed.

Backing Up All Eligible Changed Data Sets on One or More Volumes

You can back up all the eligible changed data sets on one or more volumes by issuing either of the following commands:

BACKVOL VOLUMES(*volser,...,volser*)

BACKVOL VOLUMES(*volser,...,volser*) INCREMENTAL

Processing for either of these commands is the same as that explained in "Phase 4: Backing Up SMS-Managed Volumes from Storage Groups with the Automatic Backup Attribute" on page 186, except:

- No list of volumes to process is retrieved
- If you specify the BACKUPDEVICECATEGORY parameter of the BACKVOL command, DFSMShsm backs up the data sets to the volume type you specify

Backing Up All Eligible Changed Data Sets in One or More Storage Groups

You can back up all of the eligible changed data sets in one or more storage groups by issuing either of the following commands:

BACKVOL STORAGEGROUP(*sgn,...,sgn*)
BACKVOL STORAGEGROUP(*sgn,...,sgn*) INCREMENTAL

Processing for either of these commands is the same as explained in "Backing Up All Eligible Changed Data Sets on One or More Volumes," where DFSMShsm determines each volume involved from the definitions of the specified storage groups.

DFSMShsm checks the AUTO BACKUP attribute of the management class for each changed data set, when the **STORAGEGROUP** parameter is used. The SMS-managed data set is checked for eligibility to be backed up only if AUTO BACKUP=YES. All other values for this attribute and all other attributes in the management class are evaluated.

Deleting Expired Backup Versions from the Backup Volumes

Availability management does not delete backup versions when a data set is deleted. To do so defeats the purpose of availability management. Thus, over time, a number of unneeded backup versions can accumulate on your backup volumes. You can use the following command to find and delete these unneeded backup versions.

EXPIREBV EXECUTE

This command causes DFSMShsm to search the BCDS for old, unwanted backup versions and to delete them based on attributes in the management class for each data set.The EXECUTE parameter of the command causes DFSMShsm to actually delete the BCDS records.

Note: While data sets are being changed, new versions are created and excess versions are rolled off (deleted). EXPIREBV provides a capability to apply the expiration criteria without the creation of a new version. If you modify only the maximum version limits but data sets are not backed up, only by

using the EXPIREBV (or explicitly by BDELETE) will the new expiration criteria be applied and expired versions deleted.

When EXPIREBV is used, two executions are required to delete unwanted versions. The first execution determines that the data set was scratched, and stores the date of the first EXPIREBV execution as the scratch date in the DFSMShsm control data set MCB record. A subsequent execution deletes the unwanted backup versions when the requested time has passed. The requested time is specified in the management class for SMS data sets, and in the EXPIREBV command for non-SMS data sets.

The EXECUTE parameter has an alternative DISPLAY parameter that causes DFSMShsm to list only the backup versions that would be deleted if the EXECUTE parameter were specified. The command also has STARTKEY and ENDKEY parameters to let you specify the range of data set names for which you want the BCDS records searched and a RESUME parameter to allow you to pick up where you left off if the EXPIREBV command was interrupted. An OUTDATASET | SYSOUT parameter allows you to specify the destination for the listing prepared when you specify the DISPLAY parameter. When you specify EXECUTE, the output list goes to the backup activity log.

For information about how the EXPIREBV command works with ABARS, see "DFSMShsm Coexistence Considerations for Aggregate Groups and Management Classes" on page 302 and "Expiring ABR Records in the BCDS" on page 320.

EXPIREBV Processing

The following terms are referred to in the following description: *date of scratch* or *scratch date*, *cataloged version*, and *uncataloged version*:

Programming Interface Information

- The date of scratch or scratch date refers to the date on which the EXPIREBV command determines that a data set for which a backup version exists has been deleted. The scratch date is generally not the date on which the data set was actually scratched, but is the date EXPIREBV was next run after the data set was scratched. The date is stored in the BCDS MCB record in the 4-byte field MCBSCRD (offset X'48', packed decimal). It can be displayed using the FIXCDS command.

  Note: DFSMShsm honors a scratch date stored by the DFSMShsm space maintenance and recovery utilities if the date is newer than the most recent backup date for the data set and if EXPIREBV has not already stored the scratch date itself.

End of Programming Interface Information

- A cataloged version refers to a backup version of a data set that was cataloged when the backup version was made; that is, the data set was cataloged when it was backed up.
- An uncataloged version is a version of a data set that was not cataloged when the version was made; that is, the data set was uncataloged when it was backed up.

Expire backup versions (EXPIREBV) processing is performed in eight steps. DFSMShsm:

1. Issues a starting message
2. Sets up the output destination

3. Retrieves a list of management class definitions
4. Determines the starting point for the scan of the BCDS
5. Determines the ending point for the scan of the BCDS
6. Processes each data set record in the BCDS between the starting and ending points
7. Ends the scan of the BCDS
8. Issues an ending message

Issue a Starting Message

If you specified DISPLAY in the EXPIREBV command, DFSMShsm sends a message to the user ID that issued the command. If you specified EXECUTE, DFSMShsm sends the starting message to the user and to the backup activity log.

Set Up the Output Destination

If you specified DISPLAY in the EXPIREBV command, DFSMShsm allocates and opens the output data set for the expire backup listing. DFSMShsm uses the OUTDATASET | SYSOUT parameter that you specified in the EXPIREBV command to allocate the data set. If an error occurs in allocating or opening the output data set, DFSMShsm issues a message, and EXPIREBV processing ends.

Retrieve a List of Management Class Definitions

The management classes for the SMS-managed data sets contain the attributes that determine whether any particular backup version should be deleted. At the start of EXPIREBV processing, DFSMShsm requests from SMS a list of the management classes that are defined to the configuration in which DFSMShsm is running. Successful return of the list of management class definitions indicates that SMS is installed and active. If SMS is not installed and active, DFSMShsm issues a message to indicate that EXPIREBV processing could not be done for SMS-managed data sets.

Determine the Starting Point for the BCDS Scan

The starting point for the scan of the BCDS depends on:

- Whether STARTKEY is specified in the command
- Whether RESUME is specified in the command
- Whether the key of the last data set processed (last-processed key) was saved at the end of the last EXPIREBV processing in this host

If STARTKEY is specified in the command, DFSMShsm starts with the BCDS record that the command specifies.

If RESUME is specified in the command and the saved last-processed key is not blank, DFSMShsm starts with the saved last-processed key.

If neither STARTKEY nor RESUME is specified in the command or if RESUME is specified but the saved last-processed key is blank, DFSMShsm starts with the first record in the BCDS.

Determine the Ending Point for the BCDS Scan

The ending point for the scan of the BCDS depends on:

- Whether ENDKEY is specified in the command.
- Whether RESUME is specified in the command.
- Whether a saved planned ending key exists from the previous EXPIREBV processing in this host. The planned ending key is saved at the beginning of EXPIREBV processing.

If ENDKEY is specified in the command, DFSMShsm ends when the next record to be processed has a higher key value than that specified with ENDKEY.

If neither ENDKEY nor RESUME is specified in the command or if RESUME is specified without ENDKEY and the saved ending key is blank, DFSMShsm ends EXPIREBV processing after the last BCDS record is processed.

If the saved planned ending key is not blank and RESUME is specified without ENDKEY in the command, DFSMShsm ends EXPIREBV when the next record to be processed has a higher value than the saved planned ending key.

Process Each Data Set Record in the BCDS

For each data set record in the BCDS, DFSMShsm performs the following processing:

- It determines if the version is a retired version. See "Data Set Retirement" on page 95 for a description of "delete if backed up". If the version is a retired version, DFSMShsm knows that it is not an SMS-managed data set and processes it according to rules for processing non-SMS-managed data sets. EXPIREBV processing for non-SMS-managed data sets is described in Chapter 7, "Availability Management of Non-SMS-Managed Storage," on page 261.

  If the version is not a retired version, DFSMShsm continues processing.
- It determines if the data set exists and is SMS-managed.

  For each data set that does not have a retired version and does not indicate that all backup versions are for an uncataloged data set and that does not have a scratch date recorded by prior EXPIREBV processing, DFSMShsm determines the names of the storage class, management class, and data class for the data set.

  If a catalog entry exists for the data set, the data set has not been scratched.

  If the data set is not cataloged and the MCB record does not contain a scratch date, DFSMShsm assumes that the data set has been scratched, and it records the current date in the scratch date field of the MCB record in the BCDS. DFSMShsm resets an existing scratch date to the current date if the date of the most recent backup version is the same or later than the existing scratch date. Because DFSMShsm cannot be sure, it assumes that the old scratch date is invalid. This is because the data set may have been recreated and backed up after prior EXPIREBV processing found that the data set had been scratched. This creates the possibility that more then one EXPIREBV command will need to be run to remove the backup versions, but it also prevents DFSMShsm from deleting valid backup versions. The scratch date is stored when either DISPLAY or EXECUTE is specified on the EXPIREBV command.

  DFSMShsm then checks the MCB record to determine if the data set was SMS-managed when it was last backed up. If the data set was SMS-managed when it was backed up, DFSMShsm processes all of its cataloged backup versions as described here. If it was not SMS-managed, DFSMShsm processes it as described in "Processing for Expired Backup Versions of Non-SMS-Managed Data Sets" on page 269.

  If any error occurs, other than being unable to find the data set, DFSMShsm assumes that the data set still exists and processes only the uncataloged versions of the data set as described in Processing for Expired Backup Versions of Non-SMS-Managed Data Sets.

  If the MCB record already contains a scratch date, DFSMShsm assumes that the data set has been scratched and determines from the MCB if the data set was SMS-managed when it was last backed up.
- It identifies expired backup versions.

Each SMS-managed data set's backup versions are processed based on the attributes specified in the management class associated with the data set (or in the default management class if the data set is not associated with a management class). If a management class name is found but there is no definition for the management class, DFSMShsm issues a message and does not process the cataloged backup versions of that data set.

After the management class has been established, processing depends on whether the data set exists:

- If the data set exists, the management class used is the one associated with the existing data set.
  1. DFSMShsm compares the number of backup versions that exist with the value of the NUMBER OF BACKUPS (DATA SET EXISTS) attribute in the management class. If there are more versions than requested, the excess versions are deleted, starting with the oldest version.
  2. Starting with the now-oldest backup version and ending with the *third-newest* version, DFSMShsm calculates the age of the version and compares the age to the value of the RETAIN DAYS EXTRA BACKUPS attribute. If the backup version is older, it is deleted.

     The *second-newest* version is treated as if it had been created on the same day as the newest backup version. Therefore, the second-newest version (newest EXTRA backup copy) will not be expired until the number of days specified in RETAIN DAYS EXTRA BACKUPS attribute have passed since the creation of the newest backup version.

     If only one backup version exists for the data set, it will not be deleted by the EXPIREBV.
- If the data set is deleted, the management class used is the one (if any) that was associated with the data set when the last backup version was made.

  If this is the first time EXPIREBV has found this data set to be deleted:
  1. DFSMShsm compares the number of backup versions to the value of the NUMBER OF BACKUPS (DATA SET DELETED) attribute and deletes any excess versions, starting with the oldest version.
  2. Starting with the now-oldest backup version and ending with the *third-newest* version, DFSMShsm calculates the age of each cataloged version and compares it with the value of the RETAIN DAYS EXTRA BACKUPS attribute. If the backup version is older, it is deleted.

     The *second-newest* version is treated as if it had been created on the same day as the newest backup version. Therefore, the second-newest version (newest EXTRA backup copy) will not be expired until the number of days specified in RETAIN DAYS EXTRA BACKUPS attribute have passed since the creation of the newest backup version.

  If the scratch date has been set by prior EXPIREBV processing:
  1. DFSMShsm compares the number of backup versions to the value of the NUMBER OF BACKUPS (DATA SET DELETED) attribute and deletes any excess versions, starting with the oldest version.
  2. Starting with the now-oldest backup version and ending with the *third-newest* version, DFSMShsm calculates the age of each cataloged version and compares it with the value of the RETAIN DAYS EXTRA BACKUPS attribute. If the backup version is older, it is deleted.

     The *second-newest* version is treated as if it had been created on the same day as the newest backup version. Therefore, the second-newest version (newest EXTRA backup copy) will not be expired until the number of

days specified in RETAIN DAYS EXTRA BACKUPS attribute have passed since the creation of the newest backup version.

3. If the RETAIN DAYS EXTRA BACKUP attribute is NOLIMIT and the RETAIN DAYS ONLY BACKUP attribute is other than NOLIMIT, DFSMShsm calculates the difference between the current date and the scratch date, then compares the result to the value of the RETAIN DAYS ONLY BACKUP (DATA SET DELETED) attribute. If the difference between the current date and the scratch date is greater than the specified value for the attribute, DFSMShsm deletes all of the remaining backup versions.
4. If all of the extra backups are deleted, DFSMShsm calculates the difference between the current date and the scratch date and compares the result to the value of the RETAIN DAYS ONLY BACKUP (DATA SET DELETED) attribute. If the difference between the current date and the scratch date is greater than the specified value for the attribute, DFSMShsm deletes the single remaining backup version.

- It determines the disposition of the MCB record.

  If you specified the EXECUTE parameter in the EXPIREBV command, DFSMShsm can change the MCB record in the BCDS as follows:

  – If all backup versions are being deleted or no backup version existed when the EXPIREBV command first read the MCB record, DFSMShsm *deletes* the MCB record (only for SMS-managed data sets).
  – DFSMShsm *updates* the MCB record if either of the following is true:
    - The scratch date field has been changed.
    - One or more backup versions, but not all versions, are being deleted

  If you specified the DISPLAY parameter for EXPIREBV, DFSMShsm updates the MCB record only if the scratch date field has been changed (updates only that field in the record).

- It deletes the eligible backup versions.

  If you specified the EXECUTE parameter in the EXPIREBV command, DFSMShsm deletes a backup version as follows:

  – If the backup version resides on an ML1 volume, the version is scratched (with the appropriate erase-on-scratch indication) and the MCM and MCC records are deleted. In addition, if a discrete backup profile is associated with a deleted version, the profile is also deleted.
  – If the backup version resides on a non-removable DASD backup volume, the version is scratched (with the appropriate erase-on-scratch indication), and the MCC record is deleted. In addition, if a discrete backup profile is associated with a deleted version, the profile is also deleted.
  – If the backup version resides on a removable DASD volume, the backup version is invalidated at this time, but it is not scratched until the backup volume is selected for spill or cleanup processing. If the erase option is in effect for the version, the MCC record is updated to reflect that the data set should be erased when it is scratched. If the erase option is not in effect for the version, the MCC record is deleted.
  – If the backup version resides on a tape backup volume, the MCC record is deleted and the TTOC entry for the data set is invalidated.

- It reports on the eligibility of backup versions.

  If the DISPLAY parameter was specified in the EXPIREBV command, DFSMShsm creates an output line for every backup version that is eligible to be deleted.

If the EXPIREBV command specified EXECUTE, DFSMShsm sends ARC0734I messages to the backup activity log under control of the SETSYS ACTLOGMSGLVL value:

- FULL specifies that an ARC0734I message is sent for each backup version for which deletion is attempted, whether deletion is successful or not.
- EXCEPTIONONLY specifies that an ARC0734I message is sent only when a failure occurs in attempting to delete a backup version.
- REDUCED has the same effect as EXCEPTIONONLY.

If DFSMShsm is running in DEBUG mode when the EXPIREBV command is initiated, the output created is identical to the output when DFSMShsm is not in DEBUG mode, but no backup versions are deleted even if EXECUTE is specified.

Note: During the scan of the BCDS, backup versions associated with a data set name of "*bprefix*.BACK" may be skipped and not expired. The prefix qualifier is set based on the BACKUPPREFIX parameter of the SETSYS command or the UID specified in the startup procedure. Because backup versions with high level qualifiers of "*bprefix*.BACK" may not be expired and deleted, do not create data sets using this naming convention. Data sets not deleted during EXPIREBV processing are still eligible for deletion during rolloff of backup versions, or when a BDELETE or HBDELETE command is issued.

End the Scan of the BCDS

The EXPIREBV command ends as planned when the key of the next record is higher than the planned ending key. Because the planned ending key was reached, a blank key value is saved in the backup control record (BCR) as the last processed key and the planned ending key.

The key of the last MCB record processed is saved periodically in the backup control record for this host when the host is processing an EXPIREBV command. This saved last-processed key is used to enable RESUME processing across system outages. If the EXPIREBV command ends early, the key of the last MCB record processed is saved in the backup control record for the host before processing ends. This allows processing to begin at that key if RESUME is specified on the next EXPIREBV command that is issued in that host. The EXPIREBV command can be caused to end early by:

- A HOLD command with the ALL or EXPIREBV parameters. (A RELEASE EXPIREBV or RELEASE ALL command must be issued before another EXPIREBV command can be issued.)
- A SETSYS command with the NOBACKUP or EMERGENCY parameters.
- The STOP command.

Issue an Ending Message

If the EXPIREBV command specified DISPLAY, DFSMShsm sends a message to the user ID that issued the command.

If the EXPIREBV command specified EXECUTE, DFSMShsm sends a message to the user and to the backup activity log.

Deleting Backup Versions of Specific Data Sets

The EXPIREBV command allows you to delete all *expired* backup versions from all of your backup volumes. You can also delete the backup versions of specific data

sets. You can delete either specific versions or all versions of the data sets. Use the following commands to delete backup versions:

BDELETE (*dsname . . .***)**

BDELETE (*dsname . . .***) -**
VERSIONS(*backup version number . . .***)**

HBDELETE (*dsname . . .***)**

HBDELETE (*dsname. . .***) -**
VERSIONS(*backup version number . . .***)**

DFSMShsm-authorized users can issue the first two commands. Any user can issue the second two commands.

The first form of the commands deletes all backup versions of the data set. The second form of the commands deletes only those backup versions of the data set for which you specify the version number.

The last backup date field of the catalog entry is altered to a null value when all of the following apply:
- No backup versions exist
- The data set is currently cataloged
- The data set is SMS-managed
- The data set is not migrated

Transferring the Backup Profile from One Backup Version to Another

Discrete backup profiles are associated with only the most recent backup version of a data set. If the backup version being deleted is associated with a backup profile, DFSMShsm reads BCDS backup version records to determine if any remaining backup versions are of a cataloged, RACF-indicated data set.

If one of the backup versions is of a cataloged, RACF-indicated data set, the backup profile is transferred to the latest backup version of a cataloged, RACF-indicated data set.

Note: Backup profiles are not transferred to a backup version of an uncataloged data set.

Profile Creation: If the profile creation fails, message ARC0382I is issued, but the backup version associated with the backup profile is still deleted. If any backup profiles remain, they are not deleted.

Authorization Checking during HBDELETE Processing

During HBDELETE processing, DFSMShsm uses RACF to determine the user's authorization to process the data set. First, DFSMShsm makes an authorization check against the original data set name. If either a generic profile or a discrete profile cannot be found for a cataloged data set that was RACF-indicated at the time of backup, DFSMShsm determines whether or not a backup profile exists. If a backup profile exists, DFSMShsm makes a second authorization check with the backup profile name. If the second authorization check fails, the HBDELETE processing fails.

Command Recovery and Restore

You use the recovery and restore process to:

- Recover a data set that has been lost or damaged
- Access an earlier version of the data set without deleting the current version
- Restore a data set from a dump copy
- Restore a volume from a dump copy and update it from incremental backup versions
- Restore a volume from a full-volume dump
- Recover a volume from DFSMShsm backup versions

The recovered backup version can replace an existing data set, or you can rename the recovered copy so it exists in parallel with the current data set. You can recover or restore a data set or volume only by specifying the RECOVER command.

A backup extended format VSAM data set can be recovered only as an extended format data set.

If the data set was RACF indicated at the time that the dump or backup version was created or is RACF indicated at the time of the recovery or restore, DFSMShsm recovers a discrete profile for the recovered or restored data set. See "Recovering or Restoring RACF-Indicated Data Sets" on page 430 for a discussion of profile recovery.

If you want to use DFSMShsm recovery, you must have specified SETSYS BACKUP.

The recovery process has two restrictions:

- DFSMShsm cannot recover a data set to a migration volume
- DFSMShsm cannot recover a data set that is currently 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.

Data Set Recovery and Restore

You can use the following commands to recover a data set from a DFSMShsm backup version or dump copy:

- These two forms of the command recover a specific data set when no data set with the same name is cataloged in the system catalog. If there is a data set with the same name, the recovery fails.

  HRECOVER *dsname*

  RECOVER *dsname*

- These two forms of the command recover a specific data set but recover it under the fully qualified data set name specified by the NEWNAME parameter that is not currently cataloged in the system catalog. The original data set name may or may not exist in the system catalog.

  HRECOVER *dsname* **NEWNAME(***newdsname***)**

  RECOVER *dsname* **NEWNAME(***newdsname***)**

- These two forms of the command recover a specific data set under its original name, even though a data set with the same name is cataloged. The cataloged data set is replaced by the recovered data set.

HRECOVER *dsname* **REPLACE**

RECOVER *dsname* **REPLACE**

- These two forms of the command recover a specific data set to a fully qualified data set specified by the NEWNAME parameter, even though a data set is cataloged with the same name as is specified by the NEWNAME parameter. The recovered data set replaces the cataloged NEWNAME data set.

  HRECOVER *dsname* **NEWNAME(***newdsname***) -**
  REPLACE
  RECOVER *dsname* **NEWNAME(***newdsname***) -**
  REPLACE

The HRECOVER command can be used by any user; the RECOVER command can be used by DFSMShsm-authorized users.Three other parameters—DATE, GENERATION, and VERSION—can be specified that cause DFSMShsm to select a backup version or dump copy other than the most recent. The GENERATION and VERSION parameters are used to select a specific backup version. The DATE parameter allows the selection of either a dump copy or a backup version.

When the RECOVER or HRECOVER command is issued in any of the shown forms, DFSMShsm recovers the later of the most recent dump copy or the most recent incremental backup version. For a data set to be restored from a dump copy, the dump copy must have been made from a dump class that allows data set restore, and a VTOC copy must exist for the dump (except when an authorized user explicitly specifies a dump volume).

DFSMShsm's recovery process involves five steps:

- Determine if the data set being recovered currently exists on a level 0 volume
- Select a target volume
- Determine the storage class, management class, and data class to be associated with the data set
- Determine whether to recover a backup version or a dump copy
- Recover the data set to the target volume

Determining Whether the Data Set Exists

DFSMShsm considers the target data set to exist if the target data set name is cataloged. The target data set and target data set name refer to either the original name of the data set (if you did not specify a new name) or to the new name of the data set that you specified with the NEWNAME parameter of the RECOVER or HRECOVER command.

Selecting the Target Volume for Incremental Backup Version Recovery

When SMS is active, it selects the target volume for the data set to be recovered from an incremental backup version.

Selecting the Target Volume for Physical Dump Copy Restore

Since DFSMSdss does not invoke SMS to select the target SMS volume for a physical data set restore, DFSMShsm selects the final target volume and passes that volume to DFSMSdss for a data set restore from a physical dump copy. The order of precedence of DFSMShsm target volume selection is:

1. The source volume serial number of the dump copy, if FROMDUMP(DUMPVOLUME(*volser*)) is specified for a multiple-volume data set
2. The volume serial number to which the target SMS data set is currently cataloged
3. The volume serial number of the TOVOLUME parameter specified on the RECOVER command
4. The volume serial number of the FROMVOLUME parameter specified on the RECOVER command
5. The volume serial number to which the target non-SMS data set is currently cataloged
6. The source volume serial number of the dump copy, if FROMDUMP(DUMPVOLUME(*volser*)) is specified on the RECOVER command
7. The source volume serial number of the incremental backup version
8. The volume serial number to which the original data set is currently cataloged, if the NEWNAME data set is not currently cataloged and an incremental backup version is not available

If the source and selected target device types are not similar, or if the target data set and the target volume are incompatible (that is, one is SMS managed and the other is non-SMS managed), DFSMShsm fails the restore of the data set from the dump copy. If this happens, you may be required to specify the target volume, either by cataloging the target data set or by using the TOVOLUME parameter of the RECOVER command.

Determining the Storage Class, Management Class, and Data Class

Determination of the SMS classes associated with the data set depends on whether SMS is active, whether the target data set exists, and whether FORCENONSMS is specified.

When SMS Is Not Active: When SMS is not active, any data set that currently exists and is SMS-managed or that was SMS-managed when it was backed up or dumped cannot be processed unless FORCENONSMS is specified on the RECOVER command. For a discussion of using the FORCENONSMS parameter, see "Recovering Specific Data Sets" on page 271.

When SMS Is Active and the Target Data Set Exists: If the target data set exists, DFSMShsm uses the storage class, management class, and data class from the catalog entry as the SMS classes to be associated with the recovered data set.

When SMS Is Active and the Target Data Set Does Not Exist: If the target data set does not exist, DFSMShsm invokes the ACS services of DFSMSdfp to determine the SMS classes to be associated with the recovered data set. In invoking ACS services, DFSMShsm passes to the ACS routine of DFSMSdfp any of the following that are available:

- The name of the data set.
- The name of the storage class associated with the data set when the data set was incrementally backed up.

  No storage class is passed if one of the following is true:

  – The data set was not SMS-managed when it was incrementally backed up.
  – A physical dump copy is being explicitly recovered because you specified FROMDUMP or FROMDUMP DUMPVOLUME.

  - The data set does not have an incremental backup version.

    Note: If the data set was SMS-managed when it was physically dumped, and your ACS storage class routine expects DFSMShsm RECOVER to provide the storage class in effect when the data set was dumped, the data set may be filtered as non-SMS-managed, and the recovery may fail with an "unable to recover a non-SMS-managed data set to an SMS-managed volume" error. If this occurs, use FORCENONSMS or TOVOLUME to recover the data set to a non-SMS-managed volume.
- The name of the management class associated with the data set when the data set was incrementally backed up. No management class is passed if one of the following is true:
  - The data set used the default management class when it was incrementally backed up.
  - The data set was not SMS-managed when it was incrementally backed up.
  - A physical dump copy is being explicitly recovered because you specified FROMDUMP or FROMDUMP DUMPVOLUME.
  - The data set does not have an incremental backup version.
- The name of the data class associated with the data set when the data set was incrementally backed up. No data class name is passed if one of the following is true:
  - The data set had no data class when it was incrementally backed up.
  - The data set was not SMS-managed when it was incrementally backed up.
  - A physical dump copy is being explicitly recovered because you specified FROMDUMP or FROMDUMP DUMPVOLUME.
  - The data set does not have an incremental backup version.
- The volume serial number specified with the TOVOLUME parameter in the request or the volume serial number and unit type of the volume where the data set originally resided.

  Note: If your storage class ACS routine checks for a valid DASD unit name, be sure to include any unit names that may have existed in your installation at one time and are no longer in use, because the data set could have resided on such a unit when it was backed up.
- An environment indicator of RECOVER.
- 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 indication.
- The data set organization.
- The record organization for VSAM data sets.

ACS services returns the SMS class names to be associated with the recovered data set. If ACS services does not return a storage class name, the data set is not SMS-managed and is recovered as a non-SMS-managed data set.

Determine Whether to Use Backup or Dump Versions for Data Set Recovery

The non-DFSMShsm-authorized user has little control over whether a restore or a recover is done. The user's ability to cause a physical data set restore rests with the options set for the dump classes and the dump volumes. Unless the user who issues a RECOVER or HRECOVER command specifies the GENERATION,

VERSION, or DATE parameter, DFSMShsm tries to recover the most recent dump copy or backup version available (based on time of day and date).

The following conditions must exist for a data set restore to be performed instead of a recover of an incremental backup version when a RECOVER or HRECOVER command is issued:

- The data set must be currently cataloged or must have been incrementally backed up by DFSMShsm at some time. If the data set is currently cataloged and no backup version exists for it, it must have been dumped from the volume it is cataloged on.
- If FROMVOLUME is used to direct DFSMShsm to restore a data set from a dump copy, the dump copy made from the FROMVOLUME specified is used, regardless of the catalog status of the data set when the dump copy was made. The data set is restored to the FROMVOLUME specified and left uncataloged.
- An eligible dump class must exist that allows individual data set restores (that is, the dump class must have the DATASETRESTORE attribute.)
- The data set to be restored must have resided on a volume dumped to an eligible dump copy. See "Searching for Eligible Dump Copies" on page 229 for detailed information.
- The dump volumes on which the dump copy resides must have a retention period other than NOLIMIT. DFSMShsm assumes that the dump volumes are physically available to be mounted when the RECOVER command is issued.
- The data set must be listed in the dump VTOC copy data set associated with the dump generation in which the dump copy is contained.
- The eligible dump copy must have been created more recently than the most recent incremental backup version. If a date is specified on the HRECOVER or RECOVER command, the recover is done from the backup version or dump copy that was performed *before or on* the specified date.
- A specific generation or version must not have been requested. A generation for a data set refers only to the DFSMShsm incremental backup versions of the data set and not to the dump copies of the source volume.
- The volume receiving the data set and the original volume from which the dump copy was made must be of similar device type.
- The volume receiving the data set and the data set being restored must be compatible in terms of being SMS-managed. DFSMShsm will neither restore an SMS-managed data set to a non-SMS-managed volume, nor restore a non-SMS-managed data set to an SMS-managed volume.
- The version of DFSMSdss installed must be at a sufficient level to support SMS.
- The volume receiving the data set and the data set itself must both be SMS managed or non-SMS managed. DFSMShsm does not restore an SMS-managed data set to a non-SMS-managed volume, or a non-SMS-managed data set to an SMS-managed volume.

Notes:

1. DFSMShsm always uses the backup version of SMS-managed, multiple-volume data sets unless you specify that the data set is to be restored from a dump copy.
2. See "Restoring a Data Set from a DFSMShsm Dump Copy" on page 235 for a discussion of specifying data set restore from a dump copy.
3. Only an authorized user can specify restore from a dump copy.

If any of the conditions are not met, the HRECOVER or RECOVER process continues by attempting to find a DFSMShsm backup version to recover. If no DFSMShsm backup version exists, the HRECOVER or RECOVER request fails.

To further identify the backup version or dump copy you want to recover, specify a date or a relative generation number for the backup version. When you identify a backup or dump version with a date, DFSMShsm recovers the last version DFSMShsm created on or before the date you specified. When you identify a backup version with a generation number, the number represents the number of backup versions created before the most recent one. A GENERATION(0) indicates the most recently created backup version. If DFSMShsm requires access to a level 0 volume, daily backup volume, or spill backup volume during recovery and that volume is not online, DFSMShsm dynamically allocates the volume. If the backup version is on more than one tape volume, DFSMShsm allocates each required volume and the operator mounts each required volume.

Searching for Eligible Dump Copies

If an HRECOVER command is not restricted to recovering from incremental backup volumes, or if a RECOVER command is not restricted from recovering backup versions or a specified dump volume, DFSMShsm must determine the volume from which the data set may have been dumped, in order to find an eligible dump copy. The following conditions narrow the search:

- If the data set is currently cataloged, DFSMShsm begins with the volume the data set is currently cataloged on as the original source volume. Dump copies of this volume are searched to try to find an eligible copy. If no eligible copy can be found, and this data set also has a valid DFSMShsm backup version, and if this backup version was created from a different volume from the one on which the data set is currently cataloged, the search continues with dump copies of the volume last backed up from.
- If the data set is not currently cataloged, it must have a valid DFSMShsm backup version to determine the original source volume. The volume where the data set was located, and where incremental backup last found the data set, is used to locate an eligible dump copy.
- If FROMVOLUME is used to direct DFSMShsm to restore a data set from a dump copy, the dump copy made from the FROMVOLUME specified is used, regardless of the catalog status of the data set when the dump copy was made. The data set is restored to the FROMVOLUME specified and left uncataloged.

If an eligible dump copy is not found that is more recent than an eligible backup version, the data set is recovered from the backup version.

Recovering the Data Sets to SMS-Managed Volumes

The recovery action taken depends on whether a DFSMShsm backup version or a dump copy is used to recover the data set, and on whether the data set being recovered is a VSAM or non-VSAM data set. If you are recovering an existing data set, you must specify either REPLACE or NEWNAME. If you specify REPLACE, DFSMShsm replaces the existing data set with the recovered data set. If you specify NEWNAME, DFSMShsm recovers the data set with the NEWNAME data set name.

If an out-of-space condition causes the recovery of an SMS-managed data set to fail, the RECOVER DFDSSOPTION(VOLCOUNT(N(*nn*) | ANY)) command can be used to ensure the successful recovery of the data set. The RECOVER DFDSSOPTION(VOLCOUNT(N(*nn*) | ANY)) command overrides the environment

established by the SETSYS VOLCOUNT command. If you don't know what your environment is, use the QUERY SETSYS command to determine the current operating environment.

Recovering a Non-VSAM Data Set from a DFSMShsm Backup Version: If the target data set exists, DFSMShsm deletes the existing data set. DFSMShsm passes the storage class name associated with the data set to SMS or DFSMSdss. SMS selects the volume and allocates the target data set on that volume. DFSMShsm or DFSMSdss then copies the backup version into the recovered data set.

The catalog status of a recovered generation data set (GDS) is dependent upon its catalog status before being recovered. If the GDS is not cataloged before recovery, its catalog status after recovery is DEFERRED. If the GDS is cataloged before recovery, its catalog status after recovery remains the same, with the following exception: when DFSMShsm is used as a data mover to recover a rolled-off GDS, the cataloged status of the GDS, after it is recovered, is DEFERRED. Use the IDCAMS ALTER ROLLIN command to change the status from DEFERRED to ACTIVE.

If the backup version spans multiple tape volumes, DFSMShsm issues a message to the operator asking if all necessary tapes are available. If they are not all available, DFSMShsm cannot recover the data set.

If DFSMShsm requires access to a level 0 volume, a daily backup volume, or a spill backup volume during recovery and if that volume is not on line, DFSMShsm dynamically allocates the volume. If the backup version is on more than one tape volume, DFSMShsm allocates a unit where the operator mounts each required volume.

Recovering a VSAM Data Set from a DFSMShsm Backup Version: If the target data set exists, DFSMShsm deletes the existing data set, uses the Access Method Services IMPORT command or DFSMSdss as the data mover, and passes the storage class name associated with the data set. Access Method Services or DFSMSdss determines the volume to which the data set is recovered and creates space for the data set.

DFSMShsm recovers VSAM data sets to a level 0 volume by using the Access Method Services IMPORT command or DFSMSdss as the data mover. However, because the backup version of a VSAM data set contains DFSMShsm control information, a user-issued IMPORT command or DFSMSdss command cannot be used to recover a backup version.

If DFSMShsm recovers a VSAM data set to a volume on a type of device that is different from the device from which DFSMShsm backed up the VSAM data set, the allocation for the buffer that the Access Method Services IMPORT command uses could be too small. If this occurs, you can define the target VSAM data set (base cluster and all alternate indexes) and then reissue the command.

The recovery of an integrated catalog facility catalog is restricted to those users who are recognized as DFSMShsm-authorized users. If TOVOLUME is specified, the volume specified must have the same volume serial number and device type as the volume from which the catalog was backed up. For more information about recovering the integrated catalog facility catalog, see "Backing Up and Recovering a Catalog" on page 440.

The NEWNAME parameter limits the recovery of VSAM data sets to spheres that do not have two or more paths defined on the base cluster, and recovers only the base cluster even when alternate indexes exist. After DFSMShsm recovers the base cluster, you can rebuild any necessary alternate indexes using the Access Method Services BLDINDEX command.

For VSAM data sets not cataloged, if the data set with the new name does not exist, DFSMShsm generates the new names for the data, index, and path components based on the character length of the new user-supplied cluster name:

New Cluster Name Length	Component Naming Scheme
1–38 characters	*cluster*.DATA, *cluster*.INDEX, *cluster*.PATH
39–42 characters	*cluster*.D, *cluster*.I, *cluster*.P
43–44 characters	*fourqual*.D*hhmmss*, *fourqual*.I*hhmmss*, *fourqual*.P*hhmmss*

Where *cluster* is the new user-supplied cluster name, *fourqual* is the first four qualifiers of the new user-supplied cluster name, and *hhmmss* is a time stamp.

If the data set with the new name already exists and the type (KSDS, ESDS, or RRDS) is the same, DFSMShsm reuses the entry names of the data component and index component. For information about how DFSMSdss generates new names for the data and index components, refer to the *z/OS DFSMSdss Storage Administration Guide*.

A message is printed containing the generated or existing name.

If you specified the NEWNAME parameter and the REPLACE parameter and if a data set with the name you specified already exists, DFSMShsm deletes the existing data set before it recovers the data set you specified with the RECOVER command.

The new data set name and the old data set name can be cataloged in the same or different catalogs when you specify the NEWNAME parameter.

For VSAM data sets not cataloged, if you specify NEWNAME the data and index objects of the data set being recovered must be the same as the object names exported. The object name of the data set being recovered is passed to Access Method Services IMPORT to rename the NEWNAME data set. Access Method Services verifies if the name is the same as the one exported.

Restoring a Non-VSAM Data Set from a Dump Copy: If the target data set exists, DFSMShsm deletes the data set. DFSMShsm invokes DFSMSdss with the storage class name and management class name associated with the data set. DFSMShsm also passes the target volume serial number selected by the criteria mentioned on page 225. With the information that DFSMShsm passed to it, DFSMSdss then restores the data set to that volume.

Note: Use the FORCENONSMS keyword to recover the data set to a non-SMS managed volume.

If the target data set does not exist, DFSMShsm invokes DFSMSdss with the storage class name and management class name associated with the data set when it was incrementally backed up. If the data set does not have an incremental

backup version, neither the storage class name nor the management class name is passed to DFSMSdss, and the data set may be restored as non-SMS-managed.

If you specified the NEWNAME parameter in the command and a data set with that name exists, DFSMShsm deletes the existing data set. DFSMShsm then invokes DFSMSdss to restore the NEWNAME data set. DFSMSdss restores the data set to the volume passed by DFSMShsm. If the restore is successful, DFSMShsm catalogs the restored data set.

Restoring a VSAM Data Set from a Dump Copy: If the data set does not exist on a level 0 volume, you can create it. If a backup version of the data set exists, you need not create the data set. DFSMShsm performs a recovery of the backup version (which creates the data set) then invokes DFSMSdss to restore the data set. If you are a DFSMShsm-authorized user, DFSMShsm still attempts to restore the data set even if it is not cataloged and no backup version can be found. The restored data set is not cataloged. In this case, you must use the Access Method Services DEFINE RECATALOG command to recatalog the data set. If a new name was specified, it is not used; the data set is restored with its original name. If you are not a DFSMShsm-authorized user, the recovery fails and DFSMShsm issues an error message if the data set is not allocated and no backup version can be found.

If the data set is cataloged, DFSMShsm invokes DFSMSdss, specifying REPLACE, and passes the management class name and storage class name. DFSMSdss attempts to restore the data set over the top of the existing data set. If the space is too small, DFSMSdss fails the request, and DFSMShsm writes the DFSMSdss error message in the dump activity log and issues an error message to the user.

If you specify NEWNAME and the NEWNAME data set is cataloged when the original data set is not cataloged, DFSMShsm first recovers the backup version of the original data set, using the storage class and management class appropriate to the NEWNAME data set. DFSMShsm then restores the dump copy of the data set over the top of the recovered data set. Finally, DFSMShsm deletes the cataloged NEWNAME data set and renames the restored data set to the new name.

Recovery of Direct Access Data Sets

The way to access data after a BDAM data set is recovered is an important consideration when recovering a data set to a volume with a different track length than the level 0 volume from which it was backed up. Using the DAOPTION parameter of the RECOVER command allows you to specify how the recovered data set is to be accessed (relative track or relative block). For instance, using the RELBLK option of DAOPTION allows a data set to be recovered to a device type with a smaller track size than the original level 0 volume. In this case, the data is moved to “fill out the track” rather than a “track to track” image as is the case for the RELTRK and SAMETRK options.

The DAOPTION parameter can only be used when recovering a data set as non-SMS-managed.The FORCENONSMS parameter can be used to recover a data set that would otherwise be returned as SMS managed.

Note: If the BDAM data set is to be recovered from a physical volume dump copy, the device types must be the same. Consequently, DAOPTION (RELBLK) cannot be specified.

Recovery of Extended Format Data Sets

DFSMShsm recovers or restores an extended format data set (as indicated in the MCC record), using DFSMSdss as the data mover, as follows:

- If FROMDUMP(DUMPVOLUME(*volser*)) is not specified, DFSMShsm always recovers an incremental backup version of a multiple-stripe extended format data set, even if a physical dump copy is more recent than the incremental backup version.
- If FROMDUMP(DUMPVOLUME(*volser*)) is not specified, DFSMShsm fails the recovery of a multiple-stripe extended format data set if the data set has only been backed up by a physical full-volume dump.
- An extended format data set that had been multiple-stripe when it was backed up or dumped is recovered or restored as a single stripe or multiple-stripe extended format data set, regardless of the existence of the target data set before the RECOVER command is issued. The number of stripes depends on the storage class SUSTAINED DATA RATE value, and the availability of the required volumes.
- An extended format data set that had been single stripe when it was backed up or dumped is recovered or restored as a single stripe or multiple-stripe extended format data set, regardless of the existence of the target data set before the RECOVER command is issued. The number of stripes depends on the storage class SUSTAINED DATA RATE value, and the availability of the required volumes.
- A data set that was extended format when backed up or dumped is recovered or restored as a nonextended format but still SMS-managed data set if no SMS volumes supporting extended format data sets are available.
- If the ACS routine is modified to direct an extended format data set to a non-SMS-managed volume, or if no SMS-managed DASD volumes attached to an ESCON® 3990 Model 3 are available, DFSMShsm fails the recover of the data set.
- A data set that was not extended format when it was backed up is always recovered or restored as a data set that is not extended format. DFSMShsm never converts a data set that is not extended format to an extended format data set through recover or restore processing, even if the data set is preallocated as an extended format data set before the RECOVER command is issued.
- If FORCENONSMS is specified on the RECOVER command to force an extended format data set to be recovered or restored to a non-SMS-managed volume, DFSMShsm recovers or restores the data set to a non-SMS-managed volume, but the data set becomes a data set that is not extended format.
- If FROMVOLUME is specified on the RECOVER command to recover an extended format data set, DFSMShsm fails the recover or restore.
- If reblocking is required for an extended format data set, DFSMShsm performs the reblocking.

To restore a single stripe extended format data set from a physical dump copy on a specific dump tape, you can issue:

RECOVER *dsname* **FROMDUMP(DUMPVOLUME(*volser*))**

To restore an existing multiple-stripe extended format data set from a physical dump tape, you must issue the RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) command for each stripe of the data set. If the extended format data set no longer exists, you must catalog the data set after the restore operations. DFSMShsm and DFSMSdss do not catalog a multiple-stripe extended format data set during physical data set restore processing.

Last-Backup Date after Recovery

After DFSMShsm recovers or restores a data set, the settings for the last-backup date are as follows:

- If a new name has been provided for the data set, the last-backup date is set to zero.
- If a new name has not been provided for a non-VSAM data set and the data set was recovered from a backup version, the last-backup date is set to the backup date of the version being recovered.
- If a new name has not been provided for a VSAM data set and the data set was recovered from a backup version, the last-backup date is the value set by IMPORT or DFSMSdss, whichever recovered the data set.
- If a new name has not been provided and the data set had been restored from a dump copy, the last-backup date is the value restored from the dump copy.

Reblocking during Recovery

The purpose of reblocking is to make the most efficient use of available space. If you use DFSMShsm as the data mover and specify the CONVERSION parameter of the SETSYS command, DFSMShsm can reblock some types of record organizations of physical sequential data sets during recovery.

If you use DFSMSdss as the data mover, specify CONVERSION(REBLOCKTOANY) to reblock physical sequential and partitioned data sets.

Resetting the Date-of-Scratch Field

During a recovery or restore, DFSMShsm checks the date-of-scratch field in the MCB record if NEWNAME is not specified. If a date-of-scratch exists, DFSMShsm removes the date.

Tape Mount Optimization during Recovery

The tape mount optimization for data set recoveries is accomplished by recovering all queued or soon arriving data set requests from a single and common backup tape with a single tape mount. For media before 3480s, those requests from a single and common backup tape are processed in ascending file-block ID. For 3480 (single-file-format tape cartridges) and later media, those requests from a single and common backup tape are processed in priority order on the recovery queue. The tape mount optimization is not applied to data set requests from dump tapes or when a data set spans backup tapes.

Processing Priority of Data Set Recoveries

Programming Interface Information

All data set recovery requests are queued to and processed from a queue of management work elements called the recovery queue (separate from the volume recovery queue). DFSMShsm gives all WAIT requests a higher priority on the queue than NOWAIT requests.

You can change the priorities of these requests as they are put on the queue by using the return-priority installation exit ARCRPEXT. ARCRPEXT allows you to assign a relative queuing priority such that WAIT requests can be prioritized relative to other WAIT requests, and NOWAIT requests prioritized relative to other NOWAIT requests. (In the case of a NOWAIT request initiated from a volume recover request, the queuing priority is initialized with the value propagated from the volume recover request passed earlier to the exit.) Refer to *z/OS DFSMS*

Installation Exits for a description of ARCRPEXT.

End of Programming Interface Information

Restoring a Data Set from a DFSMShsm Dump Copy

A DFSMShsm-authorized user can specifically request that a data set be restored from a dump copy rather than recovered from a backup version. The following commands specifically request that a data set be restored from a dump copy:

- RECOVER *dsname* FROMDUMP

  This form of the command restores the data set from the most recent dump copy. If a data set is cataloged with the same name as the data set being restored, the restore fails.
- RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*))

  This form of the command restores the data set from the most recent dump copy made to the specified dump class. If a data set is cataloged with the same name as the data set being restored, the restore fails.
- RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*))

  This form of the command restores the data set from a specific dump volume. The volume may or may not be the most recent dump copy. If a data set is cataloged with the same name as the data set being restored, the restore fails.
- RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) REPLACE

  This form of the command restores the data set from a specific dump volume. The restored data set replaces the cataloged data set having the same name.
- RECOVER *dsname* TOVOLUME(*volser*) UNIT(*unittype*) - FROMDUMP(DUMPGENERATION(*dgennum*))

  This form of the command restores the data set to the specified volume from a specified generation of a dump volume.
- RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - UNIT(unittype) TOVOLUME(*volser*)

  This form of the command restores the data set from a specific dump volume. This command must be used to restore a data set from a former device type when a new volume of a similar device type—but having the same volume serial number—resides on the system (for example, a 3380 volume is replaced with a 3390 volume, the same volume serial number is retained, and the 3390 volume is running in a 3380 emulation mode). The recovery is supported only when the target device type is similar to the source device type from which the dump copy was made.

The FROMDUMP parameter of the RECOVER command directs that the data set be restored from a dump copy. If the dump copy starts in file 2 or beyond on a stacked dump tape, DFSMShsm supplies the file block ID for high-speed positioning to the start of the dump copy.

You can also specify a DATE parameter to direct DFSMShsm to restore the data set from a dump made on or before the specified day.

Non-VSAM data sets and non-key-range VSAM data sets cataloged in an integrated catalog facility catalog that can be restored within the restrictions of DFSMSdss can be restored by DFSMShsm. Key-range VSAM data sets can be recovered from backup versions, but cannot be restored from dump volumes. The DFSMShsm default for a dump class does not allow physical data set restores from

dump volumes. You can allow DFSMShsm to perform restores by specifying the DATASETRESTORE subparameter of the DEFINE DUMPCLASS command.

During the data set restore, the dump volumes must be mounted in the same order that they were written during the full-volume dump. The dump volumes are read until the desired data set is found. For a dump copy stacked on a dump tape, with its own unique data set name, the search for the data set starts in the file containing the dump copy. If the dump volume has more than one valid dump copy, the desired data set is not catalogued or is migrated, and the data set has no backup copy, you must use the SOURCEVOLUME*volser* subparameter of the FROMDUMP parameter to tell DFSMShsm which dump copy to use.

To restore a non-SMS-managed, multi-volume, non-striped data set from dump tapes, you must issue the RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) command for each segment of the data set.

You can restore each segment to either non-SMS- or SMS-managed volumes. You then issue a DEFINE NONVSAM command so the segments are restored in the correct order. After all segments are restored, the entire data set can be cataloged using IDCAMS. If the data set is restored to non-SMS-managed volumes, it can optionally be copied or moved to SMS-managed volumes. The data set is not recataloged by the recover operation.

You can use the LIST command to ensure that a dump copy contains the data set to be restored. For example, you may need the list if you are not running the incremental backup function and the data set is no longer cataloged.

To request that a data set be *specifically restored* from a dump copy, the following conditions must be satisfied:

- The volume receiving the data set and the original volume from which the dump copy was made must be of similar device type.
- The volume receiving the data set and the data set being restored must be compatible in terms of being SMS-managed. (You can override this restriction by using the FORCENONSMS parameter to restore an SMS-managed data set to a non-SMS-managed volume, and the TOVOLUME parameter to direct recovery to a specific volume.)
- The FROMDUMP parameter must be specified.
- If the DUMPVOLUME subparameter is not specified, the data set must be currently cataloged or must have been incrementally backed up by DFSMShsm at some point in time. There must also be a dump copy of the original source volume in the specified class.

If the DUMPCLASS subparameter or the DUMPVOLUME subparameter is specified with the RECOVER command, DFSMShsm does not enforce the DATASETRESTORE subparameter or the permanent retention restrictions of the DEFINE command for the dump class that the dump copy is associated with. Dump volume availability is the responsibility of the authorized user in these cases; therefore, be careful not to request a data set restore from a dump volume that is not in the on-site tape library.

Rules: If neither the DUMPCLASS subparameter nor the DUMPVOLUME subparameter is specified with the FROMDUMP parameter, then the following conditions *must* be met or the command fails:

- The data set must be currently cataloged or must have been incrementally backed up by DFSMShsm at some time. (This condition is also necessary when DUMPCLASS is specified.)
- An eligible dump class must exist that allows individual data set restores. For example, the dump class must have the DATASETRESTORE attribute defined for the class.
- The data set to be restored must have resided on a volume dumped to an eligible dump copy, as described by "Searching for Eligible Dump Copies" on page 229.
- The dump volumes on which the dump copy resides must have a retention period other than NOLIMIT. DFSMShsm assumes the dump volumes are physically available to be mounted when the RECOVER command is issued.
- The data set must be listed in the dump VTOC copy data set associated with the dump generation in which the dump copy is contained. This condition must also be met even if DUMPCLASS is specified.

Restore of VSAM Data Sets

The restoration of a VSAM data set does not recatalog the data set. However, DFSMShsm can recatalog a VSAM data set if a backup version of the data set exists. To recatalog the restored data set, DFSMShsm first recovers the backup copy. The recovery of the backup copy recatalogs the data set. After recovering the backup version of the VSAM data set, DFSMShsm invokes DFSMSdss to restore the data set by replacing the version just recovered.

If a backup version of the data set does not exist or the attempt to recover the backup version fails, the VSAM data set that is restored to the target volume is uncataloged. A warning message is issued to alert you to your need to recatalog the data set. When a VSAM data set does not have a DFSMShsm backup version or has not been CATALOGED on the target volume, you must recatalog the data set by using the Access Method Services DEFINE command with the RECATALOG option. DFSMShsm does not recatalog a VSAM data set after a restore. When a restored VSAM data set replaces an existing data set or a recovered backup version, you should ensure that the catalog records reflect the desired conditions.

If DFSMShsm successfully recovered a backup version but the restore failed (for example, the allocated space or the control interval was too small), you should delete the recovered data set. After deleting the recovered data set, you should:

1. Predefine the data set on the target volume, using the proper allocation and control interval sizes
2. Reissue the RECOVER command to request the restore

If the VSAM data set exists on the target volume when the recovery request is made, neither DFSMShsm nor DFSMSdss alters any preexisting catalog information for the data set. You are responsible for ensuring that the catalog information is correct.

No catalog entries are created or modified for these cases. DFSMSdss does restore the VSAM volume data set catalog entry. The data set cannot be an integrated catalog facility catalog. DFSMShsm processing causes the catalog to be opened if it is not already opened. DFSMSdss does not restore an open integrated catalog facility catalog. If you need to restore the catalog from a dump copy, refer to "Restoring Data Sets or Volumes Directly with DFSMSdss" on page 427.

Rule: The original VSAM data set must have been backed up by DFSMShsm backup for the recover function to operate correctly.

Managing Volume Backups with 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 DFSMShsm-managed volumes. 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 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 small time frame. The term fast replication refers to the FlashCopy function supported by IBM TotalStorage Enterprise Storage Server (ESS) disk and the SnapShot function supported by IBM RAMAC Virtual Array (RVA) disk.

Using the Copy Pool Construct for Fast Replication Setup

The SMS copy pool construct contains the names of up to 256 storage groups that can be processed collectively for fast replication operations.

When you define a copy pool, the individual source pool storage group names are recorded. The actual volumes that are associated with each storage group are retrieved during subsequent processing. Consequently, the volumes that are processed during a function, such as backup, might not be the same volumes that are associated with the storage group at the time that the copy pool was established. **Example:** If one or more volumes are added to a storage group within a copy pool, those volumes are automatically processed as part of the next FRBACKUP command. You are not required to update the copy pool definition unless you want to add or remove storage groups from the pool.

Restriction: An individual storage group can be associated with more than one copy pool, but with no more than 50. Although you can include a storage group in multiple copy pools, IBM recommends against doing so when FlashCopy version 1 is used as the fast replication utility. Each individual copy pool with its associated background copies for all of the volumes in the common storage group must process completely before the next copy pool can be processed. This processing can take several hours and prevents the volumes in the noncommon storage groups in those copy pools from having a timely backup created. (FlashCopy version 2 and SnapShot do not have this restriction.)

You can maintain as many as 85 backup versions for each copy pool. Each maintained version requires a unique target volume for each source volume. If you specify five dedicated versions, there must be five target volumes that are available for each source volume that is associated with the copy pool. All target volumes must be available when you issue the FRBACKUP COPYPOOL(*cpname*) PREPARE command. If the PREPARE function is not performed, the target volumes need not be available until the actual backup version is created.

With ESS FlashCopy version 1, the source and target volumes must be in the same logical subsystem (LSS), which is limited to 256 total volumes. Thus, for three

unique versions, you could have up to 64 source volumes, leaving 192 volumes available as target volumes. The source volumes within a storage group can span logical and physical subsystems.

Related reading: For additional information about the copy pool construct, see the *z/OS DFSMSdfp Storage Administration Reference*.

Using the Copy Pool Backup Storage Group Type for Fast Replication Setup

The SMS copy pool backup storage group type contains the candidate target volumes for DFSMShsm fast replication requests. The volumes associated with the copy pool backup storage group are for DFSMShsm use. If you need to restrict DFSMShsm access to a volume in a copy pool backup storage group, vary the volume off-line to MVS.

The field that is titled ″pool storage group″ in the SMS storage group definition is provided for you to specify the name of the pool's associated copy pool backup storage group. Only pool storage groups can define associated copy pool backup storage groups. A pool storage group can specify its own copy pool backup storage group or specify a copy pool backup storage group that is shared by multiple pool storage groups.

For DFSMShsm to copy each source volume in a storage group, enough eligible target volumes must exist in the copy pool backup storage group to satisfy the needs of 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 exact size of the source volume.
- For FlashCopy:
  - Not be a primary or secondary volume in an XRC or PPRC volume pair.
  - For Version 1,
    - Reside in the same LSS as the source volume.
    - At the time of the backup, not be in a FlashCopy relationship.
- For SnapShot:
  - Reside in the same RVA/SVA as the source volume.

Related reading: For additional information about the copy pool backup storage group type and the pool storage group, see the *z/OS DFSMSdfp Storage Administration Reference*.

Using the Extend and Overflow Storage Group Types with Fast Replication

The SMS ″extend″ and ″overflow″ storage group types allow you to allocate data sets on volumes or to extend data sets onto volumes that are not a part of a main source storage group. If you have implemented the SMS extend or overflow storage groups in your environment, you must ensure that all extend and overflow storage groups are included in the storage group list for the appropriate copy pools.

DFSMS does not verify that the extend and overflow storage groups that are related to main source storage groups are included in the copy pool definition. The source volumes contained in extended and overflow storage groups also require

target volumes to be set aside for their use in the copy pool backup storage group. A backup of each volume will be made regardless of whether it contains any data or not.

Related reading: For additional information about the extend and overflow storage group types, see the *z/OS DFSMSdfp Storage Administration Reference*.

Preparing for the Fast Replication Backup

Because the task of establishing a valid DFSMShsm fast replication environment can be nontrivial, DFSMShsm provides a PREPARE keyword that you can use with the FRBACKUP command. The PREPARE keyword allows you to validate your fast replication environment for DFSMShsm. DFSMShsm preassigns the required number of target volumes to each source volume in each storage group in the specified copy pool.

When you specify the PREPARE keyword, DFSMShsm verifies that there is a sufficient number of eligible target volumes in the specified copy pool backup storage group. **Example:** If you specified in the copy pool definition that DFSMShsm should keep two backup versions, then DFSMShsm preassigns two target volumes to each source volume in the copy pool.

You should use the PREPARE keyword whenever there is a change in the fast replication environment. Changing any of the following definitions will change the fast replication environment:

- The volumes in the pool or copy pool backup storage group are changed or removed.
- The number of backup versions for DFSMShsm to maintain is changed.
- The storage groups defined to the copy pool have changed.

If the PREPARE function fails, make the necessary corrections and reissue the PREPARE function until it is successful.

Creating a Fast Replication Backup

The FRBACKUP command creates a fast replication backup version for each volume in every storage group defined within a copy pool. You can use the the following FRBACKUP command to initiate a fast replication backup for a copy pool:

```
FRBACKUP COPYPOOL(cpname) EXECUTE TOKEN(token)
```

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 FRBACKUP COPYPOOL command is successful only after DFSMSdss has successfully established a fast replication relationship for each source volume. If one of the volumes fails, then the entire backup version is failed with the following results:

- DFSMShsm continues to process volumes to determine if any remaining volumes will also encounter an error.

- For each volume in error, the target volume is disassociated from the source volume.
- 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.

DFSMShsm skips all volumes whose SMS status has not been defined to the system. DFSMShsm fails volumes for which it cannot obtain a UCB, such as a volume that is varied offline.

Other Fast Replication Considerations

The following additional topics are related to your use of the FRBACKUP COPYPOOL command:

- **Processing a FRBACKUP COPYPOOL command:** During the processing of a FRBACKUP COPYPOOL command, the new backup version is directed to the target volumes that contain the oldest backup version (once the requested number of versions has been reached). If a FRBACKUP COPYPOOL command fails, then there are only *n*-1 valid versions available, where *n* is the number of requested versions. When a failed version exists, the failed version is the target of the new copy until that copy is successful. The backup version is not incremented for subsequent backups until a successful version has been created.
- **When FlashCopy is the fast replication technique being used:** DFSMShsm returns control to the system after a FlashCopy relationship for each volume has been established. However, the background copies for those volumes last from minutes to hours longer, depending on criteria such as:
  - The number of background copies being processed.
  - The amount of other higher priority activity that exists in the storage subsystem.

  Restriction: When you are using FlashCopy version 1, any subsequent FRBACKUP COPYPOOL command that is issued against the same copy pool or against another copy pool with a common storage group before all the background copies are complete causes the command to fail. This restriction does not exist for FlashCopy version 2 because a source volume can be in multiple concurrent relationships.
- **Using the TSO command:** When you use commands that specify WAIT or NOWAIT, the results that are sent back to the user depend on whether the WAIT parameter is specified, and the success or failure of one or more volumes:
  - If WAIT is specified and all volumes process successfully, a return code of zero is returned.
  - If WAIT is specified and one or more volumes fail, a nonzero return code and associated messages are returned.
  - If WAIT is not specified, control is returned to the user after the command has been accepted and verified. All messages continue to be issued, but you are not notified when the command completes.
- **Using the TOKEN parameter:** The TOKEN parameter accepts a token of up to 40 bytes in length that is associated with that particular version of the copy pool. The token is used to reference a particular backup version. **Tip:** Use unique tokens for each version so that you can easily reference the correct version.

  The token is returned as part of the LIST and ARCXTRCT data for a copy pool.

Serialization for the Fast Replication Function

Attention: It is the responsibility of the caller of DFSMShsm to ensure that the source volumes are in a state such that DFSMShsm can process them for a

meaningful recovery. This includes protecting against data set level updates (actual changes to the data) and volume level changes (DADSM functions such as allocations, renames, extends and scratches). By default, DFSMSdss serializes on the VTOC of each volume as it is processed to prevent DADSM functions from occurring against that volume.

If the caller needs to prevent DADSM functions from being performed against the set of volumes in the copy pool during the backup, then it is the caller's responsibility to do so. This should be done by serializing on all of the VTOCs before invoking DFSMShsm. Serializing on the VTOC prevents DADSM functions and also prevents DFSMSdss functions such as DEFRAG, COPY and RESTORE from modifying the VTOCs. When you perform this serialization, use the NOVTOCENQ parameter with the FRBACKUP command to indicate that DFSMSdss should not serialize on the VTOC of each volume.

Parallelism for Fast Replication Backup

DFSMShsm processes FRBACKUP COPYPOOL commands serially. In order to maximize throughput, DFSMShsm creates the fast replication backup for the volumes in a copy pool in parallel. There are two levels of parallelism. The first level is that DFSMShsm invokes DFSMSdss with the PARALLEL option and up to 254 volume pairs. The second level of parallelism is the number of DFSMSdss invocations that DFSMShsm processes concurrently. DFSMShsm processes between 1 and 64 DFSMSdss invocations concurrently, the default being 15. The number of concurrent DFSMSdss invocations that are processed and the number of parallel volumes passed to DFSMSdss can be modified with the SETSYS MAXCOPYPOOLTASKS(FRBACKUP(*nn*) DSS(*nnn*)) command.

Recovering a Fast Replication Backup

The FRRECOV command is used to recover volumes that have a fast replication backup version. DFSMShsm can recover these volumes either individually or at the copy pool level. Use the FRRECOV command to recover the target volume back to the source volume.

You can use the following command to recover all of the volumes from a copy pool:

```
FRRECOV COPYPOOL(cpname)
```

You can use the following command to recover an individual volume:

```
FRRECOV TOVOLUME(volser)
```

There is no keyword to perform a recovery at the storage group level.

When you specify the COPYPOOL keyword, DFSMShsm recovers all of the volumes that are associated with that copy pool at the time of the backup. If one or more volumes fail, the FRRECOV command ends with a nonzero return code. You can reattempt to recover the failed volumes using the TOVOLUME keyword. Use the FROMCOPYPOOL(*cpname*) keyword if the volume that is being recovered is contained within a storage group that is defined within multiple copy pools.

If a version other than the most recent must be recovered, you can specify the GENERATION, VERSION, DATE, or TOKEN keyword. Select the appropriate generation or version from the available versions that are returned by the

DFSMShsm LIST command or ARCXTRCT macro. Use the TOKEN keyword to recover the unique version that corresponds to that token. If the token is not unique, the most recent version that corresponds to the token is recovered. If no version exists with the specified token, the recover operation fails. If DATE is specified and multiple versions exist for the specified date, the most recent version for that date is recovered.

Other Considerations for Fast Replication

The following additional topics are related to your use of the FRRECOV command:

- **Processing a FRRECOV command:** When FlashCopy is used as the fast replication utility, a recovery from a fast replication backup cannot be performed if any of the volumes are in an existing FlashCopy relationship. You should not attempt the recovery of a copy pool until all of the background copies have completed from an invocation of the FRBACKUP command. You can use the QUERY COPYPOOL command to determine if any of the source volumes for a copy pool are in a FlashCopy relationship. In addition, you can use the FRRECOV command with the VERIFY keyword to prevent the initiation of a recovery of a copy pool for which all of the background copies from the backup have not yet completed. When you specify VERIFY(Y), before initiating any volume recoveries, DFSMShsm first determines if any of the volumes are in an existing FlashCopy relationship. If one or more volumes are in an existing FlashCopy relationship, then the recover fails. If you need an immediate recovery, you have two options:

  1. Wait until all of the background copies have completed. Then the latest backup version can be used for the recovery.
  2. For FlashCopy version 1, withdraw the outstanding FlashCopy relationships using the FRBACKUP COPYPOOL(*cpname*) WITHDRAW command. Doing this invalidates the latest version of the backup and enables an immediate recovery from the prior version. This process only works for FlashCopy version 2 when the generation 1 background copies are complete.

  Warning: You should use the LIST COPYPOOL command or the ARCXTRCT macro to ensure that you do not invalidate the only valid fast replication backup version.

  When SnapShot is used as the fast replication utility, a recovery can be performed immediately after the DFSMShsm FRBACKUP command completes.
- **TSO Usage Considerations** — When you use commands that specify WAIT or NOWAIT, the results sent back to the user depend on whether the WAIT parameter is specified and the success or failure of one or more volumes:
  - If WAIT is specified and all volumes process successfully, a return code of zero is returned.
  - If WAIT is specified and one or more volumes fail, a nonzero return code and associated messages are returned.
  - If WAIT is not specified, control is returned to the system after the command has been accepted and verified. All messages continue to be issued, but you are not notified when the command completes.

Parallelism for Fast Replication Recovery

DFSMShsm processes the FRRECOV COPYPOOL commands serially. In order to maximize throughput, DFSMShsm recovers the fast replication source volumes from the target volumes in parallel. There are two levels of parallelism. The first level is that DFSMShsm invokes DFSMSdss with the PARALLEL option and up to 254 volume pairs. The second level of parallelism is the number of DFSMSdss invocations that DFSMShsm processes concurrently. DFSMShsm processes between 1 and 64 DFSMSdss invocations concurrently, the default being 15. The number of

concurrent DFSMSdss invocations that are processed and the number of parallel volumes passed to DFSMSdss can be modified with the SETSYS MAXCOPYPOOLTASKS(FRRECOV(*nn*) DSS(*nnn*)) command.

Withdrawing DFSMShsm Background Copies

A FlashCopy relationship between a source volume and a target volume that was established through a FRBACKUP command should NEVER be withdrawn outside of DFSMShsm's control. Doing so will invalidate the backup version, but DFSMShsm will still manage the version as if it was valid. This may result in a data integrity exposure.

When it is necessary to withdraw one or more relationships established by a FRBACKUP command, you can use the following command:

```
FRBACKUP COPYPOOL(cpname) WITHDRAW
```

This command withdraws all current FlashCopy relationships associated with the volumes in the most recent VALID version for the specified copy pool. That version of the backup is marked as invalid. If the most recent valid version is not generation zero and any relationships were withdrawn, then the withdrawn version is also deleted because it is no longer usable.

The withdraw can only be performed at the copy pool level. If any part of a copy pool backup version is invalidated, then the entire version is invalidated.

Warning: You should use the LIST COPYPOOL command or the ARCXTRCT macro to ensure that you do not withdraw the only valid fast replication backup version. Doing this prevents any recovery from being performed.

Holding and Releasing the Fast Replication Functions

You can use the HOLD FRBACKUP command to hold the FRBACKUP function. This command indicates to DFSMShsm that no new FRBACKUP commands are selected for processing. You can use the HOLD FRRECOV command to hold the FRRECOV function. This command indicates to DFSMShsm that no new FRRECOV commands are selected for processing.

When you issue the HOLD FRBACKUP or HOLD FRRECOV command, all WAIT requests fail and all NOWAIT requests remain on the respective queue. Requests that have already been selected continue with normal processing.

You can use the RELEASE FRBACKUP and the RELEASE FRRECOV commands to release the holds that were placed on the respective functions.

Deleting Unneeded DFSMShsm Fast Replication Backup Versions

Through normal processing, new backup versions will replace older backup versions. There are two instances in which a process other than the normal roll-off case is required to delete an unneeded copy version. The first instance is when the number of versions specified for a copy pool is decreased. The second instance is when a copy pool is renamed or is no longer needed.

When the number of versions for a copy pool is decreased (by modifying the SMS copy pool definition), the next time that the FRBACKUP COPYPOOL command is

issued for that copy pool, the unneeded versions are deleted. The target volumes that contained those versions are made available for other use.

When a copy pool is renamed or no longer needed, you must delete the backup versions that relate to that copy pool. You can use the following FRDELETE command to delete the backup versions:

```
FRDELETE COPYPOOL(cpname) VERSIONS(bvn)
```

When you specify the VERSIONS keyword, the named backup versions are deleted. If you specify the TOKEN keyword, the version with the specified token is deleted. If you specify the ALL keyword, all backup versions are deleted. Before a version is deleted, any outstanding FlashCopy relationships are withdrawn.

Obtaining DFSMShsm Fast Replication Backup Information

You can use the QUERY command, the LIST command, the REPORT command, and the ARCXTRCT macro to obtain information regarding the fast replication backup versions that DFSMShsm is managing.

Using the QUERY Command to Obtain Fast Replication Information

You can use the following parameters with the QUERY command to obtain information about the fast replication function:

- **COPYPOOL(*cpname*):** When you use the COPYPOOL(*cpname*) parameter, DFSMShsm lists the source volumes of the specified copy pool for which a FlashCopy relationship is currently established. If there are no outstanding relationships, DFSMShsm returns the ARC1821I message. For FlashCopy version 2, the source volume may be in more than one relationship. When using FlashCopy version 1, this command can be used to determine if another backup version can be created when using FlashCopy version 1, or if a recover can be performed.

  If SnapShot is the fast replication utility, then the QUERY COPYPOOL(*cpname*) command is of no value. If you invoke this command, DFSMShsm will indicate that there are no outstanding FlashCopy relationships.
- **ACTIVE:** When you use the ACTIVE parameter, DFSMShsm lists two messages that relate to fast replication. Message ARC1826I indicates if DFSMShsm is processing a FRBACKUP COPYPOOL or FRRECOV COPYPOOL command. Message ARC1822I describes an active FRBACKUP COPYPOOL or FRRECOV COPYPOOL command.
- **SETSYS:** When you use the SETSYS parameter, message ARC1823I indicates the current COPYPOOL settings for backup and recovery.
- **STATISTICS:** When you use the STATISTICS parameter, DFSMShsm lists the ARC1825I message to indicate the current day's fast replication statistics.
- **USER or REQUEST:** When you use either the USER or REQUEST parameters, DFSMShsm lists two messages that relate to fast replication. Message ARC0167I indicates if a COPYPOOL command is waiting to process. Message ARC1822I indicates active FRBACKUP COPYPOOL and FRRECOV COPYPOOL commands.

Using the LIST Command to Obtain Fast Replication Information

You can use the following parameters with the LIST command to obtain information about the fast replication function:

- **COPYPOOL:** When you use the COPYPOOL parameter, with no copy pool name, DFSMShsm returns version information for each copy pool that has at least one attempted backup. This information includes the copy pool name, version number, valid or invalid, VTOCENQ, token, backup date, and time.
- **COPYPOOL(*cpname*):** When you use the COPYPOOL(*cpname*) parameter, DFSMShsm returns detailed version information for each backup version for the specified copy pool. This information includes the copy pool name, version number, valid or invalid, VTOCENQ, token, backup date, and time. Source and target volume pairs are included in this list. (The date and time fields are zero for "Prepare" versions.)
- **COPYPOOLBACKUPSTORAGEGROUP(*cpbsgname*):** When you use the COPYPOOLBACKUPSTORAGEGROUP(*cpbsgname*) parameter, DFSMShsm returns information regarding which volumes in the specified copy pool backup storage group that DFSMShsm is using. This information includes the copy pool backup storage group name and for each volume in that storage group that was selected by DFSMShsm, the name of the source volume that the target volume is a target for.
- **PVOL:** When you use the PVOL parameter, with either the BCDS or BOTH subparameters, DFSMShsm returns information about which copy pools the volume is associated with. To determine which target volumes are associated with a particular source volume, you can use the LIST COPYPOOL(*cpname*) command.

Using the REPORT Command to Obtain Fast Replication Information

You can use the following parameters with the REPORT command to obtain information about the fast replication function:

- **DAILY:** When you use the DAILY parameter, DFSMShsm reports the following:
  - The total number of requested fast replication volume backups.
  - The total number of fast replication volume backups that failed.
  - The total number of requested fast replication volume recovers.
  - The total number of fast replication volume recovers that failed.
- **VOLUMES:** When you use the VOLUMES parameter, DFSMShsm reports the following:
  - The total number of times a fast replication backup was attempted for the volume.
  - The total number of failed backup attempts for that volume.
  - The total number of times a fast replication recover was attempted for the volume.
  - The total number of failed recover attempts for that volume.

Using the ARCXTRCT Macro to Obtain Fast Replication Information

You can use the ARCXTRCT macro to obtain information about the fast replication function. When you use the data type of COPYPOOL and the COPYPOOL keyword with it's associated subparameter, the ARCXTRCT macro returns information for each version of a specified copy pool. This information includes the version number, valid or invalid indicator, date, time, and token.

Dumping Fast Replication Target Volumes

One benefit to your use of the the fast replication function is that you can request a fast replication of a set of volumes, follow this function with a dump of each target

volume to produce a dump tape, and this dump tape can be restored directly back to the source volume. By doing this, you enable the dump function to be performed without a significant outage. DFSMShsm does not support creating a dump copy of a target volume by using either the autodump function or the BACKVOL DUMP command. You must use DFSMSdss or an equivalent product to create a dump copy of a target volume.

Preventing DFSMShsm Functions from Processing Target Volumes

The DFSMShsm automatic functions (Primary Space Management, Autobackup, Autodump and Interval Migration) only process pool type storage groups. These automatic functions will not process copy pool backup storage group types.

The BACKVOL STORAGEGROUP command only processes pool type storage groups. This BACKVOL command will not process copy pool backup storage groups.

The BACKVOL VOLUME and the MIGRATE VOLUME commands will not process any volumes that reside in a copy pool backup storage group.

Other Copy Pool Considerations

You might encounter the following situations as you use copy pools and fast replication methods.

Maintaining Catalogs with Fast Replication Backup

Because a fast replication backup is a physical copy of the data, you should take special care to ensure that the catalogs stay synchronized with the data. Catalogs should be copied with the data that they represent so that both can be recovered to the same point-in-time.

If it is necessary for you to recover an individual volume and not the corresponding catalog, you must be aware that the data on the volume will not be synchronized with the catalog and the following may occur:

- Data sets may exist on the volume but not in the catalog.
- The catalog may have entries for data sets on the volume that are not actually present.
- Data set information for existing data sets in the catalog may be inaccurate.

Increasing the Number of Volumes in a Copy Pool

When one or more volumes are added to a storage group within a copy pool, DFSMShsm includes those new volumes with any subsequent backup versions. If you must recover a version that does not contain the new volumes, and data has been placed onto those volumes, after the recovery, those volumes will contain residual data that is unusable and must be cleaned up.

Example: If you have a a storage group that has 2 volumes (volumes A and B) and you request a fast replication of those two volumes. Later on, you add a third volume (volume C) to the storage group. New data is then placed onto volume C. You discover that an error has occurred and that you need to recover the last backup version. During the recovery, volumes A and B are recovered to the point of the fast replication. Because volume C was not a part of the original fast replication backup, it is not processed during the recovery. Volume C may contain data that is no longer valid and you will need to review this data on the volume. DFSMShsm issues message ARC1810I for each volume that fits this scenario.

Decreasing the Number of Volumes in a Copy Pool

If you must remove a volume from a storage group that is defined to a copy pool, you must take special care to avoid data loss. Data loss may occur if you request a recovery of the copy pool from which the volume was defined. For this reason, during the recovery of a copy pool, DFSMShsm fails the recovery of any individual volumes that existed within the copy pool at the time of the backup but that are not a part of the copy pool at the time of recovery. You may still recover these volumes individually.

Example: If you have two copy pools (COPYPOOLA and COPYPOOLB) and each contains a single storage group, SGA and SGB respectively. A daily fast replication backup is made of each copy pool. If storage group (SGA) contains an empty volume (named VOLA) that is moved to storage group (SGB) for space needs, after the volume (VOLA) is moved, data for copy pool (COPYPOOLB) is allocated on it. The potential for data loss arises from the fact that there may be a fast replication backup version for copy pool (COPYPOOLA) that contains the volume (VOLA). This backup version represents the volume (VOLA) in its empty state. If that fast replication backup version for copy pool (COPYPOOLA) is recovered, then the recovery of copy pool (COPYPOOLA) would recover the volume (VOLA) to its empty state and overlay any new data that had been allocated on it since it was moved to storage group (SGB). To prevent this scenario from happening and others similar to it, during the recovery of a copy pool, DFSMShsm fails individual volumes that are no longer defined to the copy pool that is recovering.

You should perform the following steps when you need to recover a backup version of a copy pool that contains a volume that was removed from the copy pool:

1. Recover the copy pool. Each volume that was removed from the copy pool since the time the backup version was created is failed with the ARC1811E message.
2. For each failed volume, determine if the volume should be recovered as a part of the copy pool. If so, perform the remaining steps for those volumes that need to be recovered.
3. Determine if you need to preserve any of the existing data on the volume. If you need to preserve any of the data, you must move the data off of this volume and onto another volume. If you do not do this, the recovery of the volume will overwrite all existing data.
4. Issue the following volume recovery command to recover an individual volume:

   ```
   FRRECOV TOVOLUME(volser) FROMCOPYPOOL(cpname) VERSION(vernum)
   ```

   Note: FROMCOPYPOOL and VERSION are optional parameters
5. Determine if the volume should be put back into the copy pool and put it back, if necessary.

   Note: The volume should be returned to the copy pool if another backup version will be created after the recovery and that backup version should include the data that was recovered to the volume and any subsequent changes to that volume.

Decreasing the Number of Volumes in a Copy Pool Backup Storage Group

You must take special care when removing a volume from a copy pool backup storage group. Before doing so, you ***must*** ensure that the volume is not the target of a valid DFSMShsm backup version. You can verify this by examining the output from the DFSMShsm LIST COPYPOOLBACKUPSTORAGEGROUP command.

If a target volume needs to be removed from a valid source to target pairing, you should use the FRDELETE command to delete the DFSMShsm backup version that the volume is a part of before you remove the volume.

Defining a Storage Group to Multiple Copy Pools

Although you may include a storage group in multiple copy pools, IBM recommends against doing this when FlashCopy version 1 is used as your the fast replication utility. If the FRBACKUP COPYPOOL command is issued for a copy pool with a common storage group, then other copy pools with that same common storage group cannot be backed up until the background copies for all of the volumes in the common storage group have completed. Even though the first FRBACKUP COPYPOOL command logically completes in a manner of minutes, allowing the subsequent FRBACKUP commands to be issued, the subsequent FRBACKUP commands will fail until all of the background copies are complete, which may take several hours. This prevents the volumes in the noncommon storage groups in those copy pools from having a timely backup created.

FlashCopy version 2 and SnapShot do not have this restriction.

Recovery of a Copy Pool

When recovering a copy pool, it might be possible for a high number of volumes to fail the recovery process. (This is not expected, but is possible). If SnapShot is your fast replication utility, then you can resubmit another recovery of the entire copy pool after you have addressed the reason for the failures. If FlashCopy version 1 is your fast replication utility, then the FlashCopy relationships that were successfully established during the first recovery must be completed before the subsequent recovery can be started. If FlashCopy version 2 is your fast replication utility, the recovery can be immediately retried.

Using Multivolume Data Sets in a Fast Replication Environment

You must give special consideration to multivolume data sets in a fast replication environment. Because VTOC enqueues are obtained as each volume is processed, it is possible for a multivolume data set to be updated on one of its volumes while the other volume is being fast replicated. To avoid this from occurring, you should ensure that data set activity for multivolume data sets is quiesced during the fast replication process.

Volume Recovery and Restore

Volume recovery refers to the process of recovering a level 0 volume to its original condition as of a specified date. Volume recovery is usually necessary only after a complete volume has been damaged.

Three methods exist for DFSMShsm volume recovery:

- Full-volume restore with update from incremental backup versions
- Full-volume restore
- Volume recovery from incremental backup versions

Because full-volume restore with update is somewhat a combination of the other two methods, and therefore the most complete, it is discussed first.

When recovering volumes by using backup versions either for update or as the only source, DFSMShsm provides optimized tape usage. The optimization is accomplished by recovering all required data sets from each individual tape in priority order with a single tape mount.

When recovering volumes by using backup versions either for update (APPLYINCREMENTAL) or as the only source, DFSMShsm recovers all cataloged non-VSAM data sets—whether or not they are SMS-managed—and all cataloged VSAM non-SMS-managed data sets to the volume being recovered. DFSMShsm may recover VSAM SMS-managed data sets to other volumes in the same storage group as the volume being recovered.

The procedures for recovering an ML1 volume differ from those for recovering a level 0 volume. See "Recovering a Migration Volume" on page 507 for a discussion of the procedures for ML1 volumes.

Attention: DFSMShsm incremental recovery and volume restore with APPLYINCREMENTAL do not support a device type change on the volume being recovered. If the device type is changed after the volume is backed up and if the device type in the catalog for all data sets residing on the volume is not changed, all VSAM data sets will fail to be recovered. All non-VSAM data sets may appear to be recovered successfully, but the first allocation after the recovery will fail because of the inconsistent unit name and device type that were created by the recovery.

Specifying the Maximum Number of Data Set Recovery Tasks in Each Host

In recovering from incremental backup versions (as the only source or during APPLYINCREMENTAL processing after a volume restore), DFSMShsm can recover multiple data sets in parallel. The MAXDSRECOVERTASKS parameter of the SETSYS command controls the number of concurrent data set recovery tasks (to a maximum of 64) that will run. To decide how many such tasks DFSMShsm should run, consider how many tape units are available if you use tape backup volumes. (The actual number of tasks used in recovering a given volume is limited by the number of backup tape volumes required.) When DFSMShsm has scheduled the data set recovery tasks for one volume, you can then (if needed) request recovery of another volume, which can proceed in parallel.

Full-Volume Restore with Update

Attention: Most of the DASD that is installed today is permanently mounted (for example, the 3380 or 3390 magnetic tape device). In addition, DASD can be either online or offline. Therefore, when you are recovering a volume with full-volume restore and incremental backup versions, a possibility exists that one of your users could begin using a data set in the interval between the restore and the incremental recovery. This can be avoided by putting the SMS-managed volume in DISALL status. There are two considerations you need to be aware of before placing the volume in DISALL status:

- With DISALL status, some data sets may be recovered (as directed by your ACS routines) to different volumes in the storage group containing the volume being recovered.
- If you are recovering a catalog, or a volume belonging to a storage group using the guaranteed space attribute, the volume must be in enabled (ENABLE) status during the recovery process.

Ensure that the SMS-managed volume is in disabled (DISALL) status before issuing the RECOVER command, unless you are recovering a catalog, or a volume from a storage group using the guaranteed space attribute.

After the recovery process has completed, be sure to review the messages for failures or data sets that were not recovered. For example, if the volume you are recovering contains part of a multivolume data set, that partial data set is not recovered. After the recovery process has completed, the data set is listed as having not been recovered, giving you the opportunity to recover the entire data set, including the parts of the data set that reside on volumes not being recovered. You can issue individual RECOVER *dsn* commands for data sets such as these.

You can use the following commands to combine full-volume restore with full-volume recovery.

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPVOLUME(*tape_volser*) APPLYINCREMENTAL) -
TARGETVOLUME(*spare_volser*)

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPCLASS(*class*) APPLYINCREMENTAL) -
TARGETVOLUME(*spare_volser*)

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPGENERATION(*dgennum*) APPLYINCREMENTAL) -
TARGETVOLUME(*spare_volser*)

When APPLYINCREMENTAL is specified on a RECOVER FROMDUMP command, DFSMShsm first performs a full-volume restore as described in "Full-Volume Restore from a Dump Copy" on page 254. After the volume restore has been completed, DFSMShsm performs the following steps:

1. If integrated catalog facility catalogs reside on the volume, DFSMShsm recovers any catalog for which a backup version exists that was created following the dump. All such catalogs must be successfully recovered for the APPLYINCREMENTAL process to continue. Recovery of the catalog will fail and the APPLYINCREMENTAL process will not continue for either or both of the following conditions:

- Any catalog was backed up by DFSMShsm operating with a version of MVS/DFP™ earlier than version 2 release 3.0
- Any catalog was dumped before being exported or backed up by DFSMShsm, which uses export

2. 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.
3. DFSMShsm applies incremental data set backups from the list of candidate data sets with the following exceptions:
   - Catalogs.
   - Data sets 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 being processed, and the data set has been restored to the volume, DFSMShsm scratches it from the restored volume.
   - VSAM data sets currently cataloged as MIGRAT. Cluster, index, and data components are scratched from the restored volume.
   - Non-VSAM data sets 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 (VVDS).
   - 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 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 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 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 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 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 being recovered to SMS-managed volumes.

Notes:

1. If the data set was not on the volume when it was dumped, but was on the volume later in the volume backup (backup VTOC is the latest), then the latest backup version is applied regardless of the version date.
2. When you are running full-volume restore with update, there is one instance when DFSMShsm allows a data set backup version (made while the data set was uncataloged) to be recovered for a data set that is currently cataloged. This occurs when the volume being recovered is SMS-managed, and the backup version of a data set was made from this volume when the data set was uncataloged—the volume was converted to SMS-managed—and the data set is currently cataloged on the volume being recovered.

   If a data set is cataloged as a multiple-volume or multiple-striped extended format data set, incremental recovery of the data set fails.
3. If a dump VTOC copy data set does not exist for the volume being restored, some of the cleanup described in the preceding steps may not be performed. The extent of the cleanup activity depends on the candidate data set list that is created from the current backup and dump VTOC copies. If you do not have a dump VTOC copy that corresponds to the dump copy being restored, you should verify that the APPLYINCREMENTAL parameter causes the volume and the catalogs to be synchronized.

   It is not necessary for the volume to have been processed by the DFSMShsm volume backup function if a dump VTOC copy data set exists for the dump generation from which the volume is restored. The VTOC copy data sets are used only to build a candidate list. Rely on the current contents of the catalogs and BCDS to verify any action to be taken during the apply incremental process.
4. During volume restore with update, all non-VSAM SMS data sets are recovered to the volume being restored. VSAM SMS data sets may be recovered to other volumes that are in the same storage group as the volume being restored.

The APPLYINCREMENTAL subparameter has some restrictions: in systems running DFSMSdfp, DFSMShsm can recover catalogs with the APPLYINCREMENTAL process. To be recoverable, these catalogs must have been backed up with MVS/DFP Version 2 Release 3.0, or later. To recover these catalogs, do the following:

- Start DFSMShsm under DFSMSdfp™
- Use DFSMShsm to back up all catalogs
- Back up new catalogs before dumping

When the preceding steps are taken, DFSMShsm can recover the integrated catalog facility catalogs during the APPLYINCREMENTAL process. If any of these steps are omitted, the APPLYINCREMENTAL process stops. For this condition, you may need to update aliases manually or with the recommended integrated catalog Forward Recovery Utility (ICFRU), and restart the incremental recovery process.

If you plan to use ICFRU regularly, you should run it before starting the incremental recovery process. Use the following procedure:

1. Use the DFSMShsm RECOVER command to recover the catalogs that resided on the volume being restored.

2. Run the ICFRU on each catalog. See the instructions for the ICFRU for details.
3. Use the DFSMShsm BACKDS command to back up the updated catalogs. This updates the backup versions used during the incremental recovery process.
4. Issue the RECOVER FROMDUMP command with the APPLYINCREMENTAL parameter.

If you do not plan to use ICFRU on the system, you must audit the volumes after the recovery process to either delete data sets or delete catalog entries in order to clean up any discrepancies found. However, if ICFRU is used before the incremental recovery process, no discrepancies occur in the catalogs.

Notes:

1. If the data set has been migrated, scratched, or moved since the latest backup version was made, it must be scratched after the recovery process has completed. Because the data set has migrated, was scratched, or was moved, it is not the most current copy, and the catalog has no way to point to it.
2. If the data set was recalled since the backup of the catalog, the catalog entry must be removed by using the DELETE NOSCRATCH command. You can attempt to recover the data set from the latest backup version.
3. If a data set was created since the backup of the catalog, the catalog entry must be removed by using the DELETE NOSCRATCH command.

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 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 2 or beyond on a stacked dump tape, DFSMShsm supplies the file block ID for high-speed positioning to the start of the dump copy. The following commands can be used to request a full-volume restore:

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP TARGETVOLUME(*spare_volser*)

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP TARGETVOLUME(*spare_volser*) DATE(*date*)

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPVOLUME(*tape_volser*))TARGETVOLUME(*spare_volser*)

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPCLASS(*class*)) TARGETVOLUME(*spare_volser*)

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPCLASS(*class*)) TARGETVOLUME(*spare_volser*) -
DATE(*date*) -

RECOVER * TOVOLUME(*original_volser*) UNIT(*unittype*) -
FROMDUMP(DUMPGENERATION(*dgennum*)) -
TARGETVOLUME(*spare_volser*)

Notes:

1. The TARGETVOLUME(*spare_volser*) parameter can be used to supply the restore with a volume that has a volume serial number different from the one

that is being restored. This saves you the step of changing the volume serial number (clipping a volume) to the lost volser.

2. A DFSMSdss restriction is that the original source volume of the full-volume dump must be of the same or similar device type (for example, 3390) as the target volume of the full-volume restore.

The first form of the command restores the volume from the most recent dump copy. However, DFSMShsm must search its records to determine the most recent dump copy for the particular volume. You should not use this form of the command because of the extensive searching required.

The second form of the command narrows the search for the correct dump copy. If you know the date that the volume was dumped to the copy that you want to restore from (either the most recent or some other copy), you can specify the date. Then DFSMShsm searches only for dump copies made on or before the date you have specified.

Similarly, the other forms of the command allow you to narrow the search by specifying:

- The volumes to which the source volume was dumped
- The dump class from which the dump volumes are to be taken
- The dump generation from which the dump volumes are to be taken

When a full-volume restore is requested, DFSMShsm verifies with the tape operator that the dump volumes are available for mounting. If any dump volume is unavailable, the restore fails. Dump volumes can be stored at a location other than the data processing location.

If the volume being restored is an SMS-managed volume and the dump copy being used to restore the volume was made when the volume was *not* SMS-managed, DFSMShsm fails the command and issues an error message. Likewise, if an attempt is made to restore a non-SMS-managed volume from a dump copy of the volume when it was SMS-managed, DFSMShsm fails the command and issues an error message.

The dump volumes on which the dump copy resides must have a retention period other than NOLIMIT unless the DUMPCLASS or DUMPVOLUME parameter is specified. DFSMShsm assumes the dump volumes are physically available to be mounted when the RECOVER command is issued. DFSMShsm considers any volume with a retention period of NOLIMIT to be physically not available.

To restore an SMS-managed volume, you must have DFSMSdfp installed at the correct level to support SMS, and SMS must be active.

DFSMSdss restores a volume only if the volume is not allocated to another job. If the volume contains a catalog, the catalog address space in a DFSMSdfp environment may have the catalog (and the volume) allocated. To free the volume before starting the restore, do the following:

1. Issue the F CATALOG,OPEN(*volser*) command to determine which catalogs on the target volume are allocated.
2. For each catalog that is open, issue the F CATALOG,UNALLOCATE (*catalogname*) to cause the catalog address space to close the catalog and deallocate it. When the last catalog on the volume is deallocated, the target volume is also deallocated by the catalog address space.

3. Issue a D U,,,addr,1 command to ensure that the volume is not allocated by another user.
4. If the volume is not now marked as allocated, issue the RECOVER command to DFSMShsm.

Related reading: For additional information about a DFSMSdss full-volume restore, see the *z/OS DFSMSdss Storage Administration Reference*.

Volume Recovery from Incremental Backup Versions

Normally volume recovery is run when an entire DASD pack has been lost or damaged or a significant amount of the data has become inaccessible. A copy of the VTOC 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 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:

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, then create the queue of data sets to be recovered, then release the tape data set recovery processing. Since data sets from different volumes may 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 thus 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 just to recover one level 0 volume. As a result, volume recovery can be very time-consuming. Consider using the DFSMShsm full-volume dump restore facility (see "Full-Volume Restore with Update" on page 251) to reduce the time 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, using the *next latest* copy of the VTOC copy data set. If a similar error occurs again, the volume recovery fails.

When DFSMShsm recovers a volume, it recovers the following:

1. Any base entries for a generation data group cataloged in the catalog residing on the volume being recovered.

2. Any integrated catalog facility user catalogs that do not already exist on the volume being recovered. For information about recovering an integrated catalog facility catalog, see "Backing Up and Recovering a Catalog" on page 440.
3. All other data sets except the following:
   - Catalogs
   - VSAM data sets currently cataloged as MIGRAT
   - ICF VSAM components
   - Data sets that are cataloged as multiple-volume
   - Non-VSAM data sets that are currently cataloged as MIGRAT but the selected backup version was made when the data set was cataloged
   - Non-VSAM data sets that are currently cataloged as MIGRAT, the selected backup version was made when the data set was uncataloged, but the migration took place after the backup version was made
   - Data sets that are cataloged on a volume other than the one being recovered when the backup version of a cataloged data set is selected for recovery
   - Data sets that are currently cataloged on the volume being recovered but the backup version selected for recovery was uncataloged when it was backed up
   - Data sets that were cataloged when backed up and but not currently cataloged
   - Data sets whose creation date is later than the backup date
   - Data sets that are not SMS-managed being recovered to a SMS-managed volume

DFSMShsm considers data sets uncataloged at the time they are backed up to be unique to the volume on which they reside. This is because multiple uncataloged data sets with the same name may exist on different volumes. Backup versions of uncataloged data sets will only be recovered to the volume from which they were backed up.

DFSMShsm considers a cataloged data set to be the same data set regardless of the volume on which it is currently cataloged. Therefore, DFSMShsm will recover the most current backup version of a cataloged data set regardless of the volume on which it was cataloged when it was backed up.

Table 16 shows the process of backup version data set selection for volume recovery.

Table 16. Backup Version Data Set Selection for Recovery

First Criterion	Second Criterion	Result
If only backup versions of cataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then the cataloged backup copy is recovered.
	and if the data set is not currently cataloged,	then recovery of this data set fails.
	and if the data set is cataloged but not on the target volume,	then recovery is not attempted.

Table 16. Backup Version Data Set Selection for Recovery (continued)

First Criterion	Second Criterion	Result
If only backup versions of uncataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then recovery fails.
	and if the data set is not currently cataloged,	then the uncataloged backup version, made from this volume, is recovered.
	and if the data set is cataloged but not on the target volume,	then the uncataloged backup version, made from this volume, is recovered.
If backup versions of cataloged and uncataloged data sets exist,	and if both backup versions are from the target volume,	then DFSMShsm recovers the most recent backup copy. Recovery continues, depending on whether or not the data set was cataloged, as previously discussed.
	and if the backup version of a cataloged data set was made from a different volume, and the data set is currently cataloged on a volume other than the target volume,	then DFSMShsm attempts to recover the uncataloged backup version.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is currently cataloged on the target volume,	then DFSMShsm attempts to recover the backup version of the cataloged data set.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is not currently cataloged,	then DFSMShsm attempts to recover the uncataloged backup version.

For SMS-managed volumes, SMS must be active to recover the data sets that were backed up from that volume.

All non-VSAM SMS data sets are recovered to the volume being recovered. VSAM SMS data sets may be recovered to other volumes that are in the same storage group as the volume being recovered.

If DFSMShsm encounters a cataloged multiple-volume VSAM data set, it issues a message stating that the data set is multiple volume but does not recover the data set. If the multiple-volume data set is not cataloged or is presently cataloged as a single-volume data set, DFSMShsm recovers the data set if an eligible backup version exists.

Processing Priority of Volume Recoveries

Programming Interface Information

All volume recover requests are queued to and processed from a queue of

management work elements called the volume recovery queue (separate from the recovery queue used for data sets). DFSMShsm gives all WAIT requests a higher priority on the queue than NOWAIT requests.

You can change the priorities of these requests as they are put on the queue by using the return-priority installation exit ARCRPEXT. ARCRPEXT allows you to assign a relative queuing priority to WAIT requests so they can be prioritized relative to other WAIT requests, and NOWAIT requests prioritized relative to other NOWAIT requests. Refer to *z/OS DFSMS Installation Exits* for a description of ARCRPEXT.

End of Programming Interface Information

If you are using MVS concatenated PARMLIB support, DFSMShsm uses the PARMLIB data set containing the ARCCMDxx member and the PARMLIB data set containing the ARCSTR*xx* member indicated in the startup procedure.

Chapter 7. Availability Management of Non-SMS-Managed Storage

DFSMShsm manages non-SMS-managed storage on a system basis. That is, all the data sets on volumes are managed by a system specification, whether for number of days between backups or number of retained versions. 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.

Unlike space management, availability management of non-SMS-managed storage has no major differences in the processing techniques from those that are used for management of SMS-managed storage. However, there are some additional subtasks that must be accounted for when dealing with the availability management of non-SMS-managed storage.

References to the example system in this chapter reflect input needed to support the subtasks for non-SMS-managed storage.

Subtasks for Availability Management of Non-SMS-Managed Storage

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 be backed up
- Defining how many versions to keep for each backed up data set
- Defining which volumes are backed up automatically
- Defining which volumes are dumped automatically

Specifying How Frequently Data Sets Are Backed Up

For SMS-managed data sets, you can specify a different backup frequency for each management class. This lets you have as many different backup frequencies as you have management classes. For non-SMS-managed storage, you can specify only one backup frequency for all volumes processed by any one DFSMShsm host. You can, however, change the backup frequency of individual non-SMS-managed data sets with the (H)ALTERDS command. See "Changing the Backup Conditions for a Data Set" on page 267 for more information. Both the backup frequency specified for a DFSMShsm host and the backup frequency specified for a management class indicate the number of days that must pass between backups of any data set.

The FREQUENCY parameter of the SETSYS command controls the backup frequency for the data sets processed by each DFSMShsm host. For the example, the command that is added to the ARCCMD*xx* member for each host is:

```
SETSYS FREQUENCY(1)
```

© Copyright IBM Corp. 1984, 2004

Specifying 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.

Note: You can use the (H)ALTERDS command to specify a different value from the DFSMShsm host-wide default for individual data sets.

As with SMS-managed storage, depending on the record length that is used to define the 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 be retained. For the example system, the following command is added to the ARCCMD*xx* member for each DFSMShsm host:

```
SETSYS VERSIONS(5)
```

Related reading: For additional information about how record length affects the number of backups, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Specifying Which Volumes Are Backed Up Automatically

For non-SMS-managed storage, you can identify each individual volume that DFSMShsm backs up automatically. At the same time that you make the volume known to DFSMShsm, you can specify the kind of unit to back the volume up to.

If multiple DFSMShsm hosts are requested to process automatic backup, only the first DFSMShsm host to process a given volume will perform the function. However, should you find it necessary to limit processing of certain volumes to one DFSMShsm host, you can do so by:

- Specifying the AUTOBACKUP parameter on the ADDVOL command in the DFSMShsm host you want to perform automatic backup on the volume
- Specifying the NOAUTOBACKUP parameter on the ADDVOL command in the DFSMShsm hosts you do not want to perform automatic backup on the volume

See Figure 3 on page 14 for the example system to which you will add the volumes. DFSMShsm host 3 will do the backup processing for the non-SMS-managed volumes, so the following commands are added to the ARCCMD*xx* member for DFSMShsm host 3:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOBACKUP)
```

You can add the following commands to the ARCCMD*xx* member in DFSMShsm hosts 1 and 2:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
```

For non-SMS-managed volumes, you can restrict the kind of unit that is used when backing up the volume. The BACKUPDEVICECATEGORY parameter of the ADDVOL command lets you specify whether to back the volume up to either DASD or tape (if you have identified both DASD and tape backup volumes to DFSMShsm with the ADDVOL command).

Specifying Which Volumes Are Dumped Automatically

As with volumes to be backed up, you must identify the non-SMS-managed volumes to be dumped automatically. With the same command, you specify the dump classes to which the volume is to be dumped. If multiple DFSMShsm hosts are requested to process automatic dump, only the first host to process a given volume performs the function. However, should you find it necessary to limit processing of certain volumes to one host, you can do so by:

- Specifying the AUTODUMP parameter on the ADDVOL command in the host you want to perform automatic dump on the volume
- Specifying the NOAUTODUMP parameter on the ADDVOL command in the hosts you do not want to perform automatic dump on the volume

See Figure 3 on page 14 for the example system to which the volumes are added. You can add the following commands to the ARCCMD*xx* member for DFSMShsm host 2:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL GP0002 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL GP0003 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL UG1001 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL UG1002 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL UG1003 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL UG2001 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL UG2002 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL UG2003 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,TWOWEEK))
ADDVOL VP0001 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,FOURWEEK))
ADDVOL VP0002 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,FOURWEEK))
ADDVOL VP0003 UNIT(3390) PRIMARY(AUTODUMP -
  (ONEWEEK,FOURWEEK))
ADDVOL ML1001 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -
  AUTODUMP(ONEWEEK,TWOWEEK))
ADDVOL ML1002 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -
  AUTODUMP(ONEWEEK,TWOWEEK))
ADDVOL ML1003 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -
  AUTODUMP(ONEWEEK,TWOWEEK))
```

You can add the following commands to the ARCCMD*xx* member for DFSMShsm hosts 1 and 3:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL ML1001 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 NOAUTODUMP)

ADDVOL ML1002 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 NOAUTODUMP)

ADDVOL ML1003 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 NOAUTODUMP)
```

All volumes except VP0001, VP0002, and VP0003 are dumped in accordance with the specifications for dump classes ONEWEEK and TWOWEEK. VP0001, VP0002, and VP0003 are dumped in accordance with the specifications for dump classes ONEWEEK and FOURWEEK. Notice that the ML1 volumes ML1001, ML1002, and ML1003 have been added to the list of volumes to be dumped. Although automatic backup cannot be performed for migration volumes, automatic dump can be performed for ML1 volumes.

Automatic Availability Management of Non-SMS-Managed Volumes

Processing for availability management of non-SMS-managed volumes is the same as that for SMS-managed volumes with the following exceptions:

- No request is sent to DFSMSdfp for the management classes and storage classes.
- 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.
- A non-SMS-managed volume can contain uncataloged data sets that can be backed up.

Backing Up Cataloged and Uncataloged Data Sets with the Same Name

Although DFSMShsm and DFSMSdss do not migrate uncataloged data sets, DFSMShsm can back them up, either automatically or by command. Therefore, DFSMShsm can back up several different uncataloged data sets that have the same data set name but reside on different volumes. Depending on the record length that is used to define the BCDS, DFSMShsm can maintain up to 29, or up to 100, backup versions for any specific data set name. After the number of backup versions reaches that limit, backup versions of cataloged data sets take precedence over backup versions of uncataloged data sets of the same name. When DFSMShsm backs up the data set again, DFSMShsm deletes the backup version of

an uncataloged data set first. In the case of data sets that are indirectly cataloged to SYSRES, those on the inactive SYSRES pack are considered uncataloged by backup processing, but are considered cataloged if backed up while running backup from their own system. For this reason, they cannot be protected by incremental backup. The results of a recover command are unpredictable due to the mixture of indirectly cataloged and uncataloged data sets with the same name.

To avoid potential confusion, do not give cataloged and uncataloged data sets the same data set name.

DFSMShsm keeps backup versions of a data set name in the following order:

1. All backup versions of a cataloged data set, arranged by date. The most recently created backup version is first. The number of backup versions DFSMShsm keeps is the limit of versions that existed the last time DFSMShsm backed up the data set.
2. All backup versions of all uncataloged data sets having the same name, arranged by date. The most recently created backup version is first. The number of backup versions DFSMShsm keeps is the limit of versions that existed the last time DFSMShsm backed up the data set.

Although the limit specified by the VERSIONS parameter of the SETSYS command applies to cataloged and uncataloged data sets separately, the maximum number of versions (based on the BCDS record size) applies to the sum of the backup versions of both cataloged and uncataloged data sets with the same name. When the sum of the current backup versions is greater than the BCDS-related limit, DFSMShsm creates a new version and deletes the oldest backup version of the uncataloged data sets. For example, assume the following:

- The maximum BCDS record size allows DFSMShsm to maintain 29 backup versions for any data set.
- A cataloged data set and several uncataloged data sets have the same data set name.
- The cataloged data set currently has 24 backup versions.
- The uncataloged data sets currently have four backup versions.
- You specified a limit of 24 with the VERSIONS parameter.
- DFSMShsm backs up one cataloged data set and two uncataloged data sets with the same name.

Because DFSMShsm (in this example) tracks only 29 backup versions, here are the results:

- Twenty-four backup versions of the cataloged data set.
- Five backup versions of the uncataloged data sets. DFSMShsm keeps the five most recently created backup versions of these data sets.

You can identify backup versions of an uncataloged data set by associating the data set name with the volume serial number of the source volume for the backup version. DFSMShsm considers this association of the data set name and the source volume serial number a unique identification.

Command Availability Management for Non-SMS-Managed Storage

The activities that were specified as being available by command for SMS-managed storage are available for non-SMS-managed storage. Differences for non-SMS-managed storage are:

- A command that allows changing the backup conditions for individual data sets
- The ability to back up uncataloged data sets
- The ability to back up all non-SMS-managed primary volumes
- The ability to delete backup versions of uncataloged data sets
- The ability to specify a frequency value for a volume backed up by command
- The ability to recover uncataloged data sets
- The ability to recover an SMS-managed data set to a non-SMS-managed volume
- Differences in the way that the EXPIREBV command selects backup versions for deletion

This section describes the differences between the commands for SMS-managed storage and for non-SMS-managed storage. The forms of the commands specified for SMS-managed storage also are used for non-SMS-managed storage. However, when they are used for non-SMS-managed storage, those forms of the commands use the frequency and versions attribute specified with the system SETSYS command.

Dumping a Volume

The commands for dumping a volume are the same for non-SMS-managed storage as for SMS-managed storage. However, the dump process for non-SMS-managed volumes does not request the storage group attributes from DFSMSdfp.

Changing the Backup Conditions for a Data Set

SMS-managed storage controls the frequency of backup and the number of versions retained for each data set through the management class attributes. For non-SMS-managed storage, frequency and versions are a DFSMShsm host-wide specification. However, you can control the frequency and versions for individual data sets with the (H)ALTERDS command. You can control the frequency and the versions independently of each other. The forms for the command are:

ALTERDS *dsname . . .* **FREQUENCY(*days*)**

ALTERDS *dsname . . .* **VERSIONS(*limit*)**

ALTERDS *dsname . . .* **FREQUENCY(*days*) VERSIONS(*limit*)**

HALTERDS *dsname . . .* **FREQUENCY(*days*)**

HALTERDS *dsname . . .* **VERSIONS(*limit*)**

HALTERDS *dsname . . .* **FREQUENCY(*days*) VERSIONS(*limit*)**

Non-DFSMShsm-authorized and DFSMShsm-authorized users can use the HALTERDS form of the command. Only DFSMShsm-authorized users can use the ALTERDS form of the command.

The command causes the frequency or versions attributes for the data sets that are identified by *dsname* to be changed to the values that are specified by the command parameters. The new values are then used for all volume backup

functions that are performed on those data sets until a new ALTERDS or HALTERDS command is issued for the data set.

Security Checking with Discrete Profiles

If your data sets are protected with discrete profiles, HALTERDS processing checks the profile of the original data set for authorization to change the backup conditions for the data set. If the original data set has been deleted, DFSMShsm checks any backed up discrete profile for the data set. If the second authorization check fails, the HALTERDS processing fails.

Backing Up an Individual Data Set

Backing up an individual cataloged data set is performed in the same way as for SMS-managed data sets. However, to back up individual *uncataloged* data sets, issue the following command:

BACKDS *dsname* **UNIT(***unittype***) VOLUME(***volser***)**

HBACKDS *dsname* **UNIT(***unittype***) VOLUME(***volser***)**

The HBACKDS form of the command can be used by either non-DFSMShsm-authorized or DFSMShsm-authorized users. The BACKDS form of the command can be used only by DFSMShsm-authorized users. The UNIT and VOLUME parameters are required because DFSMShsm cannot locate an uncataloged data set without being told where it is.

Inline Backup of an Individual Data Set

Backing up individual cataloged data sets is performed the same way as for SMS-managed data sets. (See "Backing Up an Individual Data Set" on page 208.) However, the backup of uncataloged data sets is not supported by inline backup.

Related reading: For additional information about how to use inline backup, see "Inline Backup" on page 415.

Backing Up All Primary Volumes

Use the following command to back up all non-SMS-managed primary volumes:

BACKVOL PRIMARY

This command causes DFSMShsm to back up *every* volume that has been identified to it as a primary volume by the ADDVOL command, not just those volumes with the AUTOBACKUP attribute.

Note: Volumes selected for processing by BACKVOL PRIMARY are not processed by automatic backup the same day and visa versa.

Backing Up All Changed Data Sets on Volumes

You can back up all eligible changed data sets on non-SMS-managed volumes, whether the volumes are managed by DFSMShsm or not. Issue the following additional commands to back up non-SMS-managed volumes:

BACKVOL VOLUMES(*volser,volser,...volser***)**

BACKVOL VOLUMES(*volser,volser,...volser***) FREQUENCY(***days***)**

BACKVOL VOLUMES(*volser,volser,...volser*) UNIT(*unittype*) -
FREQUENCY(*days*)

BACKVOL PRIMARY INCREMENTAL

BACKVOL PRIMARY INCREMENTAL FREQUENCY(*days*)

The first two forms of the command are used for backing up volumes that are managed by DFSMShsm. The first form backs up the changed data sets on the volume in accordance with the frequency for the system as specified by the SETSYS command. The next form uses the frequency specified on the command to replace the attribute specified for the system.

The third form of the command backs up a volume that is not managed by DFSMShsm.

The last two forms of the command use either the frequency attribute specified for the system or the one specified in the command to back up all the non-SMS-managed primary volumes in the system.

Deleting Expired Backup Versions of Non-SMS-Managed Data Sets

To delete expired backup versions from the backup volumes, use the EXPIREBV command as you do for deleting expired SMS-managed backup versions. Issue the following command to delete non-SMS-managed backup versions:

EXPIREBV NONSMSVERSIONS(DELETEIFBACKEDUP(*days*))
EXECUTE

EXPIREBV NONSMSVERSIONS(CATALOGEDDATA(*days*))
EXECUTE

EXPIREBV NONSMSVERSIONS(UNCATALOGEDDATA(*days*))
EXECUTE

Processing for Expired Backup Versions of Non-SMS-Managed Data Sets

The EXPIREBV command processing for non-SMS-managed data sets is identical to that for SMS-managed data sets up to the point of determining the expired backup versions. EXPIREBV processing for SMS-managed data sets is described in "Deleting Expired Backup Versions from the Backup Volumes" on page 216.

If the command includes (DELETEIFBACKEDUP(*days*)), DFSMShsm deletes any retired version older than the value specified by the *days* subparameter. It also deletes any additional older cataloged backup versions that exist.

If the command includes (CATALOGEDDATA(*days*)), DFSMShsm processes only data sets that were cataloged when they were backed up. For deleted data sets, if a date of scratch exists in the MCB record, DFSMShsm calculates the number of days since the date of scratch, and compares the result to the *days* specified in the command, or to the default of 60 days (if *days* was not specified). If the time since the date of scratch is more than the value specified or more than the default, DFSMShsm deletes all cataloged backup versions of the data set except for the retired version. For existing data sets, the *days* subparameter is ignored, and all *excess* backup versions (that result from the (H)ALTERDS command) are deleted.

When (CATALOGEDDATA(*days*)) is used, two executions are required to delete unwanted versions. The first execution determines that the data set was scratched and stores the date of the first EXPIREBV execution as the scratch date in the DFSMShsm control data set MCB record. A subsequent execution deletes the unwanted backup versions when the requested time has passed. The scratch date is generally not the date that the data set was actually scratched, but is the date EXPIREBV was next run after the data set was scratched.

Programming Interface Information

The scratch date is stored in the BCDS MCB record in the 4-byte field MCBSCRD (offset X'48', packed decimal). It can be displayed using the FIXCDS command.

End of Programming Interface Information

Note: DFSMShsm honors a scratch date stored by the DFSMShsm space maintenance and recovery utilities if the date is newer than the most recent backup date for the data set and if EXPIREBV has not already stored the scratch date itself.

If the command includes (UNCATALOGEDDATA(*days*)), DFSMShsm processes data sets that were uncataloged when they were backed up. DFSMShsm calculates the number of days since the uncataloged backup version was created and compares the result to the days specified in the command. There is no default value for *days* for uncataloged backup versions. If the time since the uncataloged backup version was created is more than the value specified, DFSMShsm deletes that version.

Deleting Backup Versions of Specific Data Sets

Because non-SMS-managed data sets can be either cataloged or uncataloged and because you can have both cataloged and uncataloged backup versions with the same data set name, DFSMShsm provides the ability to specify that you want to delete the backup versions of the uncataloged data set. Issue the following commands to delete backup versions of uncataloged data sets:

```
BDELETE (dsname . . .) -
  FROMVOLUME(volser)

BDELETE (dsname . . .) -
  VERSIONS(backup version number . . .) -
  FROMVOLUME(volser)

HBDELETE (dsname . . .) -
  FROMVOLUME(volser)

HBDELETE (dsname . . .) -
  VERSIONS(backup version number . . .) -
  FROMVOLUME(volser)
```

DFSMShsm deletes the uncataloged backup versions that were made from the data sets with the specified name on the specified volume.

The MCB record is not deleted when no existing backup versions exist if the data set is not SMS-managed and if the MCB record contains any ALTERDS information.

Recovering Specific Data Sets

DFSMShsm recovers specific data sets to non-SMS-managed volumes under the following conditions:

- When the FORCENONSMS parameter is specified on the RECOVER command
- When SMS is not active and the backup version or dump copy being recovered was made from a non-SMS-managed data set
- When SMS is active and ACS does not assign a storage class to the data set (indicating that the data set is not currently SMS-managed)
- When the FROMVOLUME parameter is specified on the RECOVER command
- When the catalog indicates that the data set is not SMS-managed

You can issue the following additional commands to recover a non-SMS-managed data set:

- RECOVER *dsname* FROMVOLUME(*volser*)

  This command recovers an uncataloged data set to the volume from which it was backed up.
- RECOVER *dsname* FROMVOLUME(*volser*) REPLACE

  This command recovers an uncataloged data set to the volume from which it was backed up, and any existing data set of the same name on the volume is replaced.
- RECOVER *dsname* TOVOLUME(*volser*) - UNIT(*unittype*)

  This command recovers a data set to the volume specified.
- RECOVER *dsname* TOVOLUME(*volser*) - UNIT(*unittype*) FORCENONSMS

  This command recovers what may have been an SMS-managed data set to the specified non-SMS-managed volume.
- HRECOVER *dsname* TOVOLUME(*volser*) - UNIT(*unittype*)

  This is the unauthorized-user form of the command. It recovers a data set to the volume specified.
- RECOVER *dsname* NEWNAME(*newdsname*) - TOVOLUME(*volser*) UNIT(*unittype*)

  This command recovers a data set to the volume specified and gives the data set a new name.
- RECOVER *dsname* NEWNAME(*newdsname*) - TOVOLUME(*volser*) UNIT(*unittype*) - FORCENONSMS

  This command recovers what may have been an SMS-managed data set to the specified non-SMS-managed volume and gives the data set a new name.
- HRECOVER *dsname* NEWNAME(*newdsname*) - TOVOLUME(*volser*) UNIT(*unittype*)

  This command is the unauthorized user form of the command. The data set is recovered to the volume specified and is given a new name.
- RECOVER *dsname* REPLACE

  This command recovers a cataloged data set to the volume where the data set currently resides and replaces the existing data set with the recovered data set.
- RECOVER *dsname* REPLACE FORCENONSMS

  This command recovers a cataloged data set that was backed up as an SMS-managed data set to the non-SMS-managed volume where the existing data set currently resides. The existing data set is replaced with the recovered data set.
- RECOVER *dsname* REPLACE TOVOLUME(*volser*) - UNIT(*unittype*)

This command recovers a data set to the specified volume and replaces any existing data set with the same name on the specified volume.

- RECOVER *dsname* REPLACE TOVOLUME(*volser*) - UNIT(*unittype*) FORCENONSMS

  This command recovers a data set that was backed up as an SMS-managed data set to the specified non-SMS-managed volume. Any existing data set with the same name on the specified volume is replaced.
- HRECOVER *dsname* REPLACE TOVOLUME(*volser*) - UNIT(*unittype*)

  This is the unauthorized-user form of the command. It recovers a data set to a specified volume and replaces any existing data set with the same name on the specified volume.
- RECOVER *dsname* FROMVOLUME(*volser*) - NEWNAME(*newdsname*)

  This command recovers an uncataloged data set to the volume from which it was backed up and gives the recovered data set a new name.
- RECOVER *dsname* FROMVOLUME(*volser*) - REPLACE TOVOLUME(*volser*) UNIT(*unittype*)

  This command recovers an uncataloged data set to the volume specified. The data set with the same name on the specified volume is replaced.
- RECOVER *dsname* NEWNAME(*newdsname*) - REPLACE TOVOLUME(*volser*) UNIT(*unittype*)

  This command recovers a data set to the specified volume. The data set is given a new name, and any existing data set on the specified volume that has the same name as the new name is replaced.
- RECOVER *dsname* NEWNAME(*newdsname*) - REPLACE TOVOLUME(*volser*) - UNIT(*unittype*) FORCENONSMS

  This recovers a data set that was backed up as an SMS-managed data set to the specified non-SMS-managed volume. The data set is given a new name, and any existing data set on the specified volume that has the same name as the new name is replaced.
- HRECOVER *dsname* NEWNAME(*newdsname*) - REPLACE TOVOLUME(*volser*) UNIT(*unittype*)

  This is the unauthorized-user form of the command. It recovers a data set to the specified volume. The data set is given a new name, and replaces any existing data set on the specified volume that has the same name as the new name.
- RECOVER *dsname* FROMVOLUME(*volser*) - NEWNAME(*newdsname*) TOVOLUME(*volser*) - UNIT(*unittype*)

  This recovers an uncataloged data set to a specified volume and gives it a new name.
- RECOVER *dsname* FROMVOLUME(*volser*) - NEWNAME(*newdsname*) TOVOLUME(*volser*) - UNIT(*unittype*) REPLACE

  This recovers an uncataloged data set to a specified volume. The data set is given a new name, and any existing data set that has the same name as the new name on the specified volume is replaced.

The processing for recovering non-SMS-managed data sets is the same as for SMS-managed data sets with the following exceptions:

- When FROMVOLUME is specified, it implies that the data set is not SMS-managed.
- Selection of the target volume is somewhat different:
  - If TOVOLUME is specified, the target volume is the volume specified by TOVOLUME.

- If TOVOLUME is not specified and FROMVOLUME is specified, the target volume is the volume specified by FROMVOLUME.
- If neither TOVOLUME nor FROMVOLUME is specified, the target volume is the volume on which the data set is cataloged, unless DFSMShsm backed up the data set from a migration volume. If DFSMShsm backed up the data set from a migration volume, DFSMShsm recovers the data set to the volume from which the data set migrated.
- If DUMPVOLUME is specified, the target volume is the source volume from which the dump was made.
- If the specified NEWNAME data set is not currently cataloged, if the device type and volume serial number of the volume from which the incremental backup version was made differ from the device type and volume serial number of the volume on which the original data set is currently cataloged, and if the volume on which the original data set is currently cataloged has a more recent physical dump copy, the target volume is the volume on which the original non-VSAM data set is currently cataloged.
- If none of the preceding can be applied, the target volume is the volume from which the eligible backup version was made.

To request that DFSMShsm recover backup versions of data sets that were uncataloged when DFSMShsm backed them up, specify the FROMVOLUME parameter of the RECOVER command. If you do not specify FROMVOLUME, DFSMShsm recovers a backup version of a cataloged data set with the same name even though the data set may no longer be cataloged.

When a generation data set is recovered, it is recovered as an ordinary non-GDS data set.

Notes:

1. DFSMShsm fails the recovery of a non-SMS generation data set that is not currently cataloged, and whose corresponding generation data group is no longer defined.
2. DFSMShsm fails the recovery of a non-SMS generation data set that is not currently cataloged, and whose corresponding generation data group is not found through generic catalog locate.

To request that an SMS-managed data set be recovered to a non-SMS-managed volume, specify the FORCENONSMS and TOVOLUME parameters.

Recovery of VSAM Data Sets

The FROMVOLUME and TOVOLUME parameters of the RECOVER command, used when recovering non-VSAM data sets, have different results when DFSMShsm recovers VSAM data sets:

- The FROMVOLUME parameter applies only to uncataloged data sets and, therefore, is not used for VSAM data sets.
- For VSAM data sets not cataloged, the TOVOLUME parameter can specify only a volume owned by the same catalog where the data set being recovered is cataloged or will be cataloged. When you do not specify the TOVOLUME parameter, DFSMShsm recovers the data set to the volume specified in the current catalog entry. If no catalog entry exists, DFSMShsm recovers the data set to the same volume from which DFSMShsm backed up the data set.

Data Set Restore from a DFSMShsm Dump Copy

The results of the commands, in order, are:

1. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*)

   This command recovers a data set from the most recent dump copy made in the specified dump class to the specified level 0 volume.

2. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) FORCENONSMS

   This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the most recent dump copy made in the specified dump class to the specified non-SMS-managed level 0 volume.

3. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*)

   This command recovers a data set from the specified dump volume to the specified level 0 volume.

4. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) FORCENONSMS

   This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified level 0 volume.

5. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) REPLACE

   This command recovers a data set from the most recent dump copy made in the specified dump class to the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

6. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE FORCENONSMS

   This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the most recent dump copy made to the specified dump class in the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

7. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) REPLACE

   This command recovers a data set from the specified dump volume to the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

8. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE FORCENONSMS

   This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

9. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*)

   This command recovers a data set from the most recent dump copy made in the specified dump class to the specified level 0 volume. The recovered data set is given a new name.

10. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*) FORCENONSMS

This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the most recent dump copy made in the specified dump class to the specified level 0 volume. The recovered data set is given a new name.

11. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*)

    This command recovers a data set from the specified dump volume to the specified level 0 volume. The recovered data set is given a new name.

12. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*) FORCENONSMS

    This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified non-SMS-managed level 0 volume. The data set is given a new name.

13. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*)

    This command recovers a data set from the most recent dump copy that was made in the specified dump class to the specified level 0 volume. The recovered data set is given a new name and replaces any existing data set with the same new name on the specified volume.

14. RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*) FORCENONSMS

    This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the most recent dump copy that was made in the specified dump class to the specified non-SMS-managed level 0 volume. The recovered data set is given a new name, and replaces any existing data set with the new name on the specified volume.

15. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*)

    This command recovers a data set from the specified dump volume to the specified level 0 volume. The recovered data set is given a new name and replaces any existing data set with the new name on the specified volume.

16. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*) FORCENONSMS

    This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified non-SMS-managed level 0 volume. The recovered data set is given a new name and replaces any existing data set with the new name on the specified volume.

When restoring a non-SMS-managed data set from a dump volume, you can specify a TOVOLUME parameter and a UNIT parameter that cannot be specified for SMS-managed data sets. The TOVOLUME parameter directs that the data set be restored to the specified volume. The UNIT parameter identifies the unit type for restoring the data set to a volume not managed by DFSMShsm.

If the dump volume specified has more than one valid dump copy stacked on it, you can specify the SOURCEVOLUME*(volser)* subparameter of the FROMDUMP parameter to tell DFSMShsm which dump copy to use.

Data Set Recovery of Retired Data Sets

You can restore a retired data set. If the retired data set has a backup version that is marked as a retired version and a physical data set restore is done for the data set without a new name, the retired-version indication is removed from the backup control data set (BCDS). This allows the data set to be processed again by the incremental backup function.

Combining Full-Volume Restore with Volume Recovery

When combining a full-volume restore with a volume-recovery, the following considerations apply to non-SMS-managed volumes:

DFSMShsm verifies that the targeted data set has not been referred to prior to starting the APPLYINCREMENTAL process and the time that the data set is slated for recovery. If the data set has been referred to, the recovery fails with a return code of 67. Failing the recovery allows you to take action on any application that might have inadvertently used the old data set.

Restrictions:

- DFSMShsm does not honor the APPLYINCREMENTAL subparameter if it determines that a catalog that resides on the target volume owns the target volume. This restriction avoids conflicts in the way DFSMShsm supports catalogs and data sets that are cataloged.
- DFSMShsm does not back up catalogs and therefore they cannot be recovered.

Volume Recovery from Incremental Backup Versions

In addition to the considerations discussed for volume recovery from backup versions for SMS-managed volumes, the following considerations apply for non-SMS-managed volumes.

If a catalog exists on the volume being recovered, you must recover the catalog before you use DFSMShsm for the volume recovery. DFSMShsm catalogs any data sets that were cataloged when DFSMShsm backed them up, but that are not cataloged when DFSMShsm recovers them. If a data set was uncataloged when DFSMShsm backed it up, DFSMShsm recovers the data set, but does not catalog it.

If a catalog existed on the volume the last time DFSMShsm backed up the volume, DFSMShsm recovers the base entry for each generation data group cataloged. Therefore, if you want DFSMShsm to recover the base entry for each generation data group cataloged, the catalog must exist.

DFSMShsm recovers the generation data groups that are on the volume being recovered. If the generation data group base was not on the same volume as the generation data sets, you must ensure that the base entry exists before the volume is recovered. You may have to recatalog valid generation data group data sets that are on other volumes.

DFSMShsm tries only one time to recover generation data group base entries. The recovery is handled the same way as with VTOC copy data sets. If DFSMShsm uses the next-latest VTOC copy data set to recover the volume because DFSMShsm cannot use the latest VTOC copy data set, DFSMShsm recovers the next-latest copy of the generation data group base entries.

DFSMShsm attempts to recover the volume by using the *latest* VTOC copy data set and the latest copy of the generation data group base entries. DFSMShsm attempts

to recover the volume by using the *next latest* copy of the VTOC and generation data group base entries if one of the following occurs:

- An error occurs in allocating or opening the latest VTOC copy data set.
- The VTOC copy data set resides on a tape backup volume and the operator indicates before starting the actual recovery that the required tape volumes will not be available.

If similar errors occur again, the volume recovery fails.

When DFSMShsm recovers a volume, it recovers the following:

- Any base entries for a cataloged generation data group residing on the volume being recovered.
- Any user catalogs that do not already exist on the volume being recovered. For information about recovering a catalog, see "Backing Up and Recovering a Catalog" on page 440.
- Unmovable data sets.
- All other data sets.

DFSMShsm considers non-SMS uncataloged data sets to be unique by reason of the volume on which they reside. This is because multiple uncataloged data sets with the same name may exist on different volumes. Backup versions of uncataloged data sets are only recovered to the volume from which they were backed up.

DFSMShsm considers a cataloged data set to be the same data set regardless of the volume on which it is currently cataloged. Therefore, DFSMShsm recovers the most current backup version of a cataloged data set regardless of the volume on which it was cataloged when it was backed up.

Table 17 shows the process of backup version data set selection for volume recovery.

Table 17. Backup Version Data Set Selection for Recovery

First Criterion	Second Criterion	Result
If only backup versions of cataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then the cataloged backup copy is recovered.
	and if the data set is not currently cataloged,	then recovery of this data set fails.
	and if the data set is cataloged but not on the target volume,	then recovery is not attempted, and for APPLYINCREMENTAL, the data set is scratched from the volume.
If only backup versions of uncataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then recovery fails.
	and if the data set is not currently cataloged,	then the uncataloged backup version, made from this volume, is recovered.
	and if the data set is cataloged but not on the target volume,	then the uncataloged backup version, made from this volume, is recovered.

Table 17. Backup Version Data Set Selection for Recovery (continued)

First Criterion	Second Criterion	Result
If backup versions of cataloged and uncataloged data sets exist,	and if both backup versions are from the target volume,	then DFSMShsm recovers the most recent backup copy. Recovery continues, depending on whether or not the data set was cataloged, as previously discussed.
	and if the backup version of a cataloged data set was made from a different volume, and the data set is currently cataloged on a volume other than the target volume,	then DFSMShsm attempts to recover the uncataloged backup version.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is currently cataloged on the target volume,	then DFSMShsm attempts to recover the backup version of the cataloged data set.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is not currently cataloged,	then DFSMShsm attempts to recover the uncataloged backup version.

If a data set is a candidate for recovery, backup versions of the cataloged data set exist, and the data set is currently uncataloged, then DFSMShsm does not recover the data set because it appears to DFSMShsm that the data set was deleted after the backup version was created. For recovery of uncataloged data sets, a backup version of each data set is recovered to reflect the VTOC copy at the time of backup. It is up to you to delete any recovered uncataloged data sets that are unwanted.

If a data set is currently cataloged as MIGRAT, and a backup version of the cataloged data set was selected for recovery, then recovery is not performed because the data set migrated since backup.

If a data set was migrated from the target volume after the selected backup version of the uncataloged data set was created, then the recovery is not attempted because the uncataloged data set was deleted before the creation of the cataloged data set that was migrated.

If the volume you are recovering contains both a catalog and unmovable data sets, you could have a problem. After DFSMShsm recovers the catalog, DFSMShsm tries to recover the unmovable data sets. However, the allocation for the unmovable data sets can fail because the catalog could be allocated to the space where the unmovable data set must go. To solve this problem, perform the following steps:

1. After the volume recovery completes without recovering an unmovable data set, move the catalog to another volume.
2. Use the RECOVER command for the specific data sets that previously failed recovery.
3. Move the catalog back to the original volume.

After DFSMShsm has recovered a volume, you can use the VOLUME parameter of the AUDIT command to verify the status of data sets on the volume.

Chapter 8. Aggregate Backup and Recovery Support (ABARS)

Aggregate backup is the command-driven function that backs up a user-defined group (called an aggregate group) of data sets for recovery at another computer site or at the same site. *Aggregate recovery* is the command-driven function that recovers the data sets that were previously backed up by aggregate backup.

Aggregate backup and recovery support can be used to back up and recover both SMS and non-SMS-managed data.

Figure 32 on page 282 shows data movement during aggregate backup and recovery and how these functions interact with other products. The DFSMShsm aggregate backup and recovery support function is also referred to as ABARS.

Note: Only a host designated with the HOSTMODE=MAIN attribute can execute aggregate backup or aggregate recovery.

© Copyright IBM Corp. 1984, 2004

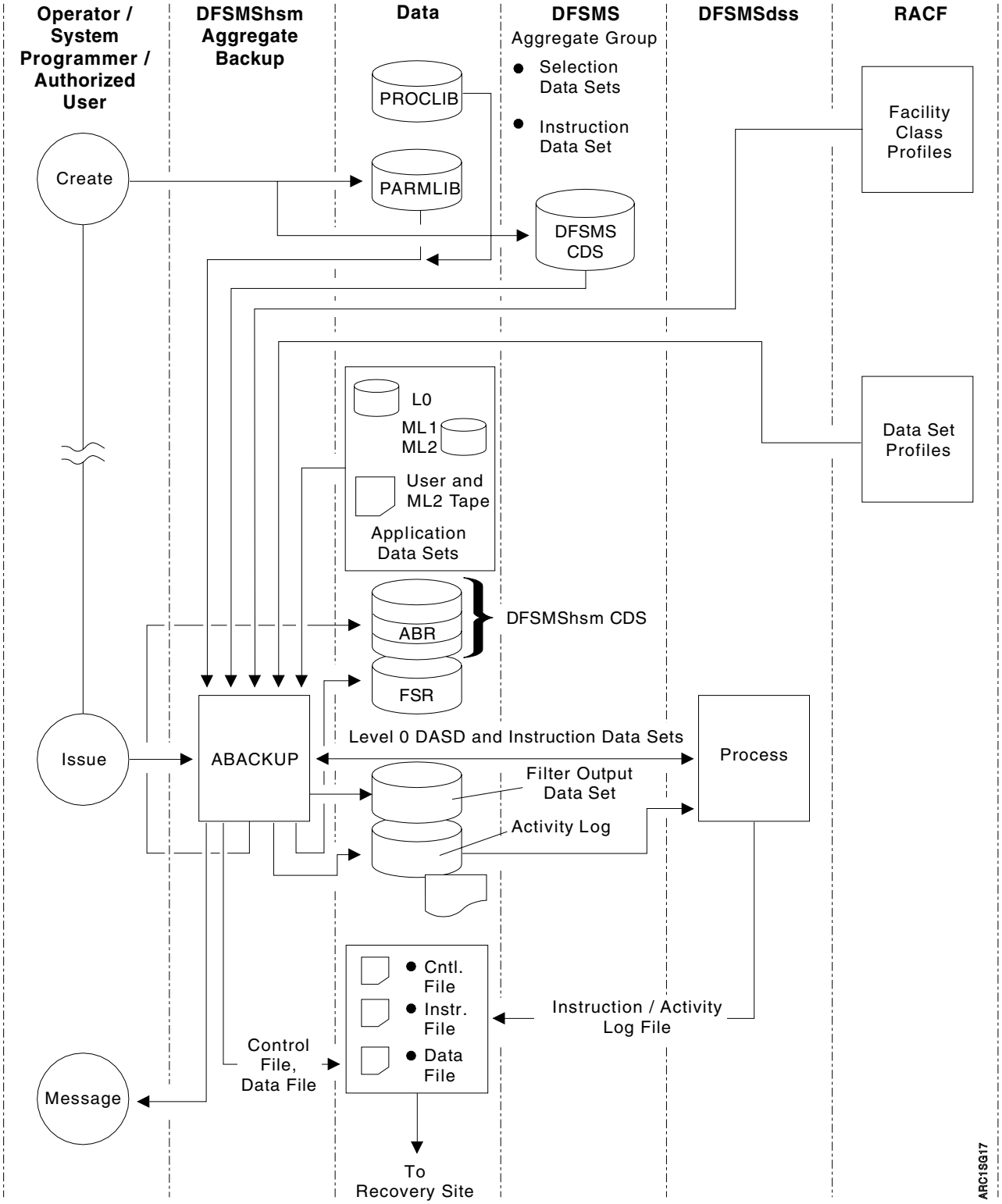


Figure 32. Aggregate Backup and Recovery Support (Part 1 of 2)

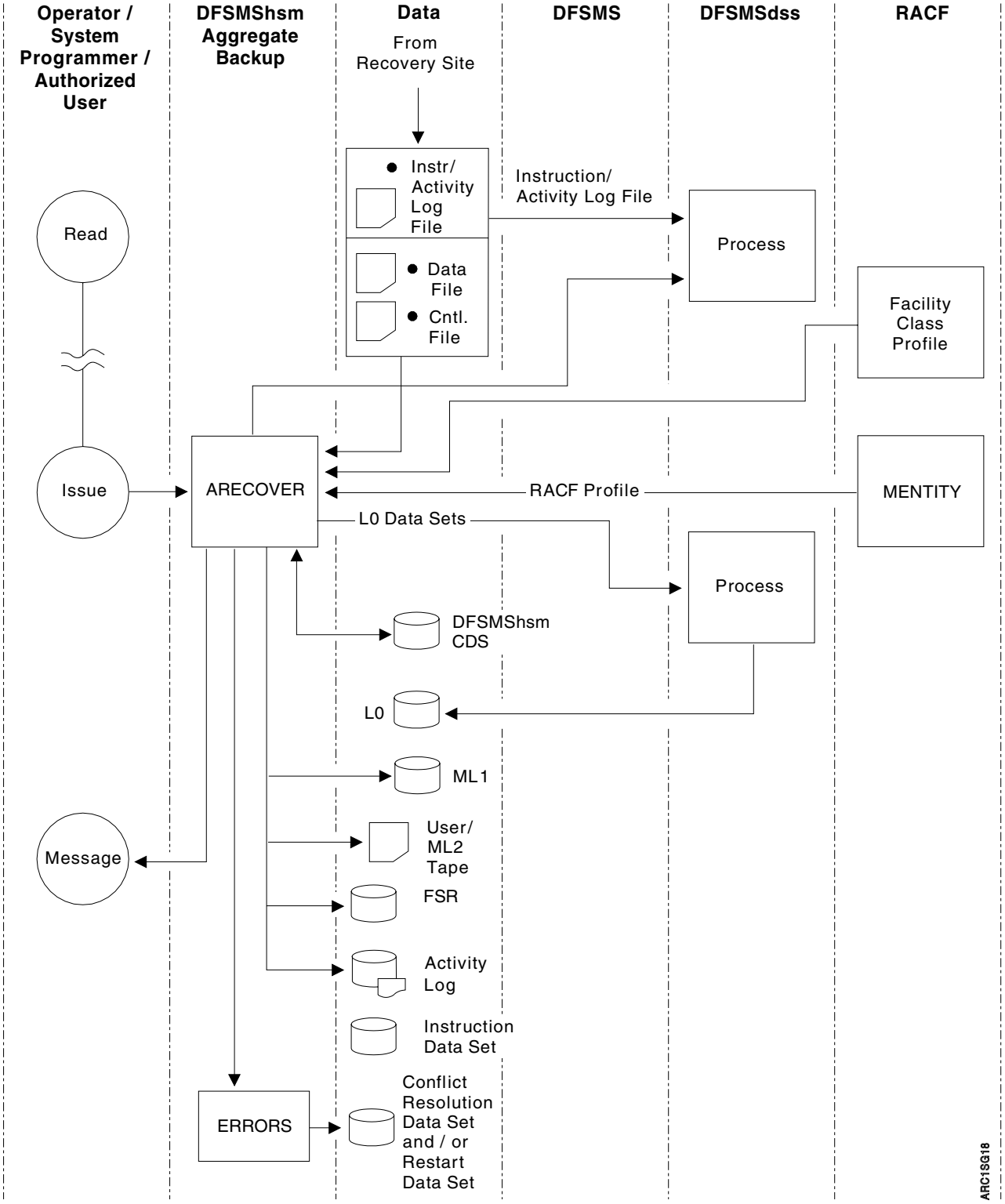


Figure 32. Aggregate Backup and Recovery Support (Part 2 of 2)

The aggregate group and management class used for aggregate backup are defined to DFSMS through Interactive Storage Management Facility (ISMF) panels.

DFSMSdss is used to move level 0 DASD data sets during aggregate backup and recovery processing.

RACF is used to ensure that RACF-protected data sets that are backed up remain RACF-protected when recovered. When RACF facility class is active and the user has restricted command authority, RACF checking of data sets is performed during aggregate backup. See “Checking Authorization during Aggregate Backup” on page 304 for more information.

Although the backup and recovery sites are referred to as two separate data processing centers, you can choose to perform both functions at the same data processing center. This might be useful if you want to divide your data processing center for work load management.

How Aggregate Backup Works

To use aggregate backup, you must first specify the data sets to be backed up. Then, you use the ABACKUP command or ISMF panels to back up the data sets to tape files. These tape files may then be physically transported or transmitted, using a transmission program like the NetView file transfer program (FTP), to a recovery site where the data sets can be recovered.

The following sections describe how data sets are specified for backup, how the ABACKUP command is used, which tape files are created by aggregate backup, and how to use the Aggregate Backup Assist (ABA) program to produce a selection data set.

Specifying Data Sets to Be Backed Up

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 are to be backed up. You may also decide to 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.

Aggregate Group

The aggregate group lists the selection data set names, instruction data set name, and additional control information used by aggregate backup in determining which data sets are to be backed up. You create an aggregate group using Interactive Storage Management Facility (ISMF) panels.

The aggregate group optionally contains a management class name that is used to obtain a management class definition that specifies the new aggregate backup parameters.

If a management class is not specified in the aggregate group definition, then DFSMShsm attempts to retrieve the contents of the default management class during aggregate backup processing. If a default management class has not been defined, then DFSMShsm defaults are used. See “Management Class Considerations” on page 301 for information about these defaults.

Management Class

The management class specifies such parameters as the maximum number of aggregate versions to be kept, the aggregate group retention parameters, the serialization option, and whether or not concurrent copy is requested.

Selection Data Set

The selection data set lists the names of the data sets to be processed during aggregate backup. You can identify the data set names in a single selection data set, or you can divide the names among as many as five selection data sets.

You can specify six types of data set lists in a selection data set: INCLUDE, EXCLUDE, ACCOMPANY, ALLOCATE, ACCOMPANYEXCLUDE, and ALLOCATEEXCLUDE. The type you specify determines which data sets are backed up and how they are recovered.

An INCLUDE data set list is a list of data sets to be copied by aggregate backup to a tape data file where they can be transported to the recovery site and recovered by aggregate recovery. The list can contain fully qualified data set names or partially qualified names with placeholders. DFSMShsm expands the list to fully qualified data set names.

An EXCLUDE data set list is a list of fully qualified or partially qualified data set names that are specifically excluded from being backed up. If the expansion of the partially qualified name results in one or more fully qualified names that should not be backed up, the EXCLUDE keyword prevents those data sets from being backed up.

An ACCOMPANY data set list is a list of tape data sets that physically accompany the control and data file from the backup site to the recovery site and that are cataloged as part of an aggregate recovery. These data sets are *not* written to the data file. You have to provide your own copy of the data set to the recovery site. For example, you may want to specify in the ACCOMPANY list any large tape data sets that would take a lot of time to copy to the data file.

An ACCOMPANYEXCLUDE data set list is a list of fully qualified or partially qualified data set names that are specifically excluded from being tracked in the control file during aggregate backup and eventually cataloged during aggregate recovery. If the expansion of the partially qualified name results in one or more fully qualified names that should not be processed, the ACCOMPANYEXCLUDE keyword prevents those data sets from being backed up.

An ALLOCATE data set list is a list of DASD data sets or GDG base names processed by aggregate backup and specified on the control file to be allocated and cataloged at the recovery site. ALLOCATE data sets are *not* copied to a data file. For example, you may want to specify an ALLOCATE data set for aggregate processing that requires the presence of an empty data set.

Note: An ALLOCATE data set list can also contain the names of ML2 migrated tape data sets.

An ALLOCATEEXCLUDE data set list is a list of fully qualified or partially qualified data set names that are to be specifically excluded from having allocation data collected during ABACKUP. If the expansion of the partially qualified name results in one or more fully qualified names that should not be allocated at the recovery site, the ALLOCATEEXCLUDE keyword prevents collecting allocation data for those data sets.

Skipped Data Sets

Regardless of how you specify data sets for backup, some data sets may be protected by the peer-to-peer remote copy (PPRC) function or by the extended remote copy (XRC) function. In either case, making backup copies is redundant

and a waste of storage space. Use the ABACKUP SKIP(PPRC) command to direct ABACKUP to skip backing up any level 0 data sets that are totally contained on PPRC volumes. Use the ABACKUP SKIP(XRC) command to direct ABACKUP to skip backing up any level 0 data sets that are totally contained on XRC volumes. In addition, you can combine the parameters in the ABACKUP SKIP(PPRC XRC) command form.

Note: You can also use the SETSYS ABARSKIP(PPRC) command or the SETSYS ABARSKIP(XRC) command (or a combination of both subparameters) to specify which types of data sets that you want all ABACKUP processing to skip.

Instruction Data Set

The instruction data set is a data set you create that contains instructions or other information you want to send to the recovery site. The instruction data set is a free-form text file.

Backing Up Applications

Aggregate backup allows you to back up user-defined groups of data sets to be recovered later as an entity at another computer site. Figure 33 shows the information flow of an aggregate backup.

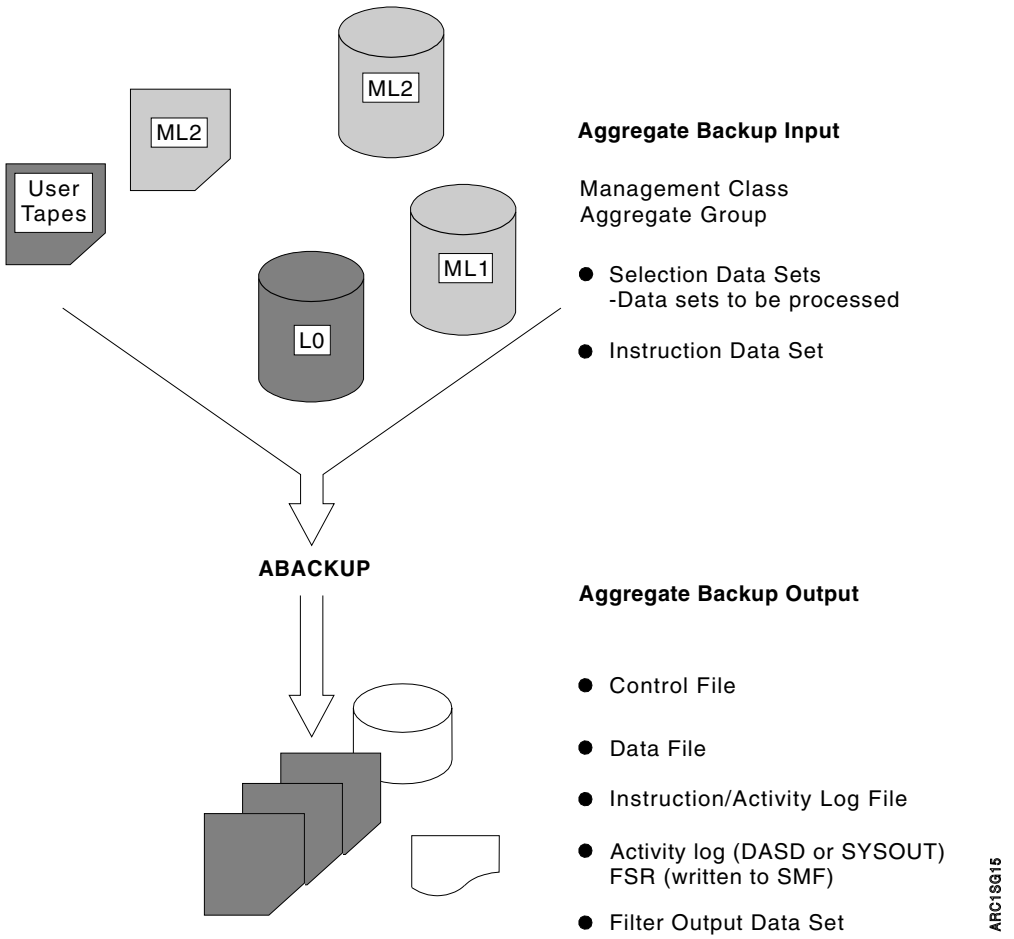


Figure 33. Aggregate Backup Information Flow

ABACKUP Command

You have two options with the ABACKUP command. You can choose to verify the contents of the aggregate backup without actually backing up any data sets, which is the same as performing a test run of aggregate backup. Then, when you are ready, you can perform the actual backup.

When you issue the ABACKUP command with the EXECUTE option, the following tape files are created for later use as input to aggregate recovery:

- Data file—Contains copies of the data sets that have been backed up.
- Control file—Contains control information needed by aggregate recovery to verify or to recover the application's data sets. The control file also specifies which ACCOMPANY and ALLOCATE data sets should be cataloged or allocated at the recovery site.
- Instruction/activity log file—Contains the instruction data set, which is optional. The instruction data set contains the instructions or information that may be needed at the recovery site. A copy of the activity log is also written to this file if the activity log is directed to DASD by the SETSYS ABARSACTLOGTYPE command.

  Note: This file is created only if an instruction data set name was specified in the aggregate group, or the ABACKUP activity log was allocated on DASD.

In addition, aggregate backup creates the following:

- Filter output data set—Created if requested with the FILTEROUTPUTDATASET parameter of the ABACKUP command.
- Activity log—Contains messages issued during aggregate backup processing.
- Functional statistics record (FSR)—Containing additional aggregate backup processing information. This includes records of the amount of CPU time used for processing ABACKUP and ARECOVER requests that can optionally be written as an SMF record if the installation specifies SETSYS SMF(*smfid*).

  Note: If the total number of volumes used for the data file, control file, instruction data set, and accompanying data set exceeds 5000, then an extension FSR is created for the aggregate. Each extension FSR contains a maximum of 5000 volume serial numbers. In addition, each extension FSR has the constant "CONT" at offset X'7C'. There are as many extended FSRs as necessary to hold all volume serial numbers processed by a particular aggregate. The extension FSRs repeat all aggregate information except the volume serial number lists.
- An ABARS activity record (ABR) is created in the BCDS. This record contains information related to an aggregate backup and recovery performed on a particular aggregate group.

Rule: You must have a BCDS when performing an aggregate backup or aggregate recovery.

Example of Data Sets Associated with an Aggregate Group

Table 18 on page 288 shows the data sets that exist in the database at the aggregate backup site. Throughout this chapter, this example is used to show each task as you prepare for and perform aggregate backup of some of these data sets.

Table 18. Data Sets at the Example Aggregate Backup Site

Data Set Name	DASD	Tape	SMS	Migrated
PAY2.PY002.SOCSEC	X		X	
PAY2.PY002.STATUS.G0001V00	X			
PAY2.PY002.STATUS.G0002V00	X			
PAY1.PY001.CHECK	X			
PAY1.PY003.CHECK	X			
PAY2.PY001.BENEFIT	X			
PAY1.PY001.PGMLIB	X			
PAY1.PY002.IRA	X			ML1
PAY1.PY001.BONUS	X			ML1
PAY1.PY002.LADDER	X			ML1
PAY1.PY001.NAMES		X		
PAY2.PY002.RETIRE		X		
PAY2.PY001.TAXES		X		

You want the following data sets to be copied to the data file:
- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002V00
- PAY1.PY001.CHECK
- PAY1.PY001.PGMLIB
- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY002.LADDER
- PAY1.PY001.NAMES

You want the data set PAY2.PY001.BENEFIT to be identified on the control file for allocation at the recovery site. You also want the data set PAY2.PY001.TAXES to be identified on the control file and to be cataloged at the recovery site.

Filtering for Generation Data Sets

Extra care must be taken when using wildcard filtering for generation data sets. You must use either of the following names:

- The fully qualified data set name

  Example: USER1.GDG.G0001V00
- The fully qualified name with relative generation

  Example: USER1.GDG(-1)
- The partially qualified base name with the last qualifier (GggggVvv) not specified

  Example: USER1.GDG.*

Partial qualification of the last qualifier (GggggVvv) is not supported unless a wildcard (% or *) is specified in the first character of the qualifier

Example: USER1.GDG.*V01 is valid but USER1.GDG.G* is not valid.

Another way to get all generations of a GDG using wildcards would be to specify USER1.GDG.%%%%%V%%.

Preparing for Aggregate Backup

You need to consider the following items as you prepare for aggregate backup:

- Minimizing data set contention on your system.

  When ABACKUP is processing data sets on level 0 volumes and the status of a data set changes (for example, it is moved or deleted) after the list of data sets to process has been created but before backup processing has begun, ABACKUP processing fails. Then ABACKUP processing must be entirely rerun.
- Specifying data sets in one or more selection data sets so they can be processed by aggregate backup
- Writing instructions and useful information in an optional instruction data set
- Defining an aggregate group to identify the selection data sets, instruction data set, and other control information for aggregate backup
- Defining a management class
- Defining aggregate backup environmental options with the SETSYS command
- Using RACF facility class to authorize certain users or operators to issue the ABACKUP command

Table 19 shows a checklist you might want to follow as you complete each of the steps needed to prepare for and perform aggregate backup.

Table 19. Aggregate Backup Checklist

___ 1.	Negotiate with the recovery site to provide the support required to recover your data sets.
___ 2.	Review the *z/OS and z/OS.e Planning for Installation* manual.
___ 3.	Define the aggregate group to identify the selection data sets, the instruction data set, and any additional control information.
___ 4.	Define the management class.
___ 5.	Create the selection data sets that list the data sets to be processed by aggregate backup. See *z/OS DFSMShsm Implementation and Customization Guide* , Chapter 4, under the heading "Data Mover Support for Availability Management Functions" for a list of data set types that are supported for aggregate backup. 1. Allocate the selection data sets. 2. Enter an INCLUDE list of the data sets to be backed up. 3. Enter an EXCLUDE list of the data sets that should not be backed up. This step is optional. 4. Enter an ACCOMPANY list of the tape data sets that should be taken with the aggregate backup data file and should be cataloged at the recovery site. This step is optional. 5. Enter an ACCOMPANYEXCLUDE list of data sets you want excluded from the ACCOMPANY list. This step is optional. 6. Enter an ALLOCATE list of data sets you want only allocated and cataloged at the recovery site. This step is optional. 7. Enter an ALLOCATEEXCLUDE list of data sets you do not want allocated and cataloged at the aggregate recovery site. This step is optional.
___ 6.	Create an instruction data set to contain any helpful information or instructions for the recovery site. This is optional.
___ 7.	Link-edit the aggregate backup installation exits. This step is optional.
___ 8.	Ensure that all level 0, ML1, and ML2 DASD volumes containing the data sets referred to for aggregate backup are online.
___ 9.	Ensure that the ABARS JCL procedure was installed in the systems JCL library (refer to the *z/OS DFSMShsm Implementation and Customization Guide*).

Table 19. Aggregate Backup Checklist (continued)

___ 10.	Determine the SETSYS parameters to be used: • Specify maximum number of ABARS address spaces • Specify activity log type, message level, and deletion processing • Specify whether output tapes should be stacked or not • Specify whether you want the ABARS FSR record written as an SMF record • Specify how DFSMSdss backs up level 0 DASD during ABACKUP processing • Indicate ABARS procedure name • Indicate TAPESECURITY(RACF), if desired • Activate desired installation exits • Specify ABARS buffer usage
___ 11.	Add the aggregate backup procedure name defined via SETSYS ABARSPROCNAME to the RACF started task table.
___ 12.	Use RACF facility class to authorize certain users or operators to issue the ABACKUP command.
___ 13.	If you want to RACF protect output tape volumes, issue the SETSYS command with the TAPESECURITY(RACF) parameter and issue the RACF command RDEFINE TAPEVOL HSMABR.
___ 14.	Issue the ABACKUP command with the VERIFY parameter. • If aggregate backup verification completes successfully (RC=0), continue to step 16. • If aggregate backup verification fails, correct any errors and repeat this step until the aggregate backup verification completes with a return code of zero.
___ 15.	Issue the ABACKUP command with the EXECUTE parameter.
___ 16.	Review the activity log for messages ARC6060I and ARC6061I that list the aggregate backup control and data files volumes. The control file volume serial numbers are required as input for aggregate recovery.

SMS Requirements during Aggregate Backup

You must have SMS active on your system to perform aggregate backup.

Prior planning is suggested if you want to back up SMS-managed data sets and have them recovered later in an environment compatible to that which existed at the backup site. It is recommended that the SMS classes retain the same attributes both at the backup and at the recovery sites. The ACS routines should also be compatible.

If you do not have identical SMS environments at both sites, SMS-managed data sets can still be successfully recovered, but not necessarily in the environment in which they existed at the backup site. The SMS-managed data sets are recovered in one of the ways described in "Considering Storage Management Subsystem (SMS) Requirements" on page 334.

Notes:

1. You might want to specify the required SMS class names and attributes in the instruction data set when you perform aggregate backup. See "Creating an Instruction Data Set" on page 297.
2. All SMS class names that are encountered during aggregate backup processing—along with their associated attributes—are listed in the ABACKUP activity log. This information can be retrieved during ARECOVER PREPARE processing and used to define the SMS environment at your recovery site. You should also consider backing up the source control data sets (SCDSs) as a separate aggregate group so that you can recover them as an aggregate before recovering application data.

Creating Selection Data Sets

A selection data set contains the names of the data sets to be processed during aggregate backup. You can identify the data set names in one selection data set, or you can split them up and list them in as many as five selection data sets.

For example, you may want to list a particular group of data sets in a separate selection data set that could then be used as part of more than one aggregate backup. Thus, a particular user's billing application could be listed in a separate selection data set to be backed up by itself. Then, if you decide you also want to back up the billing applications of all the users in your company, you can perform an aggregate backup using this same selection data set in addition to selection data sets for the other users.

A selection data set can be a partitioned data set (PDS) member, a sequential data set, or a partitioned data set extended (PDSE) member.

The Interactive Storage Management Facility (ISMF) panels allow you to allocate, browse, and edit the selection data sets.

If you create a selection data set outside of ISMF panels, you must allocate it as follows:

- Records must be in one of these two formats:
  - Fixed format, block size = 80, length = 80
  - Fixed block format, block size = any multiple of 80, length = 80
- Selection data sets must be cataloged.

Selection Data Set Syntax

You must follow these general rules when creating a selection data set:

- Entries must be entered between columns 1 and 72 inclusive.
- A comment is a string of characters preceded by **/*** and followed by ***/** symbols.
- A separator consists of a comma, one or more blanks, or a comment.
- INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE keywords are separated from one another by one or more separators.
- One or more blanks can optionally precede and follow each parenthesis in the pair.
- Continuation characters are not necessary but can be optionally specified by a hyphen (-) or plus (+) as the rightmost nonblank character, optionally preceded by one or more blanks. This follows the syntax conventions for Time-Sharing Option (TSO) CLIST.
- INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE keywords can be specified in any sequence, but they can be specified only once within a selection data set.
- The INCLUDE keyword must be specified (one INCLUDE keyword for each selection data set), but all other keywords are optional.
- Data set names specified in the INCLUDE list cannot be specified in the ALLOCATE or ACCOMPANY lists.
- If the same fully qualified data set name is specified in both the INCLUDE list and the EXCLUDE list, the data set is not selected for processing, because EXCLUDE takes precedence over INCLUDE.

If the same fully qualified data set name is specified in both the ACCOMPANY list and the ACCOMPANYEXCLUDE list, the data set is not selected for processing, because ACCOMPANYEXCLUDE takes precedence over ACCOMPANY.

If the same fully qualified data set name is specified in both the ALLOCATE list and the ALLOCATEEXCLUDE list, the data set is not selected for processing, because ALLOCATEEXCLUDE takes precedence over ALLOCATE.

- Fully qualified names are supported.
- PDS member names are not supported.
- Relative GDG names are supported when the GDG name is a fully qualified data set name without placeholders.

  Partially qualified relative GDG names are supported in the INCLUDE statement, but not in the EXCLUDE, ALLOCATE, ALLOCATEEXCLUDE, ACCOMPANY, or ACCOMPANYEXCLUDE statements.

  The following is an example of a partially qualified relative GDG name:

  ```
  INCLUDE(USERHLQ.*.GDG.*(-1))
  ```

  The statement above causes aggregate backup to back up all the minus-one generation data sets that are cataloged and match the specified filtering criteria. No other generations of the data set are backed up unless specifically requested via another data set filter or a fully qualified data set name.

  Rule: You may not specify positive generation (+*nnn*) numbers. Most positive generation numbers refer to data sets that are not yet created. In addition, the integer value within the parentheses must conform to MVS constraints and must be expressed in three digits or less.
- With the exception of relative GDG names (above), partially qualified names are supported in the INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE lists. The maximum number of partially qualified data sets supported is 1023. Use the following examples as guidelines for the use of ** and * :

PAY1.**	Specifies all data sets whose first or only qualifier is PAY1
USER2..LIST**	Specifies all data sets whose first and last qualifiers are USER2 and LIST, respectively, including USER2.LIST
PAY1.*	Specifies all data sets with two qualifiers, whose first qualifier is PAY1
PAY1.*.LOAD	Specifies all data sets with three qualifiers and whose first and last qualifiers are PAY1 and LOAD respectively
PAY1.UT*.LIST	Specifies all data sets with three qualifiers, whose first and last qualifiers are PAY1 and LIST, respectively, and whose second qualifier begins with UT

  When used with other qualifiers, the double asterisk (**) indicates either the nonexistence of leading, trailing, or middle qualifiers, or the fact that they play no role in the selection process.

  The single asterisk (*) is used in place of exactly one qualifier. It can also be used to indicate that only part of a qualifier has been specified.

The ignore character (%) is a place-holder character. Each specification of (%) represents exactly one character in the data set name. Any character in that position in the name is ignored. Use the following examples as guidelines for the use of (%)

PAY1.PY001.I%%	This matches a data set name of PAY1.PY001.IRA.
PAY1.PY001.B%%%S	This matches a data set name of PAY1.PY001.BONUS.

- Fully or partially qualified data set aliases are not supported.

ABARS supports both VSAM and non-VSAM data sets in the selection data set. ABARS allows VSAM base cluster names or VSAM component names in the selection lists. If VSAM component names are specified in the selection data set, ABACKUP processing determines the correct base cluster name that is associated with the component name and backs up the base cluster. If multiple VSAM component names are associated with the same base cluster name, the base cluster is backed up only once.

If a VSAM component name is encountered, ABACKUP processing backs up the associated base cluster and all components that are associated with that base cluster. When ABACKUP is invoking DFSMSdss, it passes DFSMSdss the base cluster name and requests SPHERE processing.

Note: If you are working with an ALLOCATE list, only the base cluster components will be processed.

When a VSAM data set is backed up from the INCLUDE list, all of its associations are backed up, including its components (base cluster, data, and index). However, when a VSAM data set is backed up from the ALLOCATE list, only the base cluster is backed up. This occurs because DFSMSdss dump processing moves actual data for a data set that is backed up from the INCLUDE list. No data (only allocation information that is necessary to allocate the data set at the recovery site) is moved for data sets backed up from the ALLOCATE list. In each instance, all information that is needed to restore the VSAM data set is backed up regardless in which list it appears. For a discussion of different data set types that ABARS supports in the INCLUDE and ALLOCATE lists, refer to the table called "Availability Management—Data Mover Support" in the *z/OS DFSMShsm Implementation and Customization Guide*.

Because of the way that ABARS handles VSAM components and because migrated VSAM data sets are cataloged as non-VSAM, the results of data set filtering can vary, depending on whether the data set is migrated and whether the VERIFY or EXECUTE keywords are specified in the ABACKUP command.

For example, if you have the following statements in your selection data set--

```
INCLUDE(hlq.TEST.CLUSTER.**)
EXCLUDE(hlq.TEST.CLUSTER)
```

--you could still back up the base cluster because the data and index components have not been excluded.

These statements work if the VSAM data set is not migrated. However, if the data set is migrated, the data component is located (ABARS does not know at the time it is VSAM) and the base cluster is backed up.

To ensure that all components of the above VSAM cluster are excluded, either exclude using a matching generic data set name; for example,

EXCLUDE(*hlq*.TEST.CLUSTER.**)

or exclude all fully qualified component names; for example,

EXCLUDE(*hlq*.TEST.CLUSTER.DATA,*hlq*.TEST.CLUSTER.INDEX)

Related reading:

- For additional information about how to use the INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE keywords, see "Selection Data Set Keywords".
- For additional information about the TSO CLIST command, see the *z/OS TSO/E Command Reference*.

Processing User Tape Data Sets in the Include List

User tape data sets in the INCLUDE list are supported up to a block size of 256K–1, which is currently the maximum size supported for tape devices. The tape data set is restored to its original size upon recovery, as long as the allocated tape device supports the block size.

Selection Data Set Keywords

The following six keywords can be used to identify data sets in a selection data set:

- INCLUDE
- EXCLUDE
- ACCOMPANY
- ACCOMPANYEXCLUDE
- ALLOCATE
- ALLOCATEEXCLUDE

The INCLUDE keyword is the only required keyword. All other keywords are optional.

INCLUDE: Data sets listed by this keyword are written to the aggregate backup data file during ABACKUP processing.

The INCLUDE keyword syntax is:

Keyword	Field
Include	*(dsn..., dsnmask...)*

- **Include** — Enter the whole word or an abbreviation of the word. The letter I is the minimum abbreviation allowed.
- *dsn..., dsnmask...* — Specifies the data set names or masks to be backed up. *dsnmask* represents a partially qualified data set name. Data sets to be backed up can be tape or DASD, migrated or nonmigrated, SMS-managed or non-SMS-managed. If a PDS is listed, all members are included in the aggregate backup. If you specify a fully qualified data set name, the data set must exist and must be cataloged, or aggregate backup fails.

Note: DFSMShsm backs up, as is, any migrated data set in the INCLUDE list, without recalling it first.

EXCLUDE: Data sets listed by this optional keyword are *not* written to the aggregate backup data file. If you specify the same fully qualified data set name in both the INCLUDE and EXCLUDE lists, the data set is not selected, because EXCLUDE takes precedence over INCLUDE.

The EXCLUDE keyword syntax is:

Keyword	Field
Exclude	*(dsn..., dsnmask...)*

- **Exclude** — Enter the whole word or an abbreviation of the word. The letter E is the minimum abbreviation allowed.
- *dsn..., dsnmask...* — Specifies the data set names or masks to be excluded from the aggregate backup. *dsnmask* represents a partially qualified data set name.

Note: Aggregate backup combines the INCLUDE and EXCLUDE lists from all selection data sets for processing. The combination of lists must result in the selection of at least one data set, or aggregate backup fails.

ACCOMPANY: Tape data sets listed with this optional keyword physically accompany the control and data files from the aggregate backup site to the aggregate recovery site and are cataloged as part of an aggregate recovery. These data sets are *not* written to the data file. You must provide your own copy of the data set to be used at the aggregate recovery site.

Example: You can have a large tape data set that would take too much time to copy to the data file. If you can specify this data set in the ACCOMPANY list, and the data set is cataloged at the aggregate recovery site. Then you must transport the tape containing the data set to the aggregate recovery site.

The ACCOMPANY keyword syntax is:

Keyword	Field
ACCOMPANY	*(dsn..., dsnmask...)*

- **ACCOMPANY** — Enter the whole word, or the alias AC.
- *dsn..., dsnmask...* — Specifies the partially and fully qualified tape data set names or masks to accompany the data file from the aggregate backup site to the aggregate recovery site.

ACCOMPANYEXCLUDE: Data sets listed by this optional keyword are excluded from the list of ACCOMPANY data sets. Data sets in the ACCOMPANY list must not be in the INCLUDE or ALLOCATE lists, and must be cataloged tape data sets.

The ACCOMPANYEXCLUDE keyword syntax is:

Keyword	Field
ACCOMPANYEXCLUDE	*(dsn..., dsnmask...)*

- **ACCOMPANYEXCLUDE** — Enter the whole word or an abbreviation of the word. ACCEXCLUDE is the alias, and ACCOMPANYE is the minimum abbreviation.
- *dsn..., dsnmask...* — Specifies the data set names or masks to be excluded from the aggregate backup. *dsnmask* represents a partially qualified data set name.

ALLOCATE: DASD data sets listed with this optional keyword have their space allocated and are cataloged at the aggregate recovery site as part of an aggregate recovery. These data sets are *not* written to the data file.

The ALLOCATE keyword syntax is:

Keyword	Field
ALLOCATE	*(dsn..., dsnmask...)*

- **ALLOCATE** — Enter the whole word or the alias AL.
- *dsn..., dsnmask...* — Specifies the partially and fully qualified DASD data set names or masks to be allocated and cataloged at the aggregate recovery site as part of an aggregate recovery. Data sets in the ALLOCATE list must not be in the INCLUDE or ACCOMPANY lists, and must be cataloged DASD data sets. The DASD data sets must conform to one of the following data set organization types:
  - Physical sequential (PS)
  - Partitioned organization (PO)
  - Direct access (DA)
  - Virtual storage access method (VSAM)
    - Key sequenced data sets
    - Linear data sets
    - Relative record data sets
    - Entry sequenced data sets
  - User catalog
  - GDG base name

  All other data set organizations are *not* supported.

  To determine how to allocate space at ARECOVER time, DFSMShsm recalls each migrated data set included in the ALLOCATE list.

Rule: During aggregate recovery, VSAM data sets from the ALLOCATE list must be recovered as SMS-managed data sets. Therefore, SMS must be active at the recovery site, and ACS filtering must direct these data sets to SMS-managed storage.

ALLOCATEEXCLUDE: Data sets listed by this optional keyword are excluded from the allocate list.

The ALLOCATEEXCLUDE keyword syntax is:

Keyword	Field
ALLOCATEEXCLUDE	*(dsn..., dsnmask...)*

- **ALLOCATEEXCLUDE** — Enter the whole word or an abbreviation of the word. ALLEXCLUDE is the alias, and ALLOCATEE is the minimum abbreviation.
- *dsn..., dsnmask...* — Specifies the data set names or masks to be excluded from the aggregate backup. *dsnmask* represents a partially qualified data set name.

Examples of Selection Data Sets

At the example site, two selection data sets are allocated: PAY1SDS1 and PAY1SDS2.

PAY1SDS1 contains the following lines:

```
INCLUDE(PAY1.**)
EXCLUDE(PAY1.PY%%3.**)
```

PAY1SDS1 causes aggregate backup to back up all data sets whose names begin with PAY1, excluding any data sets with the middle qualifier PY%%3. Use of the % character has been explained in a previous section.

PAY1SDS2 contains the following lines:

```
INC(PAY2.PY002.SOCSEC -
    PAY2.PY002.STATUS.G0002V00 -
    PAY2.PY002.RETIRE)
ACC(PAY2.PY001.TAXES)
ALLOC(PAY2.PY001.BENEFIT)
```

PAY1SDS2 causes aggregate backup to back up all data sets listed. In addition, PAY2.PY001.TAXES is processed as an ACCOMPANY data set and PAY2.PY001.BENEFIT is processed as an ALLOCATE data set.

Creating an Instruction Data Set

Creating the instruction data set is optional. The instruction data set contains instructions or any other information you think is necessary to assist the recovery site in recovering and running the application. It is an informal, free-form text file.

You may find it useful to include the following types of information in the instruction data set:

- SMS attributes
- Description of the application
- RACF environment
- Software requirements
- Hardware requirements
- Unique application execution instructions

If you decide to back up the instruction data set, you must create it before you issue the ABACKUP command. The instruction data set must be a sequential data set and it must be cataloged. ISMF panels allow you to allocate, browse, and edit the instruction data set.

The instruction data set can be recovered during the aggregate recovery prepare step, before actually recovering the aggregate data sets. This can be helpful, especially if people other than the backup site personnel will run the application at the recovery site.

You should include the instruction data set name in the aggregate group definition so that the instruction data set is identified to aggregate backup. During aggregate backup, the instruction data set is backed up to an instruction/activity log file.

Related reading:

- (For additional information about creating the instruction data set, see the *z/OS DFSMSdfp Storage Administration Reference*.
- For additional information about the aggregate group, see “Defining an Aggregate Group” on page 298.

Example Instruction Data Set

The instruction data set, PAYROLL.INSTRUCT, that was created at the example site is shown in Figure 34 on page 298. PAYROLL.INSTRUCT is used for the example site aggregate group, PAY1.

```
THIS IS THE INSTRUCTION DATA SET FOR AGGREGATE GROUP PAY1.

PERFORMING AGGREGATE BACKUP ON PAY1 BACKS UP THE CRITICAL PAYROLL FILES AT
THE END OF A PAY PERIOD. RELEVANT FILES INCLUDE THE EMPLOYEE ROSTER
(PAY1.PY001.NAMES), AMOUNT OF EACH PAYCHECK (PAY1.PY001.CHECK), AMOUNT FROM
EACH CHECK PLACED IN IRA AND SOCIAL SECURITY DEDUCTION
(PAY1.PY002.IRA AND PAY2.PY002.SOCSEC), AND AMOUNT (IF ANY) OF CHECK
THAT WAS A BONUS IN ADDITION TO SALARY (PAY1.PY001.BONUS).

TO ENSURE PROPER EXECUTION, VERIFY THAT CORRECT RELEASE LEVELS ARE PRESENT FOR
 DFSMShsm, DFSMSdss, AND DFSMSdfp AND THAT THE CORRECT SMS CLASSES ARE ACTIVE.

YOU WILL NEED THE FOLLOWING SMS CLASSES AND THEIR ASSOCIATED ATTRIBUTES
SET UP TO RECOVER THOSE FILES THAT WERE SMS-MANAGED AT ABACKUP, AND TO
ENSURE THAT THEY WILL BE SMS-MANAGED UPON RECOVERY.

DATA CLASS NAME: PHYSSEQ
DESCRIPTION: PHYSICAL SEQUENTIAL DATA SETS

 RECFM        : FB
 LRECL        : 80
 SPACE
   AVGREC     : U
   PRIMARY    : 50
   SECONDARY  : 50
 RETPD        : 175
 VOLUME COUNT : 1

STORAGE CLASS NAME: BATCHDASD
DESCRIPTION: DASD DATA SETS PROCESSED IN BATCH MODE

AVAILABILITY:
  GUARANTEED SPACE             : NO
  GUARANTEED SYNCHRONOUS WRITE : NO

MANAGEMENT CLASS NAME : BATCH
DESCRIPTION           : BATCH DATA SETS

EXPIRATION ATTRIBUTES
  EXPIRE AFTER DAYS NON-USAGE : 150
  EXPIRE AFTER DATE/DAYS      : 366

RETENTION LIMIT :
PARTIAL RELEASE : YES

MIGRATION ATTRIBUTES
  PRIMARY DAYS NONUSAGE : 7
  LEVEL 1 DAYS NONUSAGE : 45
  COMMAND/AUTO MIGRATE  : YES

GDG MANAGEMENT ATTRIBUTES
  ROLLED OFF GDS ACTION : EXPIRE

BACKUP ATTRIBUTES
  BACKUP FREQUENCY
  # OF BACKUP VERSIONS (DS EXISTS)  : 3
  # OF BACKUP VERSIONS (DS DELETED) : 1
  RETAIN DAYS ONLY BACKUP VERSION   : 366
  (DS DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS : 100
  ADMIN OR USER COMMAND BACKUP      : YES
  AUTO BACKUP                       : YES

EXECUTE ARECOVER WITH THE VERIFY OPTION TO VERIFY THE AGGREGATE RECOVERY.
```

Figure 34. Example of an Instruction Data Set

Defining an Aggregate Group

The aggregate group identifies the selection data sets, instruction data set, and additional control information required to perform aggregate backup. The aggregate group is an SMS construct you must define through a set of ISMF panels before you can perform aggregate backup. You must activate the newly defined SMS configuration before issuing the ABACKUP command.

Note: If you modify the aggregate group definition, the SMS configuration should be reactivated. This is not required if you are only changing the contents of the selection or instruction data sets.

You must specify the following information in the aggregate group:

Aggregate group name
The name can be up to 8 characters long.

Selection data sets
Insert the names of one-to-five selection data sets you created for this aggregate backup. See "Creating Selection Data Sets" on page 291.

Description
This is an optional description of the aggregate group.

Copies
Insert the number of copies of the ABARS output files to be created. The maximum number of copies is 15, and the default is one copy.

Output data set prefix
The prefix identifies the output data sets created by aggregate backup. These output data sets are created with the following naming conventions:

- *outputdatasetprefix*.D.C*nn*V*nnnn* and *outputdatasetprefix*.O.C*nn*V*nnnn* for the data file
- *outputdatasetprefix*.C.C*nn*V*nnnn* for the control file
- *outputdatasetprefix*.I.C*nn*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 the output data sets from one aggregate backup.

Example: If you specify the *outputdatasetprefix* as 'PAY1', the data, control, and instruction/activity log files would have the names PAY1.D.C01V0001, PAY1.O.C01V0001, PAY1.C.C01V0001, and PAY1.I.C01V0001, respectively.

Note: C*nn* is the copy number and V*nnnn* is the version number generated during the ABACKUP operation.

Since you can make up to 15 copies in a single ABACKUP, the copy number identifies to which copy the file belongs. The version number is the same for all copies made in a single ABACKUP operation.

Example: If an aggregate group definition indicates that two copies are to be made, the first time that you perform an ABACKUP function on that aggregate group, the following files are created:

```
PAY1.D.C01V0001
PAY1.O.C01V0001
PAY1.C.C01V0001
PAY1.I.C01V0001

PAY1.D.C02V0001
PAY1.O.C02V0001
PAY1.C.C02V0001
PAY1.I.C02V0001
```

The next time that you perform an ABACKUP on that aggregate group, you create the following files:

```
PAY1.D.C01V0002
PAY1.O.C01V0002
PAY1.C.C01V0002
PAY1.I.C01V0002

PAY1.D.C02V0002
PAY1.O.C02V0002
PAY1.C.C02V0002
PAY1.I.C02V0002
```

Related reading: For additional information about how to define the aggregate group by using ISMF panels, see the *z/OS DFSMSdfp Storage Administration Reference*.

Example Aggregate Group

Figure 35 and Figure 36 on page 301 show the ISMF panels used to create the aggregate group, PAY1, at the example site.

```
  Panel Utilities Scroll Help
--------------------------------------------------------------------------------
DGTDCAG1                    AGGREGATE GROUP DEFINE                 Page 1 of 2
COMMAND ===>


SCDS NAME  . . . . . : PAY1.SCDS
AGGREGATE GROUP NAME : PAY1

TO DEFINE AGGREGATE GROUP, SPECIFY:

  DESCRIPTION ===> APPLICATION BACKUP EXAMPLES
              ===>

  BACKUP ATTRIBUTES
    NUMBER OF COPIES  . . . . . . . 1              (1 to 15)
    MANAGEMENT CLASS NAME . . . . . MCLASS1        (1 to 8 characters, to be
                                                    defined in current SCDS)

  OUTPUT DATA SET NAME PREFIX . . . PAY1
                                                   (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 35. Aggregate Group ISMF Panel 1

```
DGTDCAG2                    AGGREGATE GROUP DEFINE                 Page 2 of 2
COMMAND ===>


SCDS NAME:              PAY1.SCDS
AGGREGATE GROUP NAME:  PAY1

TO EDIT A DATA SET, SPECIFY NUMBER ===>     (1, 2, 3, 4, 5, or 6)
  SELECTION DATA SETS:            (1 to 44 characters)
   1 ===> PAY1SDS1
     MEMBER NAME ===>             (1 to 8 characters)
   2 ===> PAY1SDS2
     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 ===> PAY1.INSTRUCT

USE ENTER TO PERFORM VERIFICATION; USE UP COMMAND TO VIEW PREVIOUS PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.
```

Figure 36. Aggregate Group ISMF Panel 2

Notice that the same name was specified for the aggregate group and the output data set prefix. This makes it easier to remember later which control and data files belong to a specific aggregate group.

Management Class Considerations

The ABARS aggregate group definition can contain a management class name that is used to obtain a management class definition. Use the management class definition to determine the following aggregate backup parameters:

1. The maximum number of aggregate group backup versions to be kept (a specific number of versions up to 9999, or NOLIMIT).

   Note: Specifying a limit allows the creation of a new version to roll off (delete) all existing versions that exceed the current limit.

   NOLIMIT specifies that no roll-off occurs.

2. Aggregate group backup RETAIN parameters for ABACKUP versions:

   ```
   RETAIN ONLY VERSION = DAYS  |  WEEKS | MONTHS | YEARS(nnnn) | NOLIMIT

   RETAIN EXTRA VERSIONS = DAYS  |  WEEKS | MONTHS | YEARS(nnnn) | NOLIMIT
   ```

   Note: These values do not affect the expiration date on the actual tape labels.

3. The serialization option:

   ```
   COPY SERIALIZATION = C | F | blank
   ```

   Specify C (continue) for “tolerate an enqueue failure”, and F (fail) for “do not tolerate an enqueue failure”. Blank means a null entry.

4. The ABACKUP COPY TECHNIQUE. You can specify:

   STANDARD Concurrent copy session not requested.

   PREFERRED Concurrent copy session requested for each data set: aggregate backup continues if a concurrent copy session is not obtained for a particular data set.

   REQUIRED Concurrent copy session requested for each data set: the entire aggregate backup fails if a concurrent copy session is not obtained for a particular data set.

In addition:

- DFSMShsm generates a default expiration date of 99365 for the ABARS output files. (Installation exit ARCEDEXT is called, if active, to allow you to specify expiration dates such as 99000 that have special meaning to tape management products).
- When rolling off an ABARS version, DFSMShsm expires tape data sets without regard to the tape expiration date.
- DFSMShsm saves the management class name in the ABR record so that EXPIREBV can obtain the latest retain values.

Notes:

1. If the management class specified in the aggregate group definition does not exist, ABACKUP fails.
2. If a management class is not specified in the aggregate group definition, ABACKUP continues using the default management class. If a default management class is not defined, ABACKUP uses the following DFSMShsm defaults:
   - RETAIN ONLY VERSION = NOLIMIT
   - RETAIN EXTRA VERSIONS = DAYS(14)
   - NUMBER OF VERSIONS = 2
   - COPY SERIALIZATION = STATIC
   - ABACKUP COPY TECHNIQUE = STANDARD
3. The default management class name is never stored in the ABR record.

DFSMShsm Coexistence Considerations for Aggregate Groups and Management Classes

In DFSMShsm, the aggregate group definition has been modified from previous versions of DFSMShsm to add the NUMBER OF COPIES and MANAGEMENT CLASS NAME attributes. The DESTINATION attribute is no longer used in DFSMShsm. The TOLERATE ENQUEUE FAILURE attribute has been moved to the management class and is now called the COPY SERIALIZATION attribute.

Use ISMF to redefine the aggregate group definitions when converting to DFSMShsm and DFSMSdfp. If the aggregate group is not redefined, the existing aggregate group definitions are still valid, but all new aggregate group attributes, as well as all new management class aggregate backup attributes, revert to defaults.

The COPIES attribute in the aggregate group will default to 1.

The defaults for the management class attributes are:
- RETAIN ONLY VERSION = NOLIMIT
- RETAIN EXTRA VERSIONS = DAYS(14)
- NUMBER OF VERSIONS = 2

- COPY SERIALIZATION = FAIL
- ABACKUP COPY TECHNIQUE = STANDARD

The NUMBER OF VERSIONS aggregate backup attribute in the management class is the only attribute that ABARS processing looks at when rolling off excess ABARS versions. It does not consider the RETAIN ONLY VERSIONS or RETAIN EXTRA VERSIONS attributes during this process (these attributes are only considered when the EXPIREBV ABARSVERSIONS command is issued).

The NUMBER OF VERSIONS management class attribute works like the LIMIT parameter used when defining a GDG base. Each time an ABACKUP is performed for a particular aggregate it determines if this will create an excess version; if so, the oldest version is rolled off without regard to the RETAIN ONLY VERSIONS or RETAIN EXTRA VERSIONS attributes.

The RETAIN ONLY VERSION management class attribute is used during EXPIREBV processing when expiring excess ABARS versions. This attribute indicates how long the latest ABARS version should be retained.

The RETAIN EXTRA VERSIONS management class attribute is also used during EXPIREBV processing when expiring excess ABARS versions. This attribute indicates how long any ABARS version other than the latest version should be retained.

EXPIREBV processing for excess ABARS versions does not consider the NUMBER OF VERSIONS attribute when expiring ABARS versions.

The UNIT attributes that apply to the RETAIN ONLY VERSION and RETAIN EXTRA VERSIONS attributes are calculated in the following way:
- WEEKS are multiplied by 7
- MONTHS are multiplied by 30
- YEARS are multiplied by 365

Protecting Aggregate Backup Tape Volumes

Before aggregate backup control file, data file, and instruction/activity log file tape volumes can be protected with RACF, you must first define an aggregate backup tape connected set by issuing the following command:

```
RDEFINE TAPEVOL HSMABR
```

RACF must be active and the SETSYS TAPESECURITY(RACF) or TAPESECURITY(RACFINCLUDE) options must be in effect for DFSMShsm to use RACF to protect aggregate backup output tape volumes and add them to the HSMABR tape connected set. For more RACF information see “RACF Environment for DFSMShsm” on page 576.

Note: These tapes are not automatically added to RACF unless the above conditions are met at the recovery site. If these conditions are met, the tapes are added to RACF (at the recovery site) when the ARECOVER command is issued for this aggregate.

If TAPESECURITY(RACF | RACFINCLUDE) is specified and TAPESECURITY(EXPIRATION | EXPIRATIONEXCLUDE) is *not* specified, then the expiration date is not written to the tape header label, and the ARCEDEXT installation exit is not invoked.

Tape volumes added to the HSMABR RACF tape connected set are automatically removed by aggregate version rolloff processing during aggregate backup processing, during EXPIREBV processing, or if an aggregate backup failure occurs after the output tapes are opened. A tape volume may be deleted from the HSMABR tape connected set by entering the following RACF command:

RALTER TAPEVOL HSMABR DELVOL(*volser*)

Checking Authorization during Aggregate Backup

By using RACF FACILITY class profile support, you can permit all console operators or any users, including those who are not DFSMShsm authorized, to issue the ABACKUP command.

If the RACF FACILITY class profile support is installed and activated, DFSMShsm checks the RACF profile of the console operators or users for authorization to issue the ABACKUP command. The user ID of OPER must be defined to RACF before console operators can be given authority to issue the ABACKUP command. If the user ID OPER is given authority to the ABACKUP command, then all console operators are authorized to issue the ABACKUP command. You can also define group names to RACF, using the RACF ADDUSER command.

If the RACF FACILITY class profile is not active, only DFSMShsm-authorized users and console operators are authorized to issue the ABACKUP command.

You can define the DFSMShsm ABACKUP command to RACF by defining the following profile names:

Command	Parameter	RACF Profile
ABACKUP		STGADMIN.ARC.ABACKUP
ABACKUP	agname	STGADMIN.ARC.ABACKUP.agname

Through the use of the RACF PERMIT command, you can authorize specific user IDs to issue these commands.

Related reading:

- For additional information about the RACF FACILITY class control of access to all commands, see "Controlling Command Access with RACF FACILITY Class" on page 582.
- For additional information about the RACF FACILITY class control of access to all commands, see *z/OS DFSMShsm Implementation and Customization Guide*.

Authorizing Comprehensive ABACKUP Command Authority

When users are added to the STGADMIN.ARC.ABACKUP profile, they receive comprehensive command authority. When users are also authorized to the STGADMIN.ARC.ABACKUP.agname profile, they receive RESTRICTED instead of COMPREHENSIVE command authority. When users have COMPREHENSIVE command authority, they may then issue the ABACKUP command for any aggregate group, and no RACF authorization checking is performed on the data sets being processed during aggregate backup.

When you define a RACF FACILITY class profile for the DFSMShsm ABACKUP command, the Universal Access (UACC) keyword on the RDEFINE command

defaults to NONE. This means that the ABACKUP command cannot be issued until a user is authorized to that profile using the RACF PERMIT command.

You can issue the following RACF command to define profiles that prevent or limit the authority of users to issue the ABACKUP command:

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

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

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

Related reading:

- For additional information about the RACF FACILITY class control of access to all commands, see "Controlling Command Access with RACF FACILITY Class" on page 582.
- For additional information about the RACF FACILITY class control of access to all commands, see *z/OS DFSMShsm Implementation and Customization Guide*.

Authorizing Restricted ABACKUP Command Authority

When users are added to the STGADMIN.ARC.ABACKUP.*agname* profile, they receive RESTRICTED command authority. If users with this RESTRICTED authority issue the ABACKUP command, they must have a minimum of READ access to all RACF-protected data sets in the aggregate group. If they do not have READ or READ/WRITE access to a RACF-protected data set, the ABACKUP command fails for that data set during verification processing. Each data set that fails authorization checking is listed in the aggregate backup activity log.

If a data set is not RACF protected, it does not fail during verification processing because no RACF authorization checking is performed on it.

You can issue the following RACF command to define profiles that prevent or limit the authority of users to issue the ABACKUP command with a specific aggregate group name:

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP.PAY1
```

The following RACF command authorizes all console operators to issue the ABACKUP command with a specified aggregate group name:

```
PERMIT STGADMIN.ARC.ABACKUP.PAY1 CLASS(FACILITY) -
  ID(OPER) ACCESS(READ)
```

Related reading:

- For additional information about the RACF FACILITY class control of access to all commands, see "Controlling Command Access with RACF FACILITY Class" on page 582.
- For additional information about the RACF FACILITY class control of access to all commands, see *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the ABARS Facility Class, see the *z/OS DFSMShsm Implementation and Customization Guide*. for more information on RACF generic profiles.

Using the Aggregate Backup Installation Exits

Programming Interface Information

Four installation exits are used by aggregate backup if they are accessible to the LOAD macro and if SETSYS EXITON(ARCM2EXT), SETSYS EXITON(ARCBEEXT), SETSYS EXITON(ARCTVEXT), and SETSYS EXITON(ARCEDEXT) commands have been issued. (You can use the QUERY SETSYS command to display the status of these exits.)

ARCM2EXT is called by aggregate backup when a data set on an ML2 volume is encountered, and it allows you to specify if the data set should be skipped or included in the aggregate backup.

ARCBEEXT is called when:
- A data set encounters an input data set I/O
- A CDS I/O is encountered
- An SDSP allocation error occurs
- An uncataloged data set is encountered
- A data set fails serialization of the MAJOR NAME = ARCDSN, MINOR NAME = *datasetname* resource
- An error occurs during DFSMSdss dump processing

ARCBEEXT allows you to specify if the data set should be skipped so that the aggregate backup can complete successfully, or if it should be included in the aggregate backup and cause aggregate backup to fail.

ARCEDEXT is called to allow you to specify a tape volume expiration date other than 99365 for the aggregate backup output files.

ARCTVEXT is called to indicate to tape management systems, excluding DFSMSrmm, that ABARS is expiring a tape volume under any of the following conditions:
- When rolling off old ABARS versions during ABACKUP processing
- When expiring ABARS versions (EXPIREBV ABARSVERSIONS)
- When ABACKUP processing fails

End of Programming Interface Information

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

Ensuring That the Necessary Volumes Are Online

Before performing aggregate backup, ensure that all level 0, ML1, and ML2 DASD volumes containing data sets to be processed by aggregate backup are online. If these volumes are not online during aggregate backup processing, a mount request is issued.

Defining SETSYS Parameters for Aggregate Backup

The SETSYS command allows you to externally define various aggregate backup environmental options.

The system programmer should include SETSYS commands in the DFSMShsm parmlib member (ARCCMD*xx*) to ensure that the settings are defined to

DFSMShsm at startup initialization. If you are an authorized DFSMShsm user, you can issue the SETSYS command before an aggregate backup to modify the SETSYS parameters that control aggregate backup processing.

To specify how many ABACKUP commands can run concurrently, use the MAXABARSADDRESSSPACE parameter. MAXABARSADDRESSSPACE specifies the maximum number of concurrent ABARS secondary address spaces that DFSMShsm supports, up to 64. When determining the value to specify for MAXABARSADDRESSSPACE, you should factor in such items as system load, resource contention, and aggregate backup and recovery activity. The overriding factor is the number of tape drives available. Aggregate backup uses at least one tape drive for each address space and uses two if backing up user tape data sets or data sets residing on ML2 tape (the default for MAXABARSADDRESSSPACE is 1).

Note: When an ABACKUP command completes, ABARS will determine if there is another ABARS request queued that has not been processed. If there is, DFSMShsm will reuse the address space to process that command. The address space will remain active until there are no further ABARS commands queued, or the SETSYS MAXABARSADDRESSSPACE command has reduced the number of ABARS address spaces and this address space is no longer required.

Rule: If you issue two ABACKUP commands with the same aggregate group name, they cannot execute concurrently even if MAXABARSADDRESSSPACE is set to a value greater than 1.

The ABARSPROCNAME parameter specifies the procedure name that is used to start an ABARS secondary address space when no such address space is available for reuse.

The EXITON(*exitname*) parameter lets you specify which installation exits (ARCBEEXT, ARCEDEXT, ARCM2EXT, ARCTVEXT) are active during aggregate backup processing.

Each time you issue the ABACKUP command, an activity log is created. You can use the SETSYS ABARSACTLOGTYPE command to specify whether you want the ABACKUP activity log automatically written to SYSOUT or to DASD. You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether or not you want DFSMShsm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs.

You can use the SETSYS ABARSBUFFERS(*n*) command to specify the number of ABARS I/O buffers, up to a maximum of nine buffers. DFSMShsm multiplies the number you specify by two to determine the total number of buffers to use for I/O processing (one-half for input, one-half for output). If the number of buffers is not specified by the SETSYS ABARSBUFFERS command, a default setting of one is used.

Use the ABARSKIP(PPRC XRC) parameter on the SETSYS command to specify that you want the backup to skip level 0 data sets that are protected by the peer-to-peer remote copy function or by the extended remote copy function.

The SETSYS ABARSTAPES(STACK) command allows you to place (stack) the ABACKUP output files from a single aggregate group on a minimum number of

tape cartridges (as few as one). The stacking function applies only to tape cartridges. Attempts to redirect ABACKUP output files to DASD when the STACK option is in effect causes ABACKUP to fail.

Stacked tapes can only be used for ARECOVER processing on hosts running DFSMS/MVS Version 1 Release 4 or subsequent releases. The default, if you don't specify the SETSYS ABARSTAPES command, is the STACK option. Therefore, the SETSYS ABARSTAPES(NOSTACK) command must be specified if you want to direct the ABACKUP output files to DASD, or if the output tapes need to be recovered on a host running a DFSMS/MVS release prior to Version 1 Release 4.

Note: When issuing an ABACKUP command, you can temporarily override the SETSYS ABARSTAPES value by using the STACK | NOSTACK parameters of the ABACKUP command.

You can select which DFSMSdss messages should be written to the ABACKUP activity log by using the SETSYS ABARSACTLOGMSGLVL(FULL | REDUCED) command. If you specify SETSYS ABARSACTLOGMSGLVL(FULL), all DFSMSdss messages are intercepted and written to the ABACKUP activity log. If you specify SETSYS ABARSACTLOGMSGLVL(REDUCED), only the DFSMSdss messages ending in a "W" (warning messages) or "E" (error messages) are written to the ABACKUP activity log; no informational messages are written.

ABARS records the CPU time for processing ABACKUP requests, along with any applicable account code information. This information is written to the ABR record and can be obtained using the LIST AGGREGATE command. It is also written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANy system data sets. Installations can use the information to calculate charge backs for ABARS requests.

The SETSYS ABARSOPTIMIZE(*n*) command allows installations to adjust performance when invoking DFSMSdss to back up level 0 DASD data sets that are specified in an ABARS INCLUDE data set list. You can specify that DFSMSdss read one, two, or five tracks at a time, or one cylinder at a time. You can override the SETSYS ABARSOPTIMIZE value on an individual ABACKUP command by issuing the ABACKUP command with the OPTIMIZE(*option*) parameter.

Note: If the SETSYS DEBUG option is in effect when the ABACKUP command begins processing, DFSMShsm cannot move any data. The DEBUG option causes all ABACKUP commands to be processed as if the VERIFY option had been specified.

Related reading:

- For additional information about the SETSYS commands, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the SETSYS commands, see the *z/OS DFSMShsm Storage Administration Reference*.
- For additional information about the ABARSPROCNAME parameter, see the *z/OS DFSMShsm Storage Administration Reference*.

Example SETSYS Commands For Aggregate Backup

The following SETSYS commands are entered for the example site before performing aggregate backup:

SETSYS MAXABARSADDRESSSPACE(2) ABARSPROCNAME(SAMPLE)

SETSYS ABARSACTLOGTYPE(DASD)

The maximum number of secondary address spaces is set to two, which means that only two ABARS commands can run at a time. The procedure name used to start the secondary address space is identified as SAMPLE, and the activity log will be allocated to DASD.

To review the SETSYS parameters in effect for aggregate backup and recovery enter:

QUERY ABARS

Determining the Unit Type for the Control and Data Files

Before performing aggregate backup, you must determine the tape unit type to be allocated for the data, control, and instruction/activity log files. The unit type must be a supported tape device, as specified in the *z/OS DFSMShsm Implementation and Customization Guide*. It is important that the unit type is compatible with the unit type that will be used at the aggregate recovery site.

You can specify the unit type in one of two ways:

1. Use the SETSYS ABARSUNITNAME(*unitname*) command. For *unitname*, substitute the name of a valid unit type.

   If you use this method, you can issue ABACKUP and ARECOVER commands without using their UNIT parameters. When DFSMShsm processes the ABARS control, data, and instruction/activity log files, it uses the unit type specified in the SETSYS ABARSUNITNAME command.
2. Specify the unit type later as part of the ABACKUP command. See "ABACKUP Command" on page 311.

Note: If you do not specify the unit type by using the SETSYS ABARSUNITNAME parameter or if you do not use the UNIT parameter with individual ABACKUP or ARECOVER commands, DFSMShsm selects a default unit type of 3590-1.

Minimizing Data Set Contention

The following are recommendations on how to minimize data set contention during aggregate backup. **If you do not follow these recommendations, aggregate backup may fail.**

- Do not perform ABACKUP processing during AUTOMIGRATION processing.
- Do not perform ABACKUP processing during AUTOBACKUP processing if you are processing VSAM data sets.
- Do not use aggregate backup while the application is in use.
- Use the concurrent copy feature of ABARS.

Performing ABACKUP processing concurrent with volume dumping or while backing up non-VSAM data sets is not a problem.

Data sets cannot be migrated or recalled during aggregate backup processing.

When ABACKUP processes data sets on level 0 volumes and the status of a data set changes (for example, it is moved or deleted) after catalog information about

the list of data sets to process has been created (but before serialization), ABACKUP processing fails. The ABACKUP processing must be entirely rerun.

Data Sets That DFSMShsm Dynamically Allocates during Aggregate Backup Processing

DFSMShsm dynamically allocates the following data sets (SYSIN, FILTER, and authorization-checking) during aggregate backup processing. These data sets are used while aggregate backup is executing and are dynamically deallocated before completion of aggregate backup processing. These data sets are allocated with a unit type of SYSALLDA. (If they are non-SMS-managed, you must have at least one STORAGE- or PUBLIC-type device online to use the system.) You need to be aware of these data set names so that the high-level qualifiers used by ABARS can be defined to the system. You may want to modify your ACS filters to direct allocation of these data sets to specific volumes.

SYSIN data set: This data set is used to pass DUMP parameters to DFSMSdss.

The SYSIN data set name is:

authid.SYSIN.*agname*.T*hhmmss*.D*yyyyddd*

Where

authid is the DFSMShsm-authorized user ID
agname is the aggregate group name that is processed
hhmmss is hours, minutes, seconds
yyyyddd is the four-digit year followed by the Julian day of the year

FILTER data set: This data set is used to pass a list of data set names to DFSMSdss.

The FILTER data set name is:

authid.FILTER.*agname*.T*hhmmss*.D*yyyyddd*

Where

authid is the DFSMShsm-authorized user ID
agname is the aggregate group name that is processed
hhmmss is hours, minutes, seconds
yyyyddd is the four-digit year followed by the Julian day of the year

Authorization-checking data set: This data set is used when checking RACF authorization for user tape data sets.

The authorization-checking data set name is:

authid.D*yyddd*.T*hhmmss*

Where

authid is the DFSMShsm-authorized user ID
yyddd is the year and Julian day
hhmmss is hours, minutes, seconds

Performing Aggregate Backup

Figure 37 on page 311 shows what occurs when you invoke aggregate backup by issuing the ABACKUP command.

The following sections describe the aggregate backup invocation in more detail.

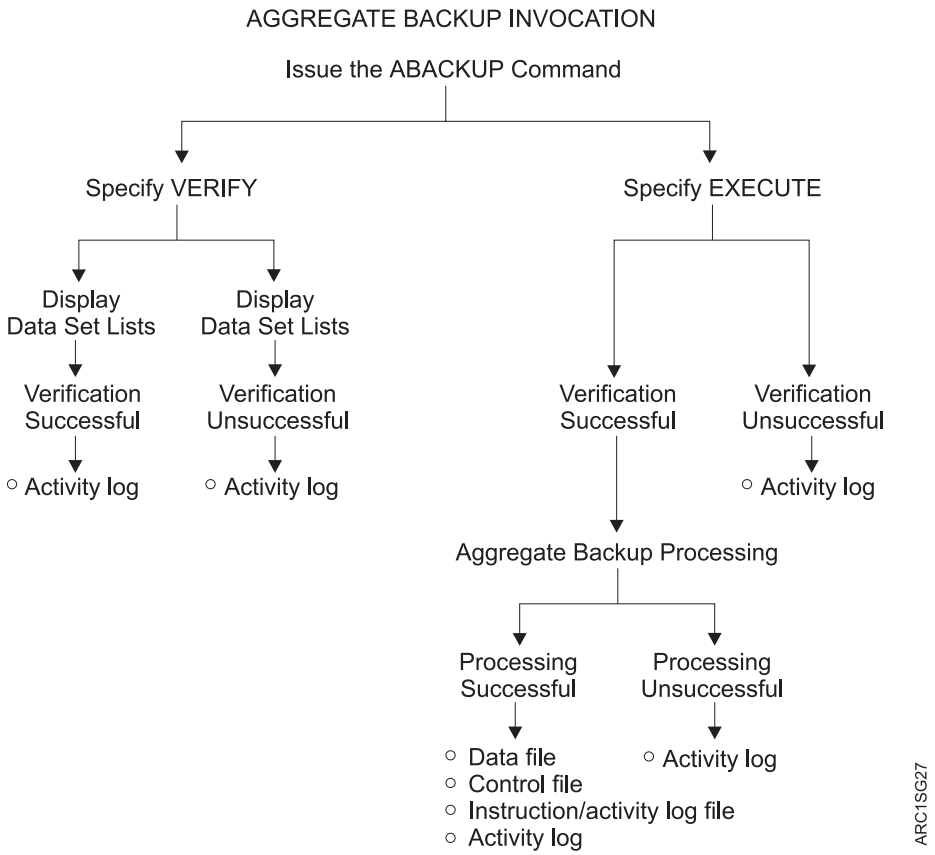


Figure 37. Aggregate Backup Invocation

ABACKUP Command

To perform aggregate backup, Issue the ABACKUP command. If you only want to verify the content of an aggregate backup without actually having the data sets backed up, specify the VERIFY parameter. This is like performing a test run of aggregate backup. After aggregate backup verification processing completes, the ABACKUP command stops. No data sets are backed up.

Note: If you have specified the FILTEROUTPUTDATASET parameter on the ABACKUP command, verification processing will place the results of the data set filtering into the data set specified. These results consist of a list of data sets selected for ABACKUP processing. This list can be useful when developing the data set filters in the selection data set.

When you are ready to have your data sets backed up, you can issue the ABACKUP command and specify the EXECUTE parameter. After successful data set verification processing, your data sets are backed up.

The following sections describe in more detail what occurs when you issue the ABACKUP command with the VERIFY or EXECUTE parameter.

The *agname* and EXECUTE | VERIFY parameters are required on the ABACKUP command.

For *agname*, specify the name of the aggregate group you defined earlier. See "Defining an Aggregate Group" on page 298.

The VERIFY parameter lists all input tape volumes that are required at the aggregate backup site if a subsequent ABACKUP EXECUTE command is submitted. It also performs verification processing of both migrated and nonmigrated data sets, detecting validation errors before data movement actually takes place.

For *unittype* in UNIT(*unittype*), specify the tape unit type to be allocated for the aggregate backup data, control, and instruction/activity log files.

After successful aggregate backup of an application, the optional MOVE parameter causes ABACKUP to delete all data sets in the application. The different data sets are affected in the following manner:

- DASD data sets are deleted
- Tape data sets are uncataloged
- ACCOMPANY data sets are uncataloged
- ALLOCATE data sets are deleted from DASD with the exception of user catalogs in the ALLOCATE list
- MIGRATED data sets are deleted

Note: If users with restricted RACF command authority issue the ABACKUP command with the MOVE parameter, they must have ALTER access authority for each data set belonging to the aggregate group, or aggregate backup verification fails with a related error message.

If a user has comprehensive RACF command authority and issues the ABACKUP command with the MOVE parameter, no RACF checking of individual data sets is required.

The optional FILTEROUTPUTDATASET(*dsname*) parameter allows DFSMShsm to direct a fully-qualified list of data set names from the filtering criteria from the INCLUDE/EXCLUDE lists, as well as the names of ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE data sets, to a separate output data set. If the user has restricted ABACKUP command authority, then that user must have a minimum of update authority to append to the existing RACF-protected filter output data set or allocation authority to create a new filter output data set. For *dsname*, substitute a data set name of up to 44 characters. FILTEROUTPUTDATASET can be abbreviated FODS.

When the FILTEROUTPUTDATASET parameter is specified along with the PROCESSONLY parameter, the filter output data set contains the data sets that pass the filtering criteria and also match the category designated in the PROCESSONLY parameter.

When you use the optional OPTIMIZE(*option*) parameter with an ABACKUP command it overrides the value specified by the SETSYS ABARSOPTIMIZE command. The OPTIMIZE parameter allows you to specify the performance level used by DFSMSdss during dump processing of level 0 DASD data sets. You can specify that DFSMSdss read one, two, or five tracks at time, or one cylinder at a time.

The optional PROCESSONLY parameter directs DFSMShsm to process only the specified categories. You can specify one or more of the categories L0, ML1, ML2,

and USERTAPE. The categories can be specified in any order. Specification of all categories will be accepted and is equivalent to not specifying the PROCESSONLY parameter.

Restriction: During ABACKUP, the PROCESSONLY parameter is processed only for data sets that exist (cataloged) during the creation of the control file. If the data set is deleted between the time that data set filtering is complete and the time that there is a successful search (locate) for it during the building of the control file, there is no way to determine where the data set (L0, ML1, ML2, or USERTAPE) resides, nullifying any meaning the PROCESSONLY option would have had.

PROCESSONLY can be abbreviated to PONLY. The following are recognized aliases for category names:

Category	**Alias(es)**
L0	LEVEL0, PRIMARY
ML1	MIGRATIONLEVEL1
ML2	MIGRATIONLEVEL2

The optional, and mutually exclusive, STACK and NOSTACK parameters allow you to override the specified SETSYS ABARSTAPES value, or its default value of STACK. The STACK option causes ABACKUP to place the ABACKUP output files on a minimum number of tape cartridges. The stacked output files can only be recovered by hosts running DFSMS/MVS Version 1 Release 4, or subsequent releases. The STACK option only applies to output being directed to tape. Using the STACK option and attempting to direct the ABACKUP output files to DASD causes ABACKUP to fail.

The NOSTACK option causes ABACKUP output to be handled in the same manner it was handled prior to DFSMS/MVS Version 1 Release 4.

The SKIP option causes DFSMShsm to skip data sets on volumes that are protected by PPRC or XRC. That is, these data sets are not backed up.

You might find it helpful to use the following DFSMShsm commands to manage and query an ABACKUP command:

- HOLD ABACKUP prevents all currently queued aggregate backup requests from being processed.
- HOLD ABACKUP(*agname*) prevents currently queued requests for a specific aggregate group from being processed.
- HOLD ABACKUP EOD prevents currently queued requests from being processed, and additionally stops processing of any currently running ABACKUP requests.
- HOLD ABACKUP(*agname*) EOD stops processing of the aggregate group specified, and prevents further requests for this aggregate group from being selected for processing.
- RELEASE ABACKUP releases all queued aggregate backup requests that have been held by using the HOLD ABACKUP command or by DFSMShsm.
- RELEASE ABACKUP(*agname*) releases all queued requests for the specified aggregate group.
- STOP causes an orderly shutdown of DFSMShsm after any currently running ABACKUP commands complete processing.

- STOP DUMP causes DFSMShsm to end processing immediately and to provide a dump. Any ABACKUP command in process is canceled immediately.
- QUERY or HQUERY can display information relevant to ABACKUP.
- CANCEL or HCANCEL can cancel any existing queued ABACKUP.

Related reading: For additional information about the commands listed in this section, see the *z/OS DFSMShsm Storage Administration Reference*.

Examples of ABACKUP Commands

In the example system, the ABACKUP command is first issued with the VERIFY parameter:

```
ABACKUP PAY1 VERIFY
```

Aggregate backup verification is performed in the example system aggregate group, PAY1. Remember, no data sets are actually backed up.

To invoke aggregate backup processing, the ABACKUP command is issued with the EXECUTE parameter:

```
ABACKUP PAY1 EXECUTE
```

Aggregate backup processing is performed in the example system aggregate group PAY1. The data sets are backed up.

Issuing the ABACKUP Command with the VERIFY Parameter

When you want only to verify the content of your aggregate backup without actually backing up the data sets, issue the ABACKUP command with the VERIFY parameter. This optional test run allows aggregate backup to identify any errors, and it allows you to determine if the data sets that are backed up are what you intended. Remember, no data sets are backed up.

When you specify the VERIFY parameter, aggregate backup automatically performs verification processing. Before it directs all messages to the aggregate backup activity log, verification processing checks to see if the following conditions are met:

- The aggregate group identified in the ABACKUP command is defined.
- The management class identified in the aggregate group is defined.
- The selection data sets named in the aggregate group are allocated and cataloged and meet format restrictions.
- The instruction data set, if named in the aggregate group, is allocated and cataloged.
- Selection data set syntax is correct. See “Selection Data Set Syntax” on page 291.
- All data sets in the ALLOCATE list are DASD data sets that conform to the supported data set organization types. See “ALLOCATE” on page 296.
- All data sets in the ACCOMPANY list are tape data sets.
- No data sets in any list, other than the ALLOCATE list, are catalogs.
- All fully qualified data sets are cataloged.
- The same data set name cannot appear in the INCLUDE, ACCOMPANY, or ALLOCATE lists.
- Installation exits, if requested, can be loaded.

- If RACF facility class is active and the user has restricted authority, the user's authority to process data sets is verified.

When processing completes, determine if the ABACKUP command completed successfully with a return code of zero. You may want to review the activity log for message ARC6166I for a list of the data sets selected for INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE processing. If the list of data sets is acceptable, reissue the ABACKUP command with the EXECUTE parameter so that your data sets will be backed up as described in the following section. If the list of data sets is unacceptable, modify the selection data set entries or installation exits and reissue the ABACKUP command with the VERIFY parameter until your list of data sets is acceptable.

To aid you in reviewing the results of data set filtering, you can use the FILTEROUTPUTDATASET parameter during verification processing.

If the ABACKUP command failed, as indicated by a nonzero return code, review the aggregate backup activity log for error messages. Correct any errors and reissue the ABACKUP command with the VERIFY parameter. Repeat this process until the ABACKUP command completes successfully with a return code of zero.

Issuing the ABACKUP Command with the EXECUTE Parameter

Issue the ABACKUP command with the EXECUTE parameter when you are ready to have your data sets actually backed up. Aggregate backup always performs verification processing, as described in the previous section. The difference is that after verification processing completes successfully, aggregate backup proceeds to back up your data sets.

Aggregate backup directs all messages to the aggregate backup activity log. The aggregate backup starting and ending messages are also listed in the command activity log and displayed to the user who issued the command. When processing completes, determine if the ABACKUP command completed successfully with a return code of zero. Review the activity log for messages ARC6060I and ARC6061I for the volume serial numbers of the aggregate backup control and data files, ARC6071I for the volume serial number for the instruction/activity log file, and ARC6062I for the volume serial numbers for the tapes containing ACCOMPANY data sets. You will need these volumes later when you perform aggregate recovery.

If the ABACKUP command failed, as indicated by a nonzero return code, review the activity log for error messages. Correct any errors and reissue the ABACKUP command. The command always starts processing from the beginning. Repeat this process until the ABACKUP command completes successfully with a return code of zero. If the command fails, the data on the tape is invalid.

Since aggregate backup allows backing up of data sets that have been migrated to ML2 tapes, it is possible that the tape volumes containing the data sets to be backed up reside on an ML2 tape volume that is in use by another DFSMShsm function. See "Tape Migration Level 2 Volume Contention" on page 117 for more information. If ABACKUP determines that it cannot allocate a tape volume because it is "in use", it retries the allocation up to 90 times, at 20 second intervals. This takes approximately 30 minutes.

If the preceding retry scenario has been completed and the ML2 tape is still "in-use," DFSMShsm issues message number ARC6254A to the operator by way of

a write to operator with reply (WTOR). The operator then has the option of canceling the aggregate backup or retrying the allocation retry logic.

Aggregate backup will copy multiple migrated data sets from a specific ML2 tape with a single mount, except when a data set spans to another tape.

Aggregate Backup Output

This section describes the data, control, and instruction/activity log files, the activity log created by aggregate backup, and the FSR and ABR records containing ABACKUP CPU times and accounting codes.

Data Files

The DFSMSdss data file is an output data set tape file that contains copies of the backed up data sets. They are the data sets you specified to be backed up in the selection data sets.

The format of the data file name is *outputdatasetprefix*.D.C*nn*V*nnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group.

The name of the DFSMSdss data file in the example is PAY1.D.C01V0001. It contains copies of the following data sets, based on the selection data set entries previously made:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002V00
- PAY1.PY001.CHECK

The name of the internal I/O data file in the example is PAY1.O.C01V0001. It contains copies of the following data sets, based on the selection data set entries previously made:

- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY001.LADDER
- PAY1.PY001.NAMES

The format of the data file name is *outputdatasetprefix*.O.C*nn*V*nnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group.

If you need to direct your ABACKUP internal I/O data file for a unique copy number to DASD, this can be done only if your DFSMSdss data file for the same copy number is directed to DASD.

If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:

- The aggregate backup files must exist and be properly cataloged at the recovery site.
- You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

Control File

The control file is the output data set tape file created by aggregate backup that is used later as input to an aggregate recovery. It contains control information necessary to recover the data sets, defined to the aggregate.

Note: The control file can reside on 1 to 15 volumes. If the 15-volume limit is exceeded, aggregate backup fails. If this occurs, either use tapes with a higher density to contain the control file or define multiple aggregate groups for the application so that each aggregate backup does not back up such a large amount of data. Reissue the ABACKUP command.

The format of the control file name is *outputdatasetprefix*.C.C*nn*V*nnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group.

The name of the control file in the example is PAY1.C.C01V0001. The data set PAY2.PY001.BENEFIT is identified on the control file to be allocated at the aggregate recovery site. The data set PAY2.PY001.TAXES is identified on the control file to be cataloged at the aggregate recovery site.

Instruction/Activity Log File

If you have previously specified an instruction data set name in the aggregate group definition, or if the ABACKUP activity log is directed to DASD by using SETSYS ABARSACTLOGTYPE(DASD), then the optional instruction/activity log file is written to a different tape volume than the control and data files. See "Creating an Instruction Data Set" on page 297.

The format of the instruction/activity log data set name is *outputdatasetprefix*.I.C*nn*V*nnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group. See "Defining an Aggregate Group" on page 298.

The name of the instruction/activity log file in the example is PAY1.I.C01V0001. It contains a copy of the instruction data set, PAYROLL.INSTRUCT, that was previously created.

Note: This data set would not be created unless INSTRUCTION DATASET is specified or the ABACKUP activity log is directed to DASD.

Activity Log

An activity log is created each time you issue the ABACKUP command. You might find the activity log useful because it contains messages written during aggregate backup processing. Review messages ARC6060I and ARC6061I for a list of the volume serial numbers for the control and data files, ARC6071I for the volume serial number for the instruction/activity log file, and ARC6062I for the volume serial numbers for the tapes containing the ACCOMPANY data sets. These volumes and their serial numbers are required later as input to aggregate recovery.

The format of the activity log name is:

mcvtactn.H*mcvthost*.ABACKUP.*agname*.D*yyddd*.T*hhmmss*

where

mcvtactn is the activity log high-level qualifier. (The default value is HSMACT.)

H*mcvthost* is the DFSMShsm host identification.

ABACKUP is a constant for aggregate backup.

agname is the name of the aggregate group being backed up.

D*yyddd* is the date.

T*hhmmss* is the time.

The activity log is written to either SYSOUT or a DASD data set; you determine this by using the ABARSACTLOGTYPE parameter of the SETSYS command. If you specify DASD and the allocation attempt fails, an attempt is made to allocate the activity log to SYSOUT. If the activity log is directed to DASD, it is backed up to the instruction/activity log file. See "Defining SETSYS Parameters for Aggregate Backup" on page 306 for more information on setting the activity log output type.

You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether or not you want DFSMShsm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs. The default is N, which specifies no automatic deletion.

Figure 38 on page 319 is an example of the aggregate backup activity log.

```
PAGE 0001  DFSMSHSM 1.4.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   97.238  12:24
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP PAY1, AT 12:24:33, STARTED TASK = DFHSMABR.ABAR0122
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP PAY1 WILL BE ROUTED TO SYSOUT=A
ARC6152I THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM THE ALLOCATE LIST DURING AGGREGATE
BACKUP FOR AGGREGATE GROUP PAY1
  PAY2.PY001.BENEFIT
ARC6152I THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM THE ACCOMPANY LIST DURING AGGREGATE
BACKUP FOR AGGREGATE GROUP PAY1
  PAY2.PY001.TAXES
ARC6004I 000{ ABACKUP PAGE 0001         5695-DF175  DFSMSdss         Data Set Services          97.238  12:34
ARC6004I 000{ ABACKUP ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 000{ ABACKUP  DUMP DATASET(FILTERDD(SYS00005)) -
ARC6004I 000{ ABACKUP  OUTDDNAME(SYS00004  -
ARC6004I 000{ ABACKUP                 ) OPTIMIZE(3) SPHERE -
ARC6004I 000{ ABACKUP  ALLDATA(*) -
ARC6004I 000{ ABACKUP  SHARE
ARC6004I 000{ ABACKUP ADR101I       RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ARC6004I 000{ ABACKUP ADR109I       RI01 (01), 97238 12:34:04 INITIAL SCAN OF USER CONTROL STATEMENTS
COMPLETED.
ARC6004I 000{ ABACKUP ADR006I (001)-SETUP(01), 97238 12:34:05 EXECUTION BEGINS
ARC6004I 000{ ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 000{ ABACKUP                          PAY2.PY002.SOCSEC
ARC6004I 000{ ABACKUP                          PAY2.PY002.STATUS.G0002V00
ARC6004I 000{ ABACKUP                          PAY1.INSTRUCT
ARC6004I 000{ ABACKUP                          PAY1.PY001.CHECK
ARC6004I 000{ ABACKUP                          PAY1.PY001.PGMLIB
ARC6004I 000{ ABACKUP ADR013I (001)-CLTSK(01), 97238 12:34:13 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 000{ ABACKUP ADR012I (SCH)-DSSU (01), 97238 12:34:13 DFSMSdss PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
ARC6064I DATA SET PAY1.PY001.NAMES HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6064I DATA SET PAY2.PY002.RETIRE HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6064I DATA SET PAY1.PY002.IRA HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6064I DATA SET PAY1.PY002.LADDER HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6369I STORAGE REQUIREMENTS FOR AGGREGATE GROUP PAY1 ARE: L0=1105K, ML1=98K, ML2=0, TOTAL=1203K
ARC6061I VOLUMES USED FOR CONTROL FILE PAY1.C.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  A00009
ARC6060I VOLUMES USED FOR DATA FILE PAY1.D.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  A00008
ARC6060I VOLUMES USED FOR DATA FILE PAY1.O.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  A00008
ARC6062I VOLUMES USED FOR ACCOMPANY DATA SETS DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
  A00001
ARC6051I AN INSTRUCTION DATA SET WAS NOT SPECIFIED FOR AGGREGATE GROUP PAY1
ARC6055I AGGREGATE BACKUP HAS COMPLETED FOR AGGREGATE GROUP PAY1, AT 12:36:53, RETCODE = 000
```

Figure 38. Example of an Aggregate Backup Activity Log

ABACKUP CPU Time and Accounting Codes

ABARS records the CPU time for processing ABACKUP and ARECOVER requests. This information is written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANy system data sets.

The ABACKUP CPU time and ARECOVER CPU time are also recorded in the ABR record.

The ISMF aggregate group definition panel allows you to specify an accounting code for an aggregate group. The accounting code gets written to the FSR control block (and associated SMF record where one exists) along with the ABACKUP CPU processing time for the aggregate. Installations can then use the information to assist in calculating charges to users for ABARS requests.

The accounting code is also written to the ABR record, and the ABACKUP control file. It is written to the ABACKUP control file so the code can be used at the recovery site without requiring an aggregate definition at the recovery site.

The accounting code is optional, and can be up to 32 characters in length. Valid characters are alphanumeric characters, blanks, commas, periods, parentheses, hyphens, slashes, ampersands, and apostrophes (single quotes).

DFSMShsm Coexistence Considerations for New ABACKUP Output Data Sets

In DFSMShsm, the ABACKUP output files are physical sequential data sets, whereas in previous versions of DFSMShsm these files were generation data sets. This change allows DFSMShsm to fully manage the ABACKUP output files by using the management class aggregate backup attributes.

Once your installation has converted to DFSMShsm and is processing aggregate backups, new GDSs are not created for existing ABACKUP output file GDGs. This means that GDG roll-off processing does not take place for the old ABACKUP output file GDSs, and these GDSs and their associated GDG base definitions need to be manually deleted once you have determined that they are no longer needed. You can do this by using an IDCAMS batch job with a DELETE statement for each remaining GDS and a DELETE GDG statement for each remaining GDG base.

Note: DFSMShsm will not consider old GDSs and GDG bases during roll-off processing for old aggregate backup versions and EXPIREBV ABARSVERSIONS processing.

Expiring ABR Records in the BCDS

You can use the EXPIREBV ABARSVERSIONS command to expire ABR records created during aggregate backup or recovery processing.

The ABARSVERSIONS keyword is mutually exclusive from any data set processing performed during EXPIREBV processing. ABARSVERSIONS is also mutually exclusive with the NONSMSVERSIONS parameter. The STARTKEY and ENDKEY parameters can be used with ABARSVERSIONS, as can OUTDATASET | SYSOUT. DISPLAY and EXECUTE are still required parameters and have the same processing affect. .

SETSYS BACKUP | NOBACKUP has no relevance with the ABARSVERSIONS parameter. However, a BCDS must exist or the EXPIREBV ABARSVERSIONS command fails.

When ABARSVERSIONS has been specified, the BCDS is searched and each ABR record is checked to see if the ABARS version has expired. For expired ABARS versions, the control file name is located in the catalog and a list of volume serial numbers that the control file resides on is requested. When necessary, the volumes are deleted from the RACF HSMABR tape volume list. The control file is uncataloged after the volume serial number list has been exhausted. When the file does not exist in the catalog, it is not an error condition, and the ABR record can still be deleted.

EXPIREBV must obtain the management class definition to determine which ABACKUP copies have expired. The management class definition contains the following:

- RETAIN EXTRA VERSIONS—This value is used to expire all versions except the latest version.
- RETAIN ONLY VERSION—This value is used to expire the last remaining version.

If, by using the ARCEDEXT installation exit, you have specified special tape management system expiration dates for the ABACKUP output files, the preceding two values should be set to NOLIMIT. This prevents DFSMShsm from attempting to delete ABARS versions during ABACKUP and EXPIREBV processing by using management class criteria.

Note: The preceding two values have total control over deletion of ABARS versions, unless the EXPIREBV command is issued with the RETAINVERSIONS keyword.

If the management class specified in the most current version of the ABR record for a specific aggregate group does not exist, expiration of the aggregate group being processed is skipped. If an aggregate group name has been specified on the EXPIREBV command, the command fails for this case.

If it is active, installation exit ARCTVEXT is called for each volume being processed, indicating to the exit that the volume being expired is an ABARS tape volume.

This process is repeated for the data files and the instruction/activity log files.

After the control file, data files, and instruction/activity log file have been uncataloged and removed from the HSMABR tape connected set, the ABR record is deleted and processing continues with the next ABR record available in the BCDS.

Note: It is possible to code SMS ACS routines that will direct ABARS files to DASD. If this is done, and the files are migrated to DASD, an DFSMShsm DELETE will be issued to remove them during expiration processing. If that delete fails, the file may remain cataloged with no ABR record pointing to it. If this happens, the data set should be manually deleted.

When you use EXECUTE with the ABARSVERSIONS parameter, message ARC6377I is issued to show the status of each file in the ABR record. Message ARC6376I is also issued to show whether or not the ABR record was deleted.

When you use DISPLAY with the ABARSVERSIONS parameter, a report is generated that lists information about the ABR records that would have been omitted had EXECUTE been specified. The file status shows whether the file will be uncataloged when EXECUTE is run, or whether the file does not exist in the catalog (“Not in Catalog”). If “Catalog Error” is in the file status, an error has occurred when DFSMShsm retrieved the catalog information, and the ABR record cannot be deleted.

A sample of the output obtained when the ABARSVERSIONS parameter is used with DISPLAY is shown in Figure 39 on page 322.

```
DISPLAY OF ABARS VERSIONS ELIGIBLE FOR EXPIRATION AT 08:36:24 ON 1995/02/07 FOR XA2

     COMMAND INPUT: ABARSVERSIONS

     SUBMITTER'S USERID: B999999

     AGNAME = agname1    VERSION = 0001

       AGGREGATE VERSION KEY = agname1.1995102000101   DELETED*
       (* DETERMINED ON 1995/02/07)
         AGGREGATE GROUP FILES       FILE STATUS
            prefix.C.C01V0001        Uncataloged
            prefix.D.C01V0001        Uncataloged
            prefix.O.C01V0001        Not in Catalog
            prefix.I.C01V0001        Uncataloged

       AGGREGATE VERSION KEY = agname1.1995102000102   NOT DELETED*
       (* DETERMINED ON 1995/02/07)
         AGGREGATE GROUP FILES       FILE STATUS
            prefix.C.C02V0001        Uncataloged
            prefix.D.C02V0001        Not in Catalog
            prefix.O.C02V0001        Uncataloged
            prefix.I.C02V0001        Catalog Error

     AGNAME = agname2    VERSION = 0003     EXP DATE = 1995/05/17

       AGGREGATE VERSION KEY = agname2.1995137000301   DELETED*
       (* DETERMINED ON 1995/02/07)
         AGGREGATE GROUP FILES       FILE STATUS
            prefix.C.C01V0003        Not in Catalog
            prefix.O.C01V0003        Uncataloged
     END OF DISPLAY - 00000002 ABARS VERSIONS ELIGIBLE FOR EXPIRATION
```

Figure 39. Example of DISPLAY ABARSVERSIONS Parameter Output

Related reading: For additional information about the EXPIREBV command, see the *z/OS DFSMShsm Storage Administration Reference*.

ABACKUP Considerations

If an instruction data set is specified in the aggregate group and DASD logging is specified, DFSMSdss is used to write the instruction data set and the activity log data set into one file on a tape (or tapes). To allow the control file to save the instruction and activity log volume serial number information, the instruction data set and activity log files must be written before the control file. Because of this restriction, the activity log written to tape does not contain volume serial number information for the control file, nor the completion code from ABACKUP processing.

In order to facilitate coexistence with previous releases of ABARS, where the control file is the first file sequence on the tape, the instruction and activity log files are written to a separate tape from the control file, if an instruction data set is specified in the aggregate group or DASD logging is specified.

The log is written to an output tape *only* if:

- The data file has been successfully written to tape.
- DASD logging has been specified at the aggregate backup site using the SETSYS ABARSACTLOGTYPE command, and the DASD activity log has been successfully processed.

If DASD logging is not specified during ABACKUP processing, the following processing takes place:

- A message is written to the log indicating that DASD has not been specified for the ABACKUP process.
- A blank activity log name is written to the ABR record and control file header.

The file containing the activity log data set and instruction data set has a naming convention of:

prefix.I.C*nn*V*nnnn*

where

prefix identifies the prefix specified on the Aggregate Group Define/Alter panel.

I is a constant to identify this as the file containing the instruction data set and the activity log.

C*nn* is the constant "C" followed by *nn* which identifies the copy number specified for ABACKUP.

V*nnnn* is the constant "V" followed by *nnnn* which identifies the version number.

If the ABACKUP activity log is being written to SYSOUT and an instruction data set has not been specified in the aggregate group definition, DFSMSdss is not called to DUMP the data sets (there will *not* be a tape mount request for the instruction or activity files).

After the data files are written to tape, the ABACKUP activity log (DASD only) is closed and the instruction data set and activity log are DUMPed to tape using DFSMSdss. All messages are intercepted and stored in memory during DFSMSdss processing.

After the instruction and activity log files are written to tape, the DASD ABACKUP activity log is reopened to allow the stored messages—and any additional messages—to be written to it.

Note: These additional messages are not written to the activity log file that is seen at the ARECOVER site.

The internal I/O data file for a unique copy number can be directed to DASD only if the DFSMSdss data file for the same copy number was directed to DASD, or there is no data file for this invocation of ABACKUP.

If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:

- The aggregate backup files must be properly cataloged and exist at the recovery site.
- You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

ABARS User Catalog and Catalog Alias ABACKUP Support

When DFSMShsm determines that there are user catalogs in the ALLOCATE list, it provides aggregate backup and recovery support for the catalogs and their associated ALIAS definitions.

When a user catalog is specified in the ALLOCATE list, the necessary data is obtained to define that catalog at the recovery location. The LISTCAT function also provides a list of the aliases defined to the specific catalog.

There is no limit to the number of user catalogs that the user may specify in the ALLOCATE list. If the user specifies any catalog type other than an user catalog, an error message is issued and ABACKUP verification fails.

If a catalog data set name is encountered in either the INCLUDE or ACCOMPANY lists, an error message is issued and ABACKUP verification fails.

The aggregate backup verification step lists the user catalogs specified in the ALLOCATE list and all of the aliases defined to those catalogs. This action is performed when the user specifies the ABACKUP command with the VERIFY parameter.

Note: All errors encountered when processing catalogs in the ALLOCATE list result in error messages.

Aggregate Backup Assist

Aggregate backup assist (ABA) is a stand-alone program that supports the DFSMShsm ABARS function. It supports ABARS by producing a selection data set that can be used as input to the aggregate backup function.

How Aggregate Backup Assist Works

When supplied with a list of job names associated with an aggregate group (that is, a group of applications, an application, or a subset of an application) the ABA program scans system management facilities (SMF) records that have been collected over a period of time. ABA extracts the names of data sets that have been opened by the jobs on the supplied list. ABA then produces a list of those data set names in the form of a selection data set.

Note: The input SMF records must be collected for a sufficient period of time to capture the processing of all of the applications in the aggregate group.

To shorten ABA's run time, the IFASMFDP utility program may be used before ABA is run to extract the necessary types of SMF records from an installation's SMF data set. See "Input Requirements" on page 326.

What Aggregate Backup Assist Produces

ABA produces a selection data set consisting of one ABARS INCLUDE statement specifying all of the data set names found during analysis of the SMF records. The data set names are specified in three sections delimited by block comments: cataloged DASD data sets, cataloged tape data sets, and uncataloged data sets. The data set names in each section are sorted in alphabetical order.

Recommendation: Before you issue the ABACKUP command that uses the selection data set that has been created, you should edit the selection data set to delete the names of uncataloged data sets.

Special Considerations for ABARS

ABA has been designed to assist you in identifying data sets related to an aggregate group. You need to ensure that all data sets necessary to rebuild the working environment are identified and collected during the aggregate backup process.

Note: Although the output selection data set is in a form that may be usable as input to the aggregate backup function, you need to examine closely and modify the selection data set, if necessary, before actually using it in an ABACKUP command. If you use the data set created by ABA as input to aggregate backup, the data set name must be the same as the selection data set in the aggregate group.

When verifying the contents of the ABA-generated selection data set, look for problems such as the data set:

- Not containing all necessary data set names associated with the aggregate group (data sets not allocated or not opened during the SMF interval)
- Containing data set names that are not part of the aggregate group
- Containing data set names that no longer exist (flagged as uncataloged)
- Containing tape data set names that can be moved to an ACCOMPANY list
- Containing DASD data set names that can be moved to an ALLOCATE list

The following hints assist you in determining which data sets to back up and how to collect them.

Uncataloged data sets: ABARS does not support uncataloged data sets. If you want to back up uncataloged data sets, you should manually catalog the data sets before issuing the ABACKUP command. Any uncataloged data sets you do not choose to catalog should be removed from the ABA-generated selection data set.

Nonaggregate group data sets: The generated selection data set may contain the names of data sets not in the aggregate group; for example, SYS1.LINKLIB. When these data sets are to be excluded, you must delete their names from the generated selection data set.

DASD data sets: The contents of some DASD data sets may not be required for execution of an application. Data set names that are used only for output should be moved to an ALLOCATE list.

Tape data sets: Tape data sets are candidates for the ACCOMPANY list in the selection data set. If you do not want the burden of copying the tapes, the tape data set names should be moved to the ACCOMPANY list.

Authorization and Security Requirements

For each data set eligible for the selection data set, ABA checks whether the data set is cataloged, and attempts to obtain certain catalog information. ABA should be installed in an APF-authorized library for the following reasons:

- If any of the catalogs contain password protection, ABA may cause the MVS catalog to prompt the system operator for the password. Having APF authorization allows ABA to bypass MVS catalog security checking.
- APF authorization enables ABA to query catalogs that are RACF protected and prevents ABENDs caused by RACF authorization failures.

Input Requirements

The ABA program uses two sources of input: SMF data and a list of job names associated with an aggregate group.

SMF Data

Before running ABA, you need to ensure that your system is set up to have SMF collect record types 14, 15, and 64. Furthermore, the recorded data must contain sufficient records to span the time frame wherein all desired jobs have been processed and recorded in the SMF data sets. An aggregate group may contain applications that run only monthly, quarterly, or annually, so that you need to plan in advance to collect sufficient SMF data.

The following SMF records are required for ABA:

Type 14 INPUT or RDBACK data set activity
Type 15 OUTPUT/UPDATE/INOUT/OUTIN data set activity
Type 64 VSAM component or cluster status

Aggregate Job Names

You need to identify all of the jobs that make up the applications in the aggregate group. If your installation has formal naming conventions for job names, then generic job names may be supplied. Substitute an asterisk (*) for a string, and substitute a percent sign (%) for a character. Table 20 shows acceptable forms of generic job names.

Table 20. Acceptable Generic Job Names

Specify	ABA collects data set names for:
*AB	Any job name that ends with the string AB.
A*	Any job name that starts with the letter A.
PAY	Any job name with the string PAY in it.
NET*A	Any job name that begins with NET and ends with A.
NET%%A	Any job name that begins with NET and ends with A with two characters in between. % can be used in place of any character.

- The job names, generic or not, should be one to eight characters long.
- Each job name must be specified on a separate record in the input stream.
- A job name can occur anywhere in the record between columns 1 and 80, inclusive.
- The first non-blank string in each record is assumed to be the job name, and any other information in the record is ignored.

Storage Requirements

The amount of storage required to run ABA depends on the number of input job names and generic job names specified, and the number of data set references found to be eligible for the selection data set. ABA has been designed to run in MVS/ESA environments with the 24-bit residency mode (RMODE=24) and an addressing mode of ANY (AMODE=31). This enables ABA to take advantage of storage above the 16-megabyte line. To ensure that such storage is available to ABA, especially when you are processing large applications with many jobs and data sets, you may want to specify a large region size on the EXEC statement in the JCL which executes ABA.

DD Statements Used by ABA

Table 21 shows the JCL DD statements used by ABA. If your installation specified DYNALOC=NO in the ICEMAC macro for DFSORT™, the SORTWORK DD statements need to be specified in the JCL which executes ABA. DYNALOC=NO tells DFSORT not to dynamically allocate sort work data sets.

Table 21. DD Statements Used by ABA

DD name	Purpose
SMFIN	Defines the input SMF data set.
SORTIN	LRECL=80 and RECFM=FB defines an intermediate data set to be used as input to DFSORT. A temporary data set should be used.
SELECTDS	LRECL=80 and RECFM=F or FB defines the final output selection data set where the data set names are written. It can be either a sequential data set or a member of a partitioned data set. The record length must be 80 and the record format must be fixed or fixed block.
SYSPRINT	LRECL=133 and RECFM=FBA defines a sequential message data set.
SYSOUT	Defines a sequential message data set required by DFSORT.
SORTWK01	Specifies that SYSDA space of 2 cylinders and secondary extents of 3 cylinders each can be used for workspace by DFSORT.
SYSIN	Defines the input data set containing a list of job names in the aggregate group. This data can be in the job stream.
SYSDIAG	Defines an optional diagnostic data set. The DCB parameters are RECFM=FB and LRECL=74. The BLKSIZE must be supplied by the user. The data written to this file is binary and should not be directed to spool. **Note:** Output from the SYSDIAG data set may be required by IBM Customer Support to diagnose problems with this utility. Therefore, it is highly recommended that you always include this DD statement.

Sample JCL

```
//ABA        JOB ????
         //ABARUN    EXEC PGM=ABA,REGION=20M
         //SMFIN      DD  DSN=CUSTOMER.SMF.DATA,DISP=SHR
         //SORTIN     DD  DSN=&&SORTIN,DISP=(NEW,DELETE),
         //               SPACE=(CYL,(2,3)),
         //               DCB=(LRECL=80,RECFM=FB),
         //               UNIT=SYSDA
         //SELECTDS   DD  DSN=CUSTOMER.SELECT.DS,DISP=(NEW,CATLG),
         //               DCB=(LRECL=80,RECFM=FB),
         //               SPACE=(CYL,(1,1),RLSE),UNIT=SYSDA
         //SYSPRINT   DD  SYSOUT=*
         //SYSOUT     DD  SYSOUT=*
         //SORTWK01   DD  UNIT=SYSDA,SPACE=(CYL,(2,3))
         //SYSIN      DD  *
         PAYA*
         PAYB*
         /*
         //SYSDIAG  DD  DSN=CUSTOMER.ABA.LOG,DISP=(NEW,CATLG),
         //             DCB=(RECFM=FB,LRECL=74,BLKSIZE=7400),
         //             SPACE=(CYL,(2,1),RLSE),UNIT=SYSDA
         //* ON A 3380, THE PRIMARY ALLOCATION OF 2 CYLINDERS
         //* FOR SYSDIAG WILL HANDLE THE DIAGNOSTIC OUTPUT
         //* FOR APPROXIMATELY 18,000 UNIQUE, QUALIFIED DATA
         //* SETS.
```

Sample Output

```
INCLUDE(
         /* ------------------------------------------------*/
         /* Following are cataloged nontape data sets       */
         /* ------------------------------------------------*/
             COLE.MYDATA
             CLR.PLI.LOAD
             CLR.PLI.OBJ
             NET.NETVIEW.REL2.BNJPNL1
             ORDERS.IDS
             RINEY.DFSMS.ABC
         /*         6   cataloged nontape data sets found  */
         /* ------------------------------------------------*/
         /* Following are cataloged tape data sets          */
         /* ------------------------------------------------*/
             CUSTOMER.CMSBASE.LINKLIB  /* VOLSER=TAPE01    */
             PAY0001.MONTHLY.OCT.DATA  /* VOLSER=PAYS02    */
             PAY0001.WEEKLY.W0005.DATA /* VOLSER=PAYS02    */
             SYS1.BCDS.BACKUP.V0000007 /* VOLSER=TAPEXX    */
         /*           4   cataloged tape data sets found   */
         /* ------------------------------------------------*/
         /* Following data sets were not found in the       */
         /* normal catalog search                           */
         /* ------------------------------------------------*/
             CUSTOMER.TEMPORARY.LIST
             PAY0001.MONTHLY.OCT.REPORT
             PAY0001.WEEKLY.W0005.REPORT
         /*           3   uncataloged data sets found      */
               )
```

How Aggregate Recovery Works

To perform aggregate recovery, you must have the control file, data files, and instruction/activity log file created by a previous aggregate backup. To use the simplified ARECOVER command, you must also have an ABARS activity (ABR) record at the recovery site. You can then use the ARECOVER command to recover the data sets from the tape files.

Figure 40 on page 329 shows the information flow of an aggregate recovery.

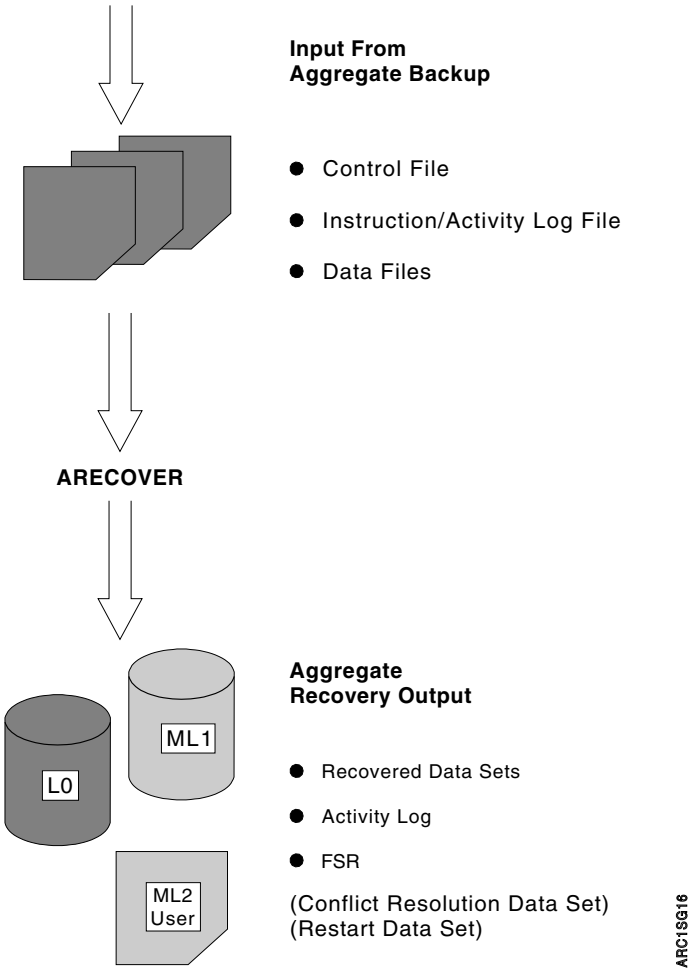


Figure 40. Aggregate Recovery Information Flow

You can choose to prepare the recovery site for the eventual recovery of the aggregate by issuing the ARECOVER command with the PREPARE parameter. You can also choose to verify the aggregate recovery without actually recovering the data sets. Then, when you are ready, you can perform the actual recovery of your data sets.

The control file and data file were created as a result of aggregate backup and are used as input for aggregate recovery.

When you issue the ARECOVER command, DFSMShsm creates:

- The recovered data sets, which are those data sets recovered from the aggregate backup tape data file. Table 22 on page 330 shows the manner in which aggregate recovery recovers, allocates, and catalogs data sets.
- An activity log, which contains messages written during aggregate recovery processing.
- The FSR, which contains the statistics for aggregate recovery processing. This includes recording the CPU time for processing ARECOVER requests. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*).
- Updates to the ABR record, which shows the CPU time for processing ARECOVER requests. The ABR shows the accumulated CPU time of each restart until the recovery of the aggregate is successful.

Aggregate recovery can also create the following:

- A conflict resolution data set that is used as input to a subsequent ARECOVER command.

  The *conflict resolution data set* contains names of data sets that the ARECOVER verification process determines to have like-named conflicts that were not resolved by the RECOVERNEWNAME parameters, the DATASETCONFLICT parameter, or the ARCCREXT installation exit. For each such data set encountered, an entry is placed into the conflict resolution data set, with a default action of BYPASS.

  You can edit the conflict resolution data set before issuing the next ARECOVER command, and indicate which conflict resolution actions are to be taken by ARECOVER verification processing for each data set entry.

  You then issue a subsequent ARECOVER command. The conflict resolution data set is read and used as input to the ARECOVER verification process.
- A restart data set is created when the recovery of data sets does not complete successfully. The restart data set contains the names of all the data sets recovered successfully. When an aggregate recovery is reissued, the restart data set ensures that data sets already processed are not recovered a second time. If you want to start aggregate recovery again from the beginning, delete the restart data set.
- Accounting code records that are written to the ABACKUP control file, FSR, and ABR. You can assign accounting codes to aggregates, which allows you to associate ABARS CPU time to specific aggregates. The codes are written to the ABACKUP control file so they can be propagated at the recovery site.

Related reading: For additional information about creating the ABR record, see "Creating an ABR Record in the BCDS" on page 338.

Recovered Aggregate Data Sets

A data set is recovered differently if it is an INCLUDE, ACCOMPANY, or ALLOCATE data set, as specified in a selection data set during aggregate backup. Table 22 summarizes how these different types of data sets are recovered.

Table 22. Aggregate Recovery Data Set Processing

Selection Data Set Keyword	Data Set Recovered	Data Set Allocated	Data Set Cataloged
INCLUDE	X	X	X
ALLOCATE		X	X
ACCOMPANY			X

Additionally, the type of device on which an INCLUDE data set is recovered is directly based on its disposition at the aggregate backup site.

Example Aggregate Recovery Site

In "Backing Up Applications" on page 286, an example aggregate backup site was used to show backup procedures. This same example is used to show how to recover applications at the example recovery site.

The following INCLUDE data sets are recovered to DASD:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002.V00
- PAY1.PY001.CHECK
- PAY1.PY002.IRA

- PAY1.PY001.BONUS
- PAY1.PY002.LADDER

The following INCLUDE data set are recovered to tape:
- PAY1.PY001.NAMES

The following ALLOCATE data set are allocated on a DASD:
- PAY2.PY001.BENEFIT

The following ACCOMPANY tape data set are cataloged:
- PAY2.PY001.TAXES

Preparing for Aggregate Recovery

You need to consider the following in order to prepare for aggregate recovery:

- Retrieving instructions or information from the instruction data set that may have been optionally created at the aggregate backup site.
- Issuing ARECOVER with the PREPARE parameter to:
  - Determine the amount of storage needed to recover aggregate groups
  - Catalog ABACKUP outfiles
  - Add volumes to RACF
- Issuing ARECOVER with the INSTRUCTION and ACTIVITY parameters
- Creating an ABR record
- Determining the aggregate recovery site constraints.
- Planning for aggregate recovery requirements for SMS.
- Defining the target volumes to be used for aggregate recovery of non-SMS data with the DEFINE ARPOOL command.

  If you have multivolume, non-VSAM, non-SMS-managed data sets, ensure that you define an adequate number to level 0 volumes for this aggregate group, using the L0VOLS subparameter of the DEFINE ARPOOL command. If no ARPOOL is defined for this aggregate group, you must have an adequate number of volumes ADDVOLed as primary volumes, or you may have allocation failures during ARECOVER processing of this aggregate group.
- Defining aggregate recovery environmental options with the SETSYS command.
- Defining a model entity to define discrete data set profiles to RACF.
- Determining if the RACF facility class is active and at least one profile is defined, so that the user authority to use the ARECOVER command is verified.
- Determining if the RACF facility class is active and if there is a REPLACE action in either the conflict resolution data set, the DSCONFLICT parameter of the ARECOVER command, or the ARCCREXT exit. If so, define the following facility class profiles and add the user to at least one of them:

  STGADMIN.ARC.ARECOVER
  STGADMIN.ARC.ARECOVER.*agname*.REPLACE
  STGADMIN.ARC.ARECOVER.*agname*.RENTGT
- Determining if the aggregate backup output has been sent by a transmit program such as the NetView™ FTP. If you are using NetView FTP, catalog the data file data sets (using the disposition parameter) and any accompanying data sets that have been transmitted. If you are using another file transfer program, ensure that the files are cataloged.

- If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:
  - The aggregate backup files must be properly cataloged and exist at the recovery site.
  - You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

Table 23 shows a checklist you may want to follow as you complete each of the steps needed to prepare for and perform aggregate recovery.

Table 23. Aggregate Recovery Checklist

___ 1.	Verify receipt of the aggregate backup tapes containing the control, data, and instruction/activity log files.
___ 2.	Issue the DEFINE ARPOOL command to specify the target volumes to be used by aggregate recovery for non-SMS-managed data.
___ 3.	Run ARECOVER with PREPARE, INSTRUCTION, and ACTIVITY parameters
___ 4.	Determine the aggregate recovery constraints. 1. Determine the DASD resources needed to perform the aggregate recovery. 2. Determine the tape unit name to be used for output data sets to be recovered. 3. Determine the tape unit name if recovering to ML2 tape.
___ 5.	Ensure that the ABARS JCL procedure has been installed in the systems JCL library (refer to the *z/OS DFSMShsm Implementation and Customization Guide*).
___ 6.	Determine the SETSYS parameters to be used: • Specify the number of ABARS address spaces • Specify activity log type, message level, and deletion processing • Indicate ABARS procname • Indicate target unit name for recovering user tape data sets • Indicate target unit name if recovering to ML2 tape • Specify space utilization for non-SMS level 0 DASD • Specify ABARS buffer usage • Specify whether the ABACKUP output tapes were stacked • Specify whether you want the ABARS FSR record written as an SMF record (SETSYS SMF) • Indicate how SMS-managed GDSs backed up from level 0 DASD should be restored. • Indicate the desired ABARSVOLCOUNT environment
___ 7.	Add the aggregate recovery procedure name defined via SETSYS ABARSPROCNAME to the RACF started task table (ICHRIN03).
___ 8.	Define the model entity to define discrete data set profiles to RACF. This step is optional.
___ 9.	Link-edit the aggregate recovery installation exits. This step is optional.
___ 10.	If the aggregate backup output has been transmitted, ensure all aggregate backup output files and any ACCOMPANY data sets are cataloged. See "Preparing for Aggregate Recovery" on page 331.
___ 11.	Use RACF facility class to authorize certain users or operators to issue the ARECOVER command.
___ 12.	Issue the ARECOVER command with the VERIFY parameter. This step is optional. • If aggregate recovery verification completes successfully with a return code of zero, continue to step 13. • If aggregate recovery verification fails, use the conflict resolution data set to assist in resolving any conflicts, and repeat this step until the recovery verification completes successfully with a return code of zero.
___ 13.	Issue the ARECOVER command with the EXECUTE parameter. • Determine if aggregate recovery completed successfully with a return code of zero. If not, correct any errors and issue the ARECOVER command with the VERIFY parameter. If verification is successful, reissue the ARECOVER command with the EXECUTE parameter. • If aggregate recovery verification fails, use the conflict resolution data set to assist in resolving any conflicts, and repeat this step until the recovery verification completes successfully with a return code of zero. • Issuing the ARECOVER command with the VOLCOUNT(ANY) parameter can help overcome some SMS allocation failures.

Authorization Checking during Aggregate Recovery

You can permit all console operators or any user, including those who are not DFSMShsm authorized, to issue the ARECOVER command by using RACF FACILITY class profile support.

If the RACF FACILITY class profile support is installed and activated, DFSMShsm checks the console operator or user RACF profiles for authorization to issue the ARECOVER command. The user ID of OPER must be defined to RACF before console operators can be given authority to issue the ARECOVER command. If the user ID OPER is given authority to the ARECOVER command, then all console operators are authorized to issue the ARECOVER command. You can also define group names to RACF, using the RACF ADDUSER command.

If the RACF FACILITY class profile is not active, only DFSMShsm-authorized users and console operators are authorized to issue the ARECOVER command.

You can define the DFSMShsm ARECOVER command to RACF by defining the following profile names:

Command	RACF Profile
ARECOVER	STGADMIN.ARC.ARECOVER
ARECOVER	STGADMIN.ARC.ARECOVER.*agname*
ARECOVER	STGADMIN.ARC.ARECOVER.*agname*.REPLACE

Specific user IDs can then be authorized to issue these commands through the use of the RACF PERMIT command.

You may restrict use of DSCONFLICT(REPLACE), REPLACE as conflict resolution data set action, or REPLACE specified by ARCCREXT, through the use of RACF facility class profile STGADMIN.ARC.ARECOVER.*agname*REPLACE.

Related reading:

- For additional information about the RACF FACILITY class and control of access to all commands, see the "Controlling Command Access with RACF FACILITY Class" on page 582.
- For additional information about the RACF FACILITY class and control of access to all commands, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Authorizing Comprehensive ARECOVER Command Authority

When users are added to the STGADMIN.ARC.ARECOVER profile they receive comprehensive command authority. A user with comprehensive command authority for the ARECOVER command is authorized to issue any ARECOVER command.

You could use the following commands to prevent or limit the authority of users to issue the ARECOVER command:

```
RDEFINE FACILITY STGADMIN.ARC.ARECOVER

PERMIT STGADMIN.ARC.ARECOVER CLASS(FACILITY) -
  ID(user ID) ACCESS(READ)
```

The PERMIT command authorizes a user to issue the ARECOVER command for all aggregate groups.

Related reading:

- For additional information about the RACF FACILITY class and control of access to all commands, see the "Controlling Command Access with RACF FACILITY Class" on page 582.
- For additional information about the RACF FACILITY class and control of access to all commands, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Authorizing Restricted ARECOVER Command Authority

When users are added to the STGADMIN.ARC.ARECOVER.agname or STGADMIN.ARC.ARECOVER.agname.REPLACE profiles, the users have restricted command authority.

You can use the following commands to prevent or limit users authority to issue the ARECOVER command with a specific aggregate group name:

```
RDEFINE FACILITY STGADMIN.ARC.ARECOVER.PAY1

PERMIT STGADMIN.ARC.ABACKUP.PAY1 CLASS(FACILITY) -
  ID(OPER) ACCESS(READ)
```

The PERMIT command gives all console operators authority to issue the ARECOVER command with a specified aggregate group name.

Related reading:

- For additional information about the RACF FACILITY class and control of access to all commands, see the "Controlling Command Access with RACF FACILITY Class" on page 582.
- For additional information about the RACF FACILITY class and control of access to all commands, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Considering Storage Management Subsystem (SMS) Requirements

Prior planning is necessary if you want to recover SMS-managed data sets in an environment comparable to that which existed at the backup site. SMS must be active to recover VSAM data sets specified in the ALLOCATE list.

SMS class names and attributes are displayed in an ARC6119I message during the aggregate recovery verification. This may help you ensure that the SMS-managed data sets are recovered as you intend.

Notes:

1. SMS status at the recovery site determines the status of the recovered data set. A data set that was SMS-managed at the backup site may not necessarily be recovered as SMS-managed and vice versa.
2. You may want to create an aggregate that includes the SMS control data sets from the aggregate backup site. This aggregate could then be restored before restoring the aggregates for application data. This would be the most efficient way to create an SMS environment at the recovery site that matches the SMS environment at the backup site.

3. When restoring an SMS-managed user catalog that was defined in the ALLOCATE statement, ABARS passes the volume serial number where the catalog resided at the backup site. If the target volume exists at the recovery site, the user catalog is allocated on that volume. If the original volume does not exist at the recovery site, the ACS routines at the recovery site direct the allocation. In the event that SMS determines the target volume to be invalid, ABARS requests SMS to relax the guaranteed space rules so that user catalogs can be directed to other volumes in a storage group associated with a guaranteed space storage class.

   Rule: If the original volume does not exist at the recovery site, you must modify your ACS routines to remove that volume from any storage group definitions associated with a guaranteed space storage class. If you do not, the recovery will fail.
4. When restoring an SMS-managed NONVSAM data set which belonged, at the time it was backed up, to a storage class with the guaranteed space attribute, ABARS first attempts to allocate the data set to the same volumes that it was backed up from. If the volumes are unavailable or no longer belong to a storage group that is associated with a guaranteed space storage class, the data set is allocated to different volumes that do exist in a storage group that is associated with a guaranteed space storage class.

Nonmigrated, SMS-managed data sets in an INCLUDE list are directed to the appropriate volumes during aggregate recovery by ACS routines. The following information is passed to ACS routines when restoring data sets backed up by DFSMSdss:

- Output volume serial numbers
- &ACSENVIR = RECOVER (environment in which ACS was invoked)
- Default data class name (if RACF installed)
- Default management class name (if RACF installed)
- Default storage class name (if RACF installed)
- Data set name
- Data set organization
- RACF owner or group (if RACF installed)
- Data set type
- Expiration date
- High-level qualifier of the data set name
- Low-level qualifier of the data set name
- Maximum size of the data set in kilobytes
- Number of qualifiers in the data set name
- Number of volumes specified by the user
- Data set record organization
- Retention period
- Size of data set in kilobytes
- Unit name

If the data sets are not directed by an ACS routine or if SMS is not active, the data sets are recovered to level 0 volumes defined with the DEFINE ARPOOL command. If no level 0 volumes are defined to an ARPOOL, aggregate recovery attempts to recover data sets to a volume with the same serial number as the volume on which the data sets resided at the backup site. In this case, if the volume serial number does not exist at the recovery site, recovery of the affected data sets will fail.

The SETSYS ABARSVOLCOUNT(NONE | ANY) command affects the way in which SMS-managed data set allocations are performed by DFSMSdss for L0 data sets dumped from primary volumes.

Determining Aggregate Recovery Site Constraints

Before performing aggregate recovery, you need to determine the DASD and tape unit resources needed to recover the control file, data files, and user tape data sets. You need an equivalent number of tapes during ARECOVER as were used during ABACKUP, assuming the same technology. You can find this information using the LIST AGGREGATE command. For the most part, this information is determined at the backup site and may have been included in the instruction data set during aggregate backup. See "Retrieving the Instruction Data Set."

The DASD space requirements depend in part on the residency of the data sets at the backup site and to which device type they will be recovered.

DASD space requirements are displayed in the ARECOVER activity log, the ABACKUP activity log (if recovered), the ABR record created during the PREPARE step, and in the FSR.

Tip: During ABACKUP processing of an aggregate group, you can use the SETSYS ABARSKIP command to skip data sets on volumes that are protected by the peer-to-peer remote copy function or by the extended remote copy function. Ensure that those volumes are online prior to issuing an ARECOVER command against that aggregate group.

During aggregate backup, user tape data sets with a block size equal to or larger than 32K bytes can be included in the selection data set. User tape data sets with block sizes up to 256K–1 bytes can also be included in the selection data set. DFSMShsm Version 2 Release 10 or a subsequent release is required to recover these data sets. In addition, your tape hardware must be capable of processing these large block sizes.

Related reading: For additional information about the LIST AGGREGATE command, see the *z/OS DFSMShsm Storage Administration Reference*.

Retrieving the Instruction Data Set

An instruction data set is optionally created for an aggregate backup. It may contain instructions or other information to help you perform that application's recovery. It is contained in the instruction data set, which is the second file on the control file tape.

The instruction data set may also include the following important information:

- Required SMS attributes
- Sample ACS routines
- Description of the application
- RACF environment
- Software requirements
- Hardware requirements
- Unique application execution instructions

To help you perform aggregate recovery, you should first retrieve the instruction data set. The instruction data set is contained in the instruction/activity log file,

outputdatasetprefix.I.C*nn*V*nnnn*. The *outputdatasetprefix* was specified in the aggregate group during aggregate backup. You can retrieve the instruction data set by issuing the ARECOVER command with the INSTRUCTION parameter.

Example Instruction Data Set

The instruction data set, PAYROLL.INSTRUCT, created at the example backup site and identified in the example aggregate group, PAY1, is shown in Figure 41.

```
THIS IS THE INSTRUCTION DATA SET FOR AGGREGATE GROUP PAY1.

PERFORMING AGGREGATE BACKUP ON PAY1 BACKS UP THE CRITICAL PAYROLL
FILES AT THE END OF A PAY PERIOD. RELEVANT FILES INCLUDE THE EMPLOYEE
ROSTER (PAY1.PY001.NAMES), AMOUNT OF EACH PAYCHECK (PAY1.PY001.CHECK),
AMOUNT FROM EACH CHECK PLACED IN IRA AND SOCIAL SECURITY DEDUCTION
(PAY1.PY002.IRA AND PAY2.PY002.SOCSEC), AND AMOUNT (IF ANY) OF CHECK
THAT WAS A BONUS IN ADDITION TO SALARY (PAY1.PY001.BONUS).
TO ENSURE PROPER EXECUTION, VERIFY THAT CORRECT RELEASE LEVELS ARE PRESENT
FOR DFSMShsm, DFSMSdss, AND DFSMSdfp AND THAT THE CORRECT SMS CLASSES ARE ACTIVE.
YOU WILL NEED THE FOLLOWING SMS CLASSES AND THEIR ASSOCIATED ATTRIBUTES
SET UP TO RECOVER THOSE FILES THAT WERE SMS-MANAGED AT ABACKUP, AND TO
ENSURE THAT THEY WILL BE SMS-MANAGED UPON RECOVERY.

DATA CLASS NAME: PHYSSEQ
DESCRIPTION: PHYSICAL SEQUENTIAL DATA SETS

 RECFM        : FB
 LRECL        : 80
 SPACE
   AVGREC     : U
   PRIMARY    : 50
   SECONDARY  : 50
 RETPD        : 175
 VOLUME COUNT : 1

STORAGE CLASS NAME: BATCHDASD
DESCRIPTION: DASD DATA SETS PROCESSED IN BATCH MODE

AVAILABILITY:
  GUARANTEED SPACE             : NO
  GUARANTEED SYNCHRONOUS WRITE : NO

MANAGEMENT CLASS NAME : BATCH
DESCRIPTION           : BATCH DATA SETS

EXPIRATION ATTRIBUTES
  EXPIRE AFTER DAYS NON-USAGE : 150
  EXPIRE AFTER DATE/DAYS      : 366

RETENTION LIMIT :
PARTIAL RELEASE : YES

MIGRATION ATTRIBUTES
  PRIMARY DAYS NONUSAGE : 7
  LEVEL 1 DAYS NONUSAGE : 45
  COMMAND/AUTO MIGRATE  : YES

GDG MANAGEMENT ATTRIBUTES
  ROLLED OFF GDS ACTION : EXPIRE

BACKUP ATTRIBUTES
  BACKUP FREQUENCY
  # OF BACKUP VERSIONS (DS EXISTS)  : 3
  # OF BACKUP VERSIONS (DS DELETED) : 1
  RETAIN DAYS ONLY BACKUP VERSION   : 366
    (DS DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS : 100
  ADMIN OR USER COMMAND BACKUP      : YES
  AUTO BACKUP                       : YES

YOU SHOULD EXECUTE ARECOVER WITH THE VERIFY OPTION TO VERIFY THE AGGREGATE RECOVERY.

PAYROLL.INSTRUCT was copied to the instruction/activity log file,
PAY1.I.C01V0001, during aggregate backup.
```

Figure 41. Example of a Retrieved Instruction Data Set

Creating an ABR Record in the BCDS

Prior to performing an ARECOVER execute or verify for a particular set of ABACKUP output tapes, issue an ARECOVER PREPARE command to create an ABR record for this recovery. This record contains an inventory of aggregate recovery activity for the aggregate version and allows you to use the simplified ARECOVER command syntax described in the *z/OS DFSMShsm Storage Administration Reference* when you actually perform aggregate recovery.

The DATASETNAME(*dsname*) parameter determines the control file data set name. You can use DSNAME as an abbreviation for DATASETNAME.

VOLUMES and XMIT are mutually exclusive parameters. VOLUMES (UNIT is optional) specifies that the volume list be used to allocate the control file to be read. XMIT specifies that the required files must already be cataloged, and the system supplies the volume list.

PREPARE, VERIFY, and EXECUTE are mutually exclusive parameters. When PREPARE is specified with the ARECOVER DATASETNAME(*dsname*) command, the ABR record is created without verification processing or data movement. When VERIFY or EXECUTE are specified, an ABR record is created if it does not exist.

Related reading: For additional information about the ARECOVER command, see the *z/OS DFSMShsm Storage Administration Reference*.

Defining ABARS Tapes to RACF

The output tape volumes generated by aggregate backup processing can be defined to RACF during ARECOVER processing. This is done when the ABR record is created by the PREPARE, VERIFY, or EXECUTE functions and when the following conditions are met:

- RACF is active at the recovery site
- You have defined TAPEVOL resource class HSMABR at the recovery site
- You have specified SETSYS TAPESECURITY(RACF | RACFINCLUDE)

Note: Tapes containing DFSMShsm output files that are recovered by DFSMShsm are also RACF-protected at the recovery site when the preceding conditions are met.

Issuing the DEFINE ARPOOL Command

The DEFINE ARPOOL command allows you to determine the target DASD level 0 and ML1 volumes to be used for aggregate recovery. These are the volumes used as target volumes for migrated and DASD data sets that have not had their allocation directed to other volumes by SMS. Data sets can have their allocations directed by SMS only when it is installed and active at the recovery site.

To ensure that the settings are defined to DFSMShsm at startup initialization, you should include the DEFINE command in the DFSMShsm parmlib member (ARCCMD*xx*). If you are an authorized DFSMShsm user, you can issue the DEFINE command before an aggregate recovery to modify existing pool configurations or add new pools.

To specify a pool of volumes to be used as target volumes for data sets recovered by aggregate recovery, use the ARPOOL*(agname)* parameter. The aggregate group

name (*agname*) identifies a pool and relates that set of volumes to an aggregate group that was previously backed up with the ABACKUP command. These volumes are used as output volumes when the application is recovered. If an aggregate group name of * is specified, the volumes are assigned to a general pool of aggregate recovery volumes. A general pool is a group of volumes defined for recovering those aggregate groups for which no specific pool has been defined. If no aggregate group name is specified, the DEFINE command fails.

Note: Even though ABARS supports a general pool of volumes for all aggregates, you should define individual aggregate recovery pools for each aggregate. Define these pools so that the order of the volumes is varied, especially for those volumes defined with the ML1VOLS parameter. For example:

```
DEFINE ARPOOL(AGG1 ML1VOLS(VOL1,VOL2,VOL3) -
                   L0VOLS(VOLA,VOLB,VOLC))

DEFINE ARPOOL(AGG2 ML1VOLS(VOL2,VOL3,VOL1) -
                   L0VOLS(VOLB,VOLC,VOLA))

DEFINE ARPOOL(AGG3 ML1VOLS(VOL3,VOL1,VOL2) -
                   L0VOLS(VOLC,VOLA,VOLB))
```

The volumes defined with the ML1VOLS parameter are serially loaded in the order in which they are defined. If several aggregate recoveries are processing with ARPOOLs that have the same volume serial numbers in the same order, then volume and channel contention are increased. Varying volume order prevents different aggregate recoveries from attempting to use the same volumes. It is apparent that this is especially critical if multiple ARECOVER commands are processing simultaneously.

ALL specifies that all currently mounted ADDVOLed level 0 and ML1 volumes are to be defined to the volume pools for aggregate recovery. It is equivalent to specifying the L0VOLS(*) and ML1VOLS(*) parameters.

ML1VOLS(*volser1,...volsern* | *) specifies a list of non-SMS-managed volumes to be used as target volumes to recover migrated data sets. If * is specified, all currently mounted ADDVOLed ML1 volumes are used as the pool of ML1 volumes for aggregate recovery.

L0VOLS(*volser1,...volsern* | *) specifies a list of non-SMS-managed volumes to be used as target volumes to recover nonmigrated, non-SMS-managed data sets. If * is specified, all currently mounted ADDVOLed primary volumes are used as the pool of level 0 volumes for aggregate recovery.

Note: If insufficient space is defined, the data set recovery fails. If the level 0 or ML1 volumes specified in the pool are not online and mounted, aggregate recovery fails.

If the DEFINE ARPOOL command is issued without the ALL, ML1VOLS, and L0VOLS parameters, the aggregate recovery pool specified by *agname* is deleted. If an aggregate group name of * is specified, the general pool of aggregate recovery volumes is deleted.

If a DEFINE ARPOOL command was not previously issued before an ARECOVER command, DFSMShsm uses all currently mounted ADDVOLed primary (59 maximum) and ML1 volumes as the ARPOOL.

The aggregate recovery pools that have been defined can be displayed by using the QUERY ARPOOL command.If L0VOLS have been specified, these volume serial numbers are passed to ACS by DFSMSdss. ACS routines may then use these volume serial numbers to determine the classes and placement of the data sets.

If ACS does not assign storage classes to the data sets, then DFSMSdss attempts to direct allocation to the volume serial number where the data set resided when backed up. If that particular volume serial number does not exist at the recovery site, recovery of the data set in question fails. In a non-SMS environment, all level 0 data sets are directed to L0VOLS volumes.

The ARPOOL volumes need not be added to DFSMShsm using the ADDVOL command before issuing the DEFINE ARPOOL command; however, the following restrictions apply to the relationship between DFSMShsm-owned and -managed volumes and volumes assigned to the aggregate recovery pool:

- A volume previously added to DFSMShsm as anything other than a PRIMARY volume cannot be defined with the L0VOLS parameter of the DEFINE ARPOOL command.
- A volume previously added to DFSMShsm as anything other than an ML1 volume cannot be defined with the ML1VOLS parameter of the DEFINE ARPOOL command.
- A volume previously defined to aggregate recovery with the ML1VOLS parameter of the DEFINE ARPOOL command can subsequently be added to DFSMShsm only as an ML1 volume.
- A volume previously defined to aggregate recovery with the L0VOLS parameter of the DEFINE ARPOOL command can subsequently be added to DFSMShsm only as a PRIMARY volume.
- A volume cannot be SMS managed.
- A volume defined as L0VOLS cannot also be defined as ML1VOLS. Conversely, a volume defined as ML1VOLS cannot also be defined as L0VOLS. This is true whether it conflicts with a previously defined pool or whether a conflict exists within the pool being defined.

You should use the ADDVOL command to add all aggregate recovery volumes to DFSMShsm as soon as possible after an ARECOVER command has successfully completed processing. If you do not do this, failures can occur when an attempt is made to refer to those data sets.

Related reading:

- For additional information about the DEFINE and QUERY commands, see the *z/OS DFSMShsm Storage Administration Reference*.
- For additional information about the DEFINE ARPOOL command, see the *z/OS DFSMShsm Storage Administration Reference*.

An Example DEFINE ARPOOL Command for Aggregate Recovery

The following DEFINE ARPOOL command is entered at the example site before performing aggregate recovery:

```
DEFINE ARPOOL(PAY1 ML1VOLS(MIG102) L0VOLS(PRIM02))
```

This command specifies that the migration volume (MIG102) and the primary volume (PRIM02) are used for the aggregate recovery of data sets identified in the example aggregate group, PAY1.

To review the aggregate recovery pools that have been defined, enter

QUERY ARPOOL

Note: Define ARPOOLs to minimize volume contention when running more than one aggregate recovery at the same time. If two or more aggregate recoveries are attempted using the same level 0 volume, one or more of the aggregate recoveries can fail because of the inability of DFSMSdss to allocate the receiving volume.

Defining SETSYS Parameters for Aggregate Recovery

The SETSYS command allows you to externally define various aggregate recovery environmental options.

To ensure that the settings are defined to DFSMShsm at startup initialization, the system programmer should include the SETSYS command in the DFSMShsm parmlib member (ARCCMD*xx*). If you are an authorized DFSMShsm user, you can issue the SETSYS command before an aggregate recovery to modify the SETSYS parameters that control aggregate processing.

To specify how many ABARS commands can run concurrently, use the MAXABARSADDRESSSPACE parameter. MAXABARSADDRESSSPACE specifies the maximum number of concurrent ABARS secondary address spaces that DFSMShsm supports (up to 64). When determining the value to specify for MAXABARSADDRESSSPACE, factor in such items as system load, resource contention, aggregate backup and recovery activity. The overriding factor is the number of tape drives available (the default for MAXABARSADDRESSSPACE is 1).

When an ARECOVER command completes, ABARS determines if there is another ABARS request queued that has not been processed. If there is, DFSMShsm reuses the address space to process that command. The address space remains active until there are no further ABARS commands queued, or the SETSYS MAXABARSADDRESSSPACE command has reduced the number of ABARS address spaces and this address space is no longer required.

The tape unit type used for the control and data file must be compatible with the tape unit type used during aggregate backup. You will specify this unit type later as part of the UNIT parameter of the ARECOVER command or in the SETSYS ABARSUNITNAME parameter.

The ABARSPROCNAME parameter specifies the procedure name used to start an ABARS secondary address space when no such address space is available for reuse. The default is DFHSMABR.

The EXITON(*exitname*) parameter lets you specify which installation exits (ARCCREXT, ARCSKEXT, ARCTVEXT) are active during aggregate recovery processing.

The OPTIMUMDASDBLOCKING parameter allows efficient utilization of DASD when ABARS performs aggregate recovery at your recovery site. When ARECOVER recovers a data set to ML1 DASD and SETSYS OPTIMUMDASDBLOCKING has been specified, ARECOVER determines the maximum block size for the data set on the ML1 DASD. This information is used

for the allocation block size of the migrated data set. The 2K records of the input migrated data set are reblocked to the optimum block size for the output data set.

If SETSYS NOOPTIMUMDASDBLOCKING is specified, DFSMShsm does not define optimum block size for the output data set.

The ARECOVERUNITNAME parameter specifies a default tape unit name for user tape data sets being recovered to a tape unit during recovery processing. If the TARGETUNIT parameter of the ARECOVER command is specified, its value takes precedence over any value specified for the ARECOVERUNITNAME parameter. However, if neither parameter is specified, a default of 3590-1 is assumed. This parameter can be overridden if SMS is active and ACS routines assign a tape storage group to the data set.

The ABARSVOLCOUNT(NONE | ANY) command allows you to affect the method of invoking DFSMSdss for restoring L0 data sets that were dumped from primary volumes. If NONE is specified, the DFSMSdss VOLCOUNT parameter is not passed to DFSMSdss during the restore. If ANY is specified, the VOLCOUNT(ANY) parameter is passed to DFSMSdss during the restore. Specifying VOLCOUNT(ANY) to DFSMSdss results in the allocation of the target data set on as many volumes as required, up to a maximum of 59 volumes.

The ARECOVERPERCENTUTILIZED(*nnn*) parameter allows you to specify to what percent DFSMShsm allows DFSMSdss to fill non-SMS DASD recovery volumes during any ARECOVER command processing. For *nnn*, substitute an integer number between 1 and 100. The default is 080 (80%).

Tip: When you issue any ARECOVER command, you can temporarily override the SETSYS ARECOVERPERCENTUTILIZED value by using the PERCENTUTILIZED parameter of the ARECOVER command.

Each time you issue the ARECOVER command, an activity log is created. You can use the SETSYS ABARSACTLOGTYPE parameter to specify whether you want the activity log automatically written to SYSOUT or to DASD. You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether or not you want DFSMShsm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs.

Stacked tapes can only be used for ARECOVER processing on hosts running DFSMS/MVS Version 1 Release 4 or subsequent releases. The default, if you don't specify the SETSYS ABARSTAPES command, is the STACK option.

When SETSYS ABARSTAPES(STACK) is specified, ARECOVER attempts to process the ABACKUP output files as if they are stacked. If the files are not stacked, ARECOVER detects the error and attempts to process the files as if ARECOVER were issued with the NOSTACK option. When SETSYS ABARSTAPES(NOSTACK) is specified, ARECOVER attempts to process the ABACKUP output files as if they are not stacked. If the files are stacked, ARECOVER detects the error and attempts to process the files as if ARECOVER were issued with the NOSTACK option.

Specifying the XMIT parameter with the ARECOVER command when using stacked output tapes has the following effect on ARECOVER processing:

- ARECOVER relies on information in the catalog to determine the file sequence numbers during the allocation of ABACKUP files. If the ABACKUP files, with the exception of the control file, are not cataloged, ARECOVER obtains the file sequence numbers from the control file.
- XMIT processing for STACKED output remains basically the same as for data that is not stacked, meaning that deallocation and allocation occur for each ABACKUP file processed. This can result in several mount requests for the same tape volume.
- If the ABACKUP output is transmitted, it is possible for the data to be received on a device type and geometry different from that on which it was originally written. Because of this, you cannot assume that the files will still be stacked when received by the network transmission software.

ABARS records the CPU time for processing ARECOVER requests, along with any applicable account code information. This information is written to the ABR record and can be obtained using the LIST AGGREGATE command. This information is written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANy system data sets. Installations can use the information to calculate charge backs for ABARS requests.

The new SETSYS ARECOVERTGTGDS(*option*) command provides greater flexibility for managing SMS-managed generation data sets that are being restored to level 0 DASD. You can specify the following options:

DEFERRED Specifies that the target data set is to be assigned the DEFERRED status.

ACTIVE Specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.

ROLLEDOFF Specifies that the target data set is to be assigned the ROLLEDOFF status.

SOURCE Specifies that the target data set is to be assigned the same status it had when it was backed up.

The default option is SOURCE, which is consistent with prior releases of DFSMShsm.

Installations are able to override the SETSYS ARECOVERTGTGDS(*option*) specification with a new ARECOVER command TGTGDS(*option*) keyword parameter. The options for ARECOVER TGTGDS are the same options as for SETSYS ARECOVERTGTGDS.

Note: If the SETSYS DEBUG parameter is specified before aggregate recovery processing, DFSMShsm does not move any data. DEBUG causes all ARECOVER commands to be processed as if the VERIFY option is specified.

Related reading:

- For additional information about the ABARSPROCNAME parameter, see the *z/OS DFSMShsm Storage Administration Reference*.
- For additional information about the ABARSVOLCOUNT(NONE | ANY) parameter, see the *z/OS DFSMSdss Storage Administration Reference*.

Example SETSYS Commands for Aggregate Recovery

The following SETSYS commands are entered at the example site before performing aggregate recovery:

```
SETSYS MAXABARSADDRESSSPACE(2)
SETSYS ARECOVERUNITNAME(3490)
SETSYS ABARSPROCNAME(SAMPLE)
```

The maximum number of secondary address spaces is set to two, which means that two ABARS commands can run at a time. If the TARGETUNIT parameter is not specified when the ARECOVER command is issued, user tape data sets are recovered to 3490 tape units as specified by the ARECOVERUNITNAME(3490) parameter. The procedure name used to start the secondary address space is SAMPLE.

To review the SETSYS parameters in effect for aggregate processing, enter:

```
QUERY ABARS
```

or

```
QUERY SETSYS
```

Defining the Model Entity

If you specify the MENTITY parameter of the ARECOVER command, you must define a model entity. The model entity is used to define discrete data set profiles for RACF protection. The model entity is applicable only for level 0 DASD and migrated data sets that have been discretely RACF-protected at the backup site.

For more information on the MENTITY parameter, see "ARECOVER Command" on page 350.

Using the Aggregate Recovery Installation Exits

Programming Interface Information

If installation exits ARCSKEXT and ARCCREXT have been previously link-edited and if the SETSYS EXITON(ARCSKEXT) and SETSYS EXITON(ARCCREXT) commands have been issued (you can use the QUERY SETSYS command to display the status of these exits), ARCSKEXT and ARCCREXT are called during aggregate recovery verification. Installation exit ARCSKEXT is called for each data set to be recovered, allowing you to specify if that data set should be skipped or included in the aggregate recovery.

Installation exit ARCCREXT is called whenever a data set naming conflict is detected. This exit allows you to specify that the data set is to be:

- Skipped—The data sets are not recovered.
- Renamed—You give the source data sets on the data file new names. Data sets must not already exist with the new names you specify.

  The rename process does not apply to migrated VSAM data sets. If you specify that a migrated VSAM data set should be renamed, it is skipped.

  Note: DFSMSdss renaming of a VSAM sphere does not cause any alternate indexes to be renamed. If a data set naming conflict exists for alternate indexes, the alternate indices are not recovered.
- Replaced—The like-named data sets at the recovery site are replaced with the corresponding aggregate backup data sets. (This is the same result as if you had specified the DSCONFLICT(REPLACE) parameter on the ARECOVER command.)

Note: When using the DSCONFLICT(REPLACE) parameter, the data set is deleted prior to attempting the recovery of the data set from the aggregate backup output files. If the recovery fails, the data set remains deleted. Causes of recovery failure include:

- I/O errors on the input tape
- Failure to allocate the new data set
- Recovering at a site that does not support the type of data set being recovered

- Renamed target—The like-named target data sets at the recovery site are renamed using the new high-level qualifier specified. This is similar to the use of the DSCONFLICT RENAMETARGET(*level*) parameter.

If installation exit ARCTVEXT has been previously link-edited and if the SETSYS EXITON(ARCTVEXT) command has been issued (you can use the QUERY SETSYS command to display the status of this exit), ARCTVEXT is called during aggregate recovery. This exit is called when rolling off old ABARS versions to indicate to tape management systems, excluding DFSMSrmm, that ABARS is expiring a tape volume.

End of Programming Interface Information

Related reading: For additional information about the ARCCREXT and ARCTVEXT installation exits, see *z/OS DFSMS Installation Exits*.

Processing Generation Data Sets

The following subsections discuss how backed-up generation data sets (GDSs) are restored during ARECOVER processing.

GDSs From Level 0 DASD in the Include List

Each generation data set (GDS) in the INCLUDE list that has been backed up from level 0 is recovered as a level 0 non-SMS-managed GDS with a status of ACTIVE if (1) ACS directs it to be recovered as not managed by SMS or if (2) SMS cannot manage the GDS because SMS is not active.

If the GDS is directed by ACS to be recovered as SMS-managed, it is recovered with the status specified by the SETSYS ARECOVERTGTGDS(*option*) command. The status can be specified as DEFERRED, ACTIVE, ROLLEDOFF, or SOURCE. SOURCE indicates that the SMS-managed data set being restored is to be assigned the same status it had when backed up. If the SETSYS ARECOVERTGTGDS command is not specified, the default is SOURCE.

If the ARECOVER command is issued with its optional TGTGDS(*option*) parameter, the value specified by the TGTGDS parameter overrides the value specified by the SETSYS ARECOVERTGTGDS command. The TGTGDS parameter, like the ARECOVERTGTGDS parameter, can be specified with the options of DEFERRED, ACTIVE, ROLLEDOFF, or SOURCE. Figure 42 on page 346 shows how level 0 GDSs are processed.

	ARECOVER			
ABACKUP	NON-SMS, NON-GDS	NON-SMS, ACTIVE GDS	SMS, Status as Specified or Default of SOURCE	SMS, NON-GDS
NON-SMS, NON-GDS	*			X
NON-SMS, ACTIVE GDS		*	X	
SMS, ACTIVE GDS		*	X	
SMS, ROLLED-OFF GDS		*	X	
SMS, DEFERRED GDS		*	X	
SMS, NON-GDS	*			X

* = Recovered as NON-SMS X = Recovered as SMS

Figure 42. Expected Results for Level 0 Data Sets in the INCLUDE List (Processed by DFDSS)

Note: Non-SMS rolled-off GDSs are not processed.

Migrated GDSs in the Include List

Non-SMS-managed GDSs in the INCLUDE list that were migrated data sets with a status of ACTIVE are always recovered as non-SMS-managed migrated GDSs, and retain a status of ACTIVE.

If SMS-managed migrated GDSs with ACTIVE status are directed by ACS to be recovered as non-SMS-managed or become non-SMS-managed because SMS is not active, they are recovered with a status of ACTIVE.

If an SMS-managed migrated GDS with a status of ACTIVE is directed by ACS to be recovered as SMS-managed, it is recovered with a status of ACTIVE.

SMS-managed migrated GDSs with a status of DEFERRED or ROLLED OFF that are directed by ACS to be recovered as SMS-managed maintain their original GDS status. If ACS directs these same data sets to be recovered as non-SMS-managed, or if they become non-SMS-managed because SMS is not active, they are recovered as uncataloged non-SMS GDSs, and DFSMShsm issues message ARC6131I informing you of this status. You can issue the following Access Method Services command to catalog the generation data sets:

```
DEFINE NONVSAM(name(dsname) VOLUMES(MIGRAT) -
  DEVICETYPES(device) NORECATALOG)
```

The generation data sets are cataloged as non-SMS-managed GDSs with a status of ACTIVE. Figure 43 on page 347 shows how migrated GDSs are processed.

	ARECOVER						
ABACKUP	NON-SMS, NON-GDS	NON-SMS, ACTIVE GDS	NON-SMS, UNCATA-LOGED	SMS, ACTIVE GDS	SMS, ROLLED-OFF GDS	SMS, DEFERRED GDS	SMS, NON-GDS
NON-SMS, NON-GDS	*						
NON-SMS, ACTIVE GDS		*					
SMS, ACTIVE GDS		*		X			
SMS, ROLLED-OFF GDS			* (ARC6131I)		X		
SMS, DEFERRED GDS			* (ARC6131I)			X	
SMS, NON-GDS	*						X

*** = Recovered as NON-SMS X = Recovered as SMS**

Figure 43. Expected Results for Migrated Data Sets in the INCLUDE List

Note: Non-SMS rolled-off GDSs are not processed.

When processing migrated data sets that are SMS-managed, ACS routines are invoked to determine if the data set should be recovered as SMS-managed or non-SMS-managed. If the ACS routines encounter a nonzero return code, they recover the migrated data sets that were SMS-managed at the backup site as non-SMS-managed data sets at the recovery site.

GDSs in the ALLOCATE List

Generation data sets in the ALLOCATE list are always recovered with a status of ACTIVE. You should not include SMS generation data sets with a status of ROLLED OFF or DEFERRED in the allocate list, because they will be recovered with a status of ACTIVE. This results in other GDSs with a status of ACTIVE being rolled off.

GDS Processing Order

The order in which the GDSs are processed determines which GDSs have an ACTIVE status after the recovery process has ended. The data sets in the INCLUDE list are processed first, and in the following order: the level 0 data sets are processed, followed by the migrated and user tape data sets. The data sets in the ALLOCATE list are processed next. The data sets in the ACCOMPANY list are processed last. If the limit is exceeded during the recovery process for a generation data group, then the oldest GDSs roll off.

Processing Migrated Data Sets in the INCLUDE List

Migrated non-SMS-managed data sets in the INCLUDE list are always recovered as non-SMS-managed data sets. You can convert the data sets (with the exception of migrated GDSs) to SMS-managed data sets by recalling them with an ACS routine that does this conversion.

Processing VSAM Data Sets in the ALLOCATE List

During aggregate recovery, VSAM data sets from the ABACKUP ALLOCATE list must be recovered as SMS-managed data sets. Therefore, SMS must be active at the recovery site, and ACS filtering must direct these data sets to SMS-managed storage.

The following verification is performed during aggregate recovery. Error messages are issued for verification failures. If EXECUTE is specified on the ARECOVER command, only those data sets not receiving verification errors are recovered.

- If SMS is not active, an error message is issued that lists the VSAM data sets from the ALLOCATE list, indicating that these data sets will not be recovered.
- If SMS is active, ACS filtering is invoked to determine if the data set can be recovered as SMS-managed. If the data set will not be SMS-managed, or SMS errors occur, messages are issued, and the indicated data set is not recovered. The following ACS variables are passed to the ACS routines:
  - &ACSENVIR = ALLOC (data set recovery environment)
  - &DSN = VSAM cluster name. If the data set was renamed during ARECOVER processing, the new name is passed.
  - &DSTYPE = PERM (permanent data set)
  - &UNIT = Unit name of ABACKUP source data set
  - &DSORG = VS (VSAM data set organization)
  - &RECORG = ES, KS, RR, LS (VSAM type - ESDS, KSDS, RRDS, LINEAR)

    If the ABACKUP source data set had associated SMS class names, these class names are passed as initial values to ACS routines. ACS may override these class names. If ACS does not override the SMS class name, and the name is not defined in the currently active configuration, an SMS ACS error occurs. The associated messages are written to the ARECOVER activity log, and verification fails.
  - &DATACLAS = source data class name
  - &MGMTCLAS = source management class name
  - &STORCLAS = source storage class name
- A data class should be defined and selected during aggregate recovery ACS filtering. This is particularly important when these data sets are allocated to a device with a different geometry than the source (that is, source=3380, target=3390). The source space allocation quantity is always recorded as an amount of cylinders or tracks, even if RECORDS was specified when the data set was defined. As a result, allocation to an unlike device may encounter over-allocation or under-allocation problems if the allocation quantity is not set by the data class.
- The storage group selected at the aggregate recovery site should contain enough volumes to support multiple volume allocation of those data sets that resided on multiple volumes at the aggregate backup site.
- Only base cluster information will be processed during ABACKUP for allocation and cataloging at the time of the ARECOVER. Users must define any alternate index (AIX) path associations at the ARECOVER site.

  Note: If an AIX/PATH, as well as base cluster allocations, are needed at the time of the ARECOVER, place the entry in the INCLUDE list where the SPHERE processing is performed. When it is placed in the INCLUDE list, data for all components of the base cluster and the associations

(AIX/PATH) are backed up and placed in the data file. This data file will be fully recovered during ARECOVER processing.

Catalog Considerations during ARECOVER Processing

There are at least three states that the catalog environment can be in at the recovery location:

1. The master and user catalogs already exist at the recovery site and have valid data set entries in them. This would be the case if you were recovering to an already functioning data center. You may choose to issue the ARECOVER command with or without the DATASETCONFLICT(REPLACE) parameter. You should use this parameter cautiously in this environment, since there is a potential of deleting existing production data sets. If you do not specify DATASETCONFLICT(REPLACE), the data sets being recovered that have name conflicts are skipped during the recovery process.
2. The Master and user catalogs already exist at the recovery site but the data set entries are invalid. This means that either the volumes associated with a data set entry are not online or the volume does not contain a valid data set entry in the VTOC. This situation usually exists when catalogs have been previously recovered as part of a full volume dump and restore of the operating system volumes containing these catalogs.

   You may choose to issue the ARECOVER command with or without specifying the DATASETCONFLICT(REPLACE) parameter. If you do not specify DATASETCONFLICT(REPLACE), the data sets being recovered that have name conflicts are skipped during the recovery process. If you do specify DATASETCONFLICT(REPLACE), ARECOVER processing detects whether the catalog entry is valid or invalid. If the catalog entry is valid, a delete is issued for data sets in conflict prior to recovering the data set from the ABACKUP tapes. If the catalog entry is invalid, a "delete noscratch" is issued for data sets in conflict.
3. You may choose to DEFINE a new catalog and associated aliases prior to issuing the ARECOVER command. This allows ARECOVER processing to catalog data set entries as data sets are being recovered. This has the advantage of avoiding any data set naming conflicts.

   The situation may also arise where there are migrated data sets in the catalog but the migration volume does not exist at the recovery site. ARECOVER processing deletes conflicting data sets when the DATASETCONFLICT(REPLACE) parameter is specified, and DFSMShsm processing handles the situation where the migration volume is not online or the VTOC entry does not exist. If an ARC0184I message is issued when attempting to retrieve the MCV record for the migration volume, issue an ADDVOL command for the migration volume and reissue the ARECOVER command.

   It is also possible that the catalog has entries for migrated data sets but the MCD record does not exist in the MCDS. ARECOVER processing detects this situation and issues a delete noscratch for the migrated data set if the DATASETCONFLICT(REPLACE) parameter was specified on the ARECOVER command. This allows the migrated data set to be deleted and aggregate recovery processing to successfully recover the migrated data set encountering this situation.
4. If you are planning to recover the user catalogs specified in the ALLOCATE statement of the selection data set by using ARECOVER, and you have recovered the master catalog from the ABACKUP site that has alias entries for

these user catalogs, you should first issue an IDCAMS EXPORT DISCONNECT command to remove this connection prior to issuing the ARECOVER command.

Performing Aggregate Recovery

Figure 44 shows what occurs when you invoke aggregate recovery by issuing the ARECOVER command.

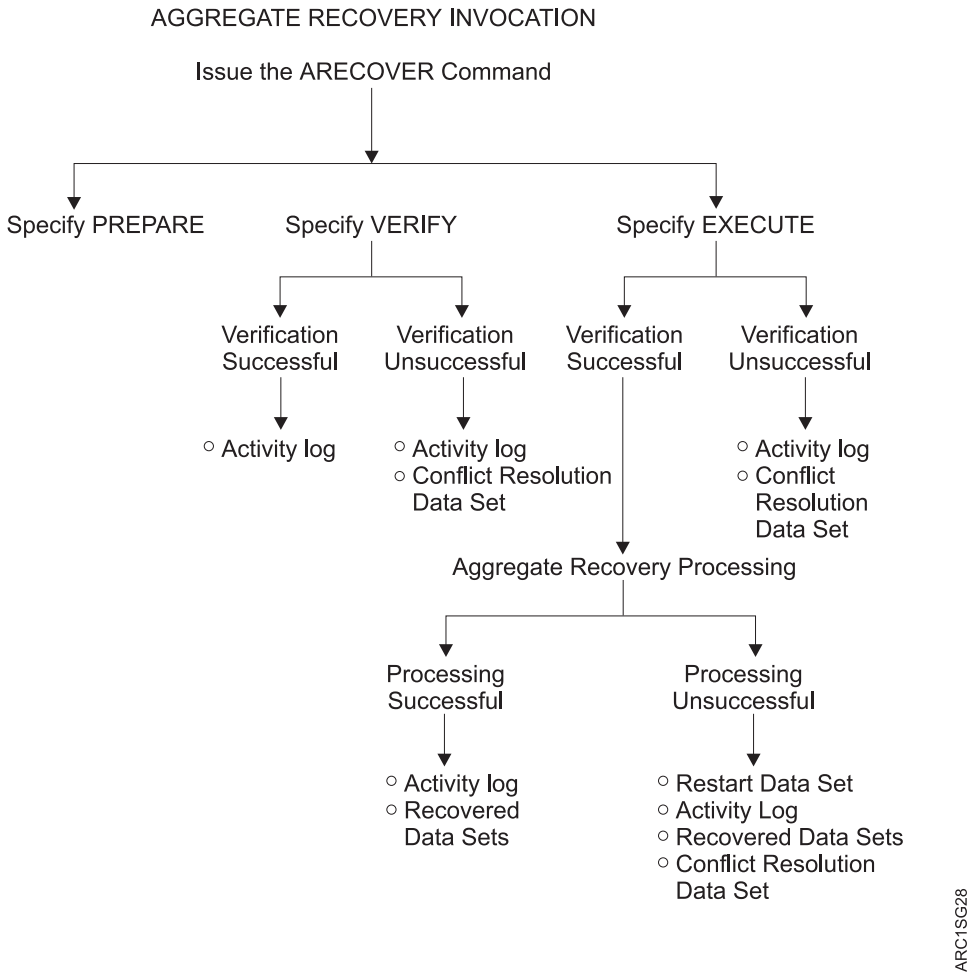


Figure 44. Aggregate Recovery Invocation

The following sections describe the aggregate recovery invocation in more detail.

ARECOVER Command

Issue the ARECOVER command when you are ready to perform aggregate recovery.

You can specify the ARECOVER command with the PREPARE parameter to create the ABR record. You can specify the ARECOVER command with the VERIFY parameter to verify the accuracy of an aggregate recovery without actually having the data sets recovered. This is like performing a test run of aggregate recovery. After aggregate recovery verification processing completes, the ARECOVER command stops. No data sets are recovered. When you are ready to have

aggregate recovery recover your data sets, you can issue the ARECOVER command and specify the EXECUTE parameter. After successful verification processing, your data sets are recovered.

Note: If SETSYS DEBUG is in effect, the VERIFY parameter is forced, regardless of other settings.

The following two sections describe in more detail what occurs when you issue the ARECOVER command with the EXECUTE parameter.

The DATASETNAME and VOLUMES | XMIT parameters are required if the ABR record was not previously created using the PREPARE parameter of the ARECOVER command. The AGGREGATE parameter is required if the DATASETNAME parameter is not used. Either the PREPARE, EXECUTE, or VERIFY parameters must be specified on the ARECOVER command.

For *dsname* in DATASETNAME(*dsname*), specify the data set name of the control file.

For *volser1,...volsern* in VOLUMES(*volser1,...volsern*), specify the serial numbers of the control files. The volume serial numbers must be specified in the same order as in message ARC6061I, issued during the aggregate backup.

When files are transferred by using a transmission program such as NetView FTP, issue the ARECOVER command using the XMIT parameter. This tells DFSMShsm that the aggregate backup output has been transmitted. Do *not* use the VOLUMES and UNIT parameters on the ARECOVER command when XMIT is specified.

Note: Specify the XMIT parameter with the ARECOVER command only if all aggregate backup output files have been cataloged. For DFSMShsm hosts that are running DFSMS Version 2 Release 10 or subsequent releases, only the control file needs to be cataloged prior to ARECOVER, as long as you are recovering the original ABACKUP files and as long as all of these files have been backed up to tape.

If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:

- The aggregate backup files must be properly cataloged and exist at the recovery site.
- You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

For *agname* in AGGREGATE(*agname*), specify the aggregate group name. Do *not* use the AGGREGATE parameter if you use the DATASETNAME parameter.

The EXECUTE parameter allows the actual ARECOVER data movement to take place when the ARECOVER command is processed. Before you can use ARECOVER AGGREGATE, one of the following actions must occur:

- You must be performing aggregate recovery on the same system where ABACKUP has been issued, and where the ABR record still exists, or
- You must have previously issued the ARECOVER DATASETNAME(*dsname*) PREPARE command.

Specify the PREPARE parameter to build the ABARS CDS record (ABR record) at the recovery site if it does not already exist, without further verification and without data movement. The PREPARE parameter also catalogs ABACKUP output files and adds volumes to RACF, if required.

The optional ACTIVITY parameter specifies that the ABACKUP activity log is to be recovered.

The optional INSTRUCTION parameter specifies that the instruction data set is to be recovered.

When the ARECOVER command is issued with the VERIFY or PREPARE parameters, DFSMShsm retrieves this information from the ABR record and issues messages during recovery processing indicating the amount of storage required to recover an aggregate group. The amount of storage required for level 0, ML1, and ML2 recovery is displayed separately, along with the combined total storage requirement. This information is maintained in the ABR record and is written to both the ARECOVER activity log and the FSR record.

Notes:

1. If the aggregate recovery is being made from DFHSM V2.5.0 or V2.6.0 ABACKUP output tapes, no storage requirements are displayed.
2. When ARECOVER EXECUTE is issued, the storage requirements of data sets not recovered is written in the ARECOVER activity log. The amounts of level 0, ML1, and ML2 storage are kept separately, as well as the combined total storage requirement.
3. When ARECOVER VERIFY is issued, and DFSMShsm determines that a RESTART data set exists, a message is written in the ARECOVER activity log indicating the storage requirements of data sets not successfully recovered in the previous pass. The amounts of level 0, ML1, and ML2 storage are kept separately, as well as the combined total storage requirement.

The optional, and mutually exclusive, STACK and NOSTACK parameters are only valid with the DATASETNAME parameter. They allow you to override the specified SETSYS ABARSTAPES value, or its default value of STACK. The option you select indicates to ARECOVER DATASETNAME processing whether or not the ABACKUP output you are going to recover is stacked. This information is needed prior to mounting the control file tape. When the control file tape is mounted, ARECOVER forward spaces to file sequence number 4 if the output is stacked, but does not if the output is not stacked.

ARECOVER can only process stacked output files on hosts running DFSMS/MVS Version 1 Release 4, or subsequent releases.

The optional UNIT(*unittype*) parameter is only valid if the DATASETNAME and VOLUMES parameters are used. For *unittype*, specify the unit type to be allocated for the control file.

The DATE or VERSION optional parameters are valid only when the AGGREGATE parameter is used. For DATE(*yyyy/mm/dd*), specify the date of the aggregate group version to be recovered. *yyyy* is the 4-digit year, *mm* is the month (01–12), and *dd* is the day of the month (01–31). If more than one aggregate version has been created on the date you specify, the most current version is selected.

For *nnnn* in VERSION(*nnnn*), specify the specific version of the aggregate group to be recovered. If neither DATE nor VERSION is specified, the most current aggregate group version is recovered.

Use the optional DATASETCONFLICT parameter to specify how you want to handle data set name conflicts.

- The RENAMESOURCE(*level*) subparameter specifies that the ABACKUP data set be recovered with the new high-level qualifier specified by *level*.
- The RENAMETARGET(*level*) subparameter specifies that the data set at the recovery site be renamed with the high-level qualifier specified by *level*, and the ABACKUP data set be recovered with its original name.

  RENAMETARGET(*level*) performs an ALTER NEWNAME for the existing data set name.Special care should be taken when you change the high-level qualifier so that the existing data set does not become uncataloged, which may occur when no catalog alias exists for the new high-level qualifier.
- The BYPASS subparameter specifies that the ABACKUP data set be skipped and not recovered.
- The REPLACE subparameter specifies that the data set at the recovery site be deleted and replaced by the ABACKUP data set.

  Notes:

  1. When you use the DSCONFLICT(REPLACE) parameter, the data set is deleted prior to attempting the recovery of the data set from the aggregate backup output files. If the recovery fails, the data set remains deleted. Recovery failure results from the following causes:
     - I/O errors on the input tape
     - Failure to allocate the new data set
     - Recovering at a site that does not support the type of data set being recovered
  2. The DATASETCONFLICT parameter may be specified in conjunction with the RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL parameters.
  3. The DATASETCONFLICT parameter may be abbreviated DSCONFLICT, RENAMESOURCE may be abbreviated RENSRC, and RENAMETARGET may be abbreviated RENTGT.

If you want to specify a predefined model entity to be used for defining discrete profiles to RACF, use the optional MENTITY parameter. Only those data sets previously protected by discrete profiles and defined to RACF before aggregate backup are protected by using discrete profiles. If the model entity is specified in the ARECOVER command and is not predefined, aggregate recovery fails.

Attention: If MENTITY is not specified and data sets exist that have been discretely RACF-protected at the backup site, the data sets are recovered as RACF indicated, and a default discrete profile is created. The profile owner is set to the high-level qualifier of the data set, and UACC is set to NONE.

If a RACF profile has been predefined for the data sets, they are recovered with this predefined profile.

Use the optional MIGRATEDDATA parameter to specify the migration level to be used when recovering migrated data sets:

- ML1 specifies that all migrated data sets are recovered to migration level 1 DASD (this is the default if you do not use MIGRATEDDATA).

- ML2 specifies that all migrated data sets are recovered to nonspecific tape volumes which are then ADDVOLed as ML2 volumes.

  Note: You must specify the tape unit name to be used when recovering migrated data sets to nonspecific volumes by use of the ARECOVERML2UNIT parameter of the SETSYS command.
- SOURCELEVEL specifies that migrated data sets that have been ML1 at the ABACKUP site are recovered to ML1 DASD; migrated data sets that have been ML2 DASD or ML2 tape are recovered to nonspecific volumes which are then ADDVOLed as ML2 volumes.

  Note: If the duplex tape option is specified for ML2 tapes, a TCN record is created to cause an automatic TAPECOPY during secondary space management. ARECOVER of ML2 data is not duplexed.

DFSMShsm automatically creates MCL records for all data sets that are successfully recovered to ML1 DASD volumes unless directed otherwise. This causes these data sets to be backed up during automatic backup processing.

Note: MCL records are not created for data sets recovered to ML2 tape volumes.

The optional NOBACKUPMIGRATED parameter allows you to specify that recovered migrated data sets are *not* to be backed up during automatic backup. If you do not specify NOBACKUPMIGRATED, DFSMShsm backs up recovered migrated data sets residing on ML1 DASD the next time automatic backup is run.

The optional PERCENTUTILIZED(*nnn*) parameter allows you to change the percentage to which DFSMShsm allows DFSMSdss to fill non-SMS DASD recovery volumes when recovering data sets to level 0 volumes. For *nnn*, substitute an integer number between 1 and 100. If SETSYS ARECOVERPERCENTUTILIZED has not been specified, the default value is 080 (80%). The value that you specify overrides any default value specified by the SETSYS ARECOVERPERCENTUTILIZED parameter.

The optional RECOVERNEWNAMEALL(*level*) parameter specifies that all data sets being recovered be renamed with the high-level qualifier specified by *level*.

The optional RECOVERNEWNAMELEVEL(*olevel1,nlevel1,...*) parameter specifies that all data sets that have a high-level qualifier *olevelx* are recovered with a new high-level qualifier *nlevelx*. The maximum number of *olevelx,nlevelx* qualifier pairs is 30.

The RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL parameters allow data sets to be renamed before verification checking for like-named conflicts. These parameters apply to data sets in the INCLUDE and ALLOCATE lists (except for user catalogs in the ALLOCATE list), but do not apply to ACCOMPANY data sets. In addition, migrated source VSAM data sets may not be renamed. If the rename criteria applies to a migrated source VSAM data set, a message is issued, and the data set is not renamed.

If the rename criteria applies to a nonmigrated source VSAM data set, the new component names are not generated by ARECOVER processing, but are generated by a) DFDSS if the data set is from the INCLUDE list, or b) SMS VSAM allocation if the data set is from the ALLOCATE list. As a result, conflict resolution checking does not check for new conflicts for individual components, since the new component names are unknown.

If RECOVERNEWNAMEALL is specified, a message is issued indicating that all data sets will be renamed with the specified level. If RECOVERNEWNAMELEVEL is specified, a message is issued listing the level pairs specified, and indicating that each data set matching the old level will be renamed to the corresponding new level. Each data set meeting the rename criteria is recovered with the new name during data movement. Like-named conflicts created by the new name are evaluated during the conflict resolution process.

If renaming a data set creates a data set name with more than 44 characters, the data set name is truncated to 44 characters, and if the last character is now a '.', it is also truncated. A message is issued indicating that truncation occurred and the resulting new name.

Programming Interface Information

Whether or not RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL is specified, if the ARECOVER installation exit ARCSKEXT (which allows any data set to be skipped) is active, it is called, prior to any conflict resolution, with the original data set name of each data set to be recovered. If a data set has been renamed by the RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL parameter, but was not selected to be skipped in ARCSKEXT by the original data set name, ARCSKEXT is called again with the new data set name. This allows a data set to be skipped based on either its original or new name.

End of Programming Interface Information

The optional TARGETUNIT parameter is used to specify the unit type to be allocated to user tape data sets being recovered to a tape device. If TARGETUNIT is not specified, the unit type defaults to the current value of the SETSYS ARECOVERUNITNAME parameter.

ACS routines can override this parameter by assigning a tape storage group. If this is the case, the unit name specified by these parameters is ignored.

When you use the optional TGTGDS(*option*) parameter with the ARECOVER command, it overrides the value specified by the SETSYS ARECOVERTGTGDS command. The TGTGDS parameter allows you to specify DEFERRED, ACTIVE, ROLLEDOFF, or SOURCE for the SMS-managed generation data set that you are restoring to level 0 DASD. SOURCE causes the data set to be assigned the status it had at the time it was backed up.

You may find it helpful to use the following DFSMShsm commands to manage and query an ARECOVER command:

- HOLD ARECOVER prevents all currently queued ARECOVER commands from being processed.
- RELEASE ARECOVER releases queued ARECOVER commands that have been held by using the HOLD command or by DFSMShsm.
- STOP causes an orderly shutdown of DFSMShsm after any currently running ABACKUP and ARECOVER commands complete processing.
- STOP DUMP causes an immediate shutdown of DFSMShsm. Any ABACKUP and ARECOVER commands in process are ended and the secondary address space is released.
- QUERY or HQUERY displays information relevant to aggregate recovery.
- CANCEL or HCANCEL cancels any existing queued aggregate recovery.

- LIST displays information from existing ABR records.

Related reading:

- For additional information about the ABR record, see "Creating an ABR Record in the BCDS" on page 338.
- For additional information about the ABARS activity log, see "Backing Up and Recovering the ABARS Activity Log" on page 360.
- For additional information about the ABARS Instruction data set, see "Recovering the ABARS Instruction Data Set" on page 359.
- For additional information about the ALTER NEWNAME function, see *z/OS DFSMS Access Method Services for Catalogs*.
- For additional information about conflict resolution, see "ARECOVER Data Set Name Conflict Resolution" on page 361.
- Refer to the *z/OS DFSMShsm Storage Administration Reference* for a detailed description of the ARECOVER command syntax and for specific examples of how to code the ARECOVER command.
- For additional information about the RACF profile, see the *z/OS Security Server RACF Command Language Reference*.
- For additional information about ABARS functions, see "Holding and Releasing ABARS Functions" on page 369.
- For additional information about the commands you can use to manage and query an ARECOVER command, see the *z/OS DFSMShsm Storage Administration Reference*.

Examples of ARECOVER Commands

For the example site, the ARECOVER command is first issued with the PREPARE parameter:

```
ARECOVER DATASETNAME(PAY1.C.C01V0001) VOLUMES(TAPE23) -
  UNIT(3490) PREPARE
```

Using the example control file, PAY1.C.C01V0001, aggregate recovery prepare is performed. The control file volume, TAPE23, is mounted on a 3490 tape unit as specified by the UNIT(3490) parameter. This action prepares the recovery site for subsequent ARECOVER processing using the aggregate group name.

The ARECOVER command is then issued with the VERIFY parameter to verify the recovery environment:

```
ARECOVER AGGREGATE(PAY1) VERIFY
```

The ARECOVER command is then issued with the EXECUTE parameter to invoke aggregate recovery processing:

```
ARECOVER AGGREGATE(PAY1) EXECUTE
```

Aggregate recovery processing is performed using the example site control file, PAY1.C.C01V0001, with a 3490 tape unit used for the control file volume, TAPE23. User tape data sets are recovered to the 3490 tape unit device type because the SETSYS ARECOVERUNITNAME(3490) command was issued previously.

The following INCLUDE data sets are recovered to DASD:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002V00

- PAY1.PY001.CHECK
- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY002.LADDER

The following INCLUDE data set is recovered to tape:
- PAY1.PY001.NAMES

The following ALLOCATE data set is allocated on a DASD:
- PAY2.PY001.BENEFIT

The following ACCOMPANY tape data set is cataloged:
- PAY2.PY001.TAXES

If the following command had been issued:

```
ARECOVER AGGREGATE(PAY1) TARGETUNIT(3400-6) EXECUTE
```

Data set PAY1.PY001.NAMES would have been recovered to a 3420 tape device. The ARECOVER TARGETUNIT parameter takes precedence over the SETSYS ARECOVERUNITNAME parameter.

If data sets were transmitted by using a transmit program such as NetView FTP, issue the following command:

```
ARECOVER DATASETNAME(PAY1.C.C01V0001) XMIT -
  TARGETUNIT(3400-6) EXECUTE
```

When the aggregate backup output files are transmitted, you must catalog the control and data files before issuing the ARECOVER command with the XMIT parameter. Transmission programs such as NetView FTP provide the option to have transmitted data sets automatically cataloged at the recovery site.

Issuing the ARECOVER Command with the PREPARE Parameter

To create an ABR record without the overhead of verification processing or data movement associated with the VERIFY or EXECUTE parameter, issue the ARECOVER command with the PREPARE parameter.

If you specify the PREPARE parameter and an ABR record already exists for the specified aggregate group version, and all the associated ARECOVER input files are properly cataloged, a message is issued, and the PREPARE function ends. If an ABR record already exists but one or more associated files are not cataloged, a message is issued for each file that is not cataloged, the ABR record is deleted (with an accompanying message), and the PREPARE function continues.

Once an ABR record is created, the control file, data files, and instruction/activity log files associated with the aggregate group version are cataloged, if previously uncataloged. The volumes containing these files are added to the RACF HSMABR TAPEVOL set, if necessary. Messages are issued indicating the catalog and RACF action taken.

Storage requirement messages are also written to the ARECOVER activity log during the PREPARE function.

Issuing the ARECOVER Command with the VERIFY Parameter

When you want only to verify the accuracy of your aggregate recovery without actually recovering the data sets, issue the ARECOVER command with the VERIFY parameter. This optional test run allows aggregate recovery to determine if any errors or data set naming conflicts exist. In the event of unresolved data set naming conflicts, aggregate recovery fails and a conflict resolution data set is created to help you resolve the conflicts. Remember, no data sets are recovered.

When performing verification processing, aggregate recovery calls installation exits ARCSKEXT and ARCCREXT, if requested and successfully loaded. ARCSKEXT allows you to specify if a data set should be skipped so that it is not included in the aggregate recovery. ARCCREXT allows you to specify how a data set naming conflict should be resolved.

If the data set naming conflicts are not resolved, the verification fails and a conflict resolution data set is created.

When processing completes, make sure that the ARECOVER command completed successfully with a return code of zero.

Related reading:

- For additional information about the ARCSKEXT and ARCCREXT installation exits, see "Using the Aggregate Recovery Installation Exits" on page 344.
- For additional information about the ARCSKEXT and ARCCREXT installation exits, see *z/OS DFSMS Installation Exits*.

Issuing the ARECOVER Command with the EXECUTE Parameter

When you are ready to have your data sets actually recovered, issue the ARECOVER command with the EXECUTE parameter. Aggregate recovery automatically performs preparation processing, if required, and verification processing, as described in "Issuing the ARECOVER Command with the VERIFY Parameter." The difference is that after verification processing completes successfully or has encountered *only* data set name conflicts, then aggregate recovery proceeds to recover your data sets. Aggregate recovery directs all messages to the aggregate recovery activity log.

When processing completes, determine if the ARECOVER command completed successfully with a return code of zero.

If the ARECOVER command has failed, as indicated by a nonzero return code, review the activity log for error messages. Correct any errors, and reissue the ARECOVER command. Repeat this process until the ARECOVER command completes successfully with a return code of zero. The restart data set, created when the ARECOVER command fails, is used when you restart the aggregate recovery. The restart data set ensures that data sets already recovered before the failure will not cause a data set conflict when you reissue the ARECOVER command.

Related reading: For additional information about the files created by an aggregate recovery, see "Aggregate Recovery Output" on page 372.

Restart Data Set

A restart data set is created when the ARECOVER command specified with the EXECUTE parameter does not complete successfully. The restart data set contains the names of all data sets successfully recovered. When you reissue the ARECOVER command, the restart data set ensures that aggregate recovery does not attempt to recover these previously recovered data sets. In this way, the restart data set avoids the possibility of numerous data set naming conflicts when you reissue an ARECOVER command that previously did not complete successfully. The format of the restart data set name is:

authid.agname.RESTART.D*yyddd*.T*hhmmss*

Where

authid is the name of the DFSMShsm-authorized user ID.

agname is the name of the aggregate group being processed. RESTART is a constant.

D*yyddd* is the date that the specific aggregate backup for this aggregate recovery was performed.

T*hhmmss* is the time that the specific aggregate backup for this aggregate recovery was performed.

Tip: The error that caused the aggregate recovery to fail may have prevented the restart data set from being closed and cataloged. If this happens, you must manually close and catalog the restart data set so that it is available to aggregate recovery when it is restarted. Message ARC6111I in the activity log identifies the restart data set name.

Recovering the ABARS Instruction Data Set

You can use the INSTRUCTION parameter instead of a DFSMSdss batch job to recover the ABARS instruction data set.

ABACKUP Considerations

If an instruction data set is specified in the aggregate group, DFSMSdss is used to write the instruction data set and the activity log data set (if DASD logging is specified), onto one file on a tape (or tapes). To save the volume serial numbers of this file in the control file, the instruction and activity log files are written before the control file. For coexistence with prior releases and for usability, the control file is the first file on the tape. Because of this necessity, the instruction and activity log files are written on a separate tape from the control file.

If an instruction data set is not specified in the aggregate group, the instruction data set name is left blank in the ABR record. A “DUMMY” instruction/activity log file is not created.

ARECOVER Considerations

Specifying INSTRUCTION with the EXECUTE, VERIFY, or PREPARE parameters causes recovery of the instruction data set, if it exists. If the instruction data set does not exist when INSTRUCTION is specified, an informational message is issued and ARECOVER processing continues.

When the INSTRUCTION parameter is specified and DFSMShsm ascertains that an instruction data set has been backed up, the following instruction and activity log file information is determined:

- If the control file indicates this backup was performed by an ABARS version previous to DFSMS/MVS Version 1, the instruction data set is not restored, a message is issued, and processing continues.
- If the DATASETNAME and VOLUMES parameters are used with the ARECOVER command, the control file is read to determine which volume serial numbers the instruction/activity log file resides on.
- If the DATASETNAME and XMIT parameters are used with the ARECOVER command, a check is made to ensure that the instruction/activity log file is cataloged.
- If the parameter AGGREGATE is used with the ARECOVER command, a check is made to ensure that the instruction/activity log file is cataloged.

If the instruction data set exists on an output tape, DFSMSdss is invoked to perform the RESTORE of the instruction data set onto a level 0 volume from the ARPOOL. If the instruction data set currently exists, it is replaced.

If the recovery of the instruction data set fails, or if the INSTRUCTION parameter is specified and there is no instruction data set on the output tape, a message is issued and processing continues.

Backing Up and Recovering the ABARS Activity Log

Since there is useful information in the ABACKUP activity log that can be used at the aggregate recovery site, ABARS includes the ABACKUP activity log data set with the instruction data set as a new file on a separate output tape.

The SETSYS ABARSACTLOGTYPE parameter

The SETSYS command ABARSACTLOGTYPE ([SYSOUT[(*class*)] | DASD]) is provided to select the ABARS activity log output data set type. If SETSYS ABARSACTLOGTYPE is not issued, it defaults to the type specified in the SETSYS ACTLOGTYPE command.

- SYSOUT(*class*) specifies that messages are written to a spool data set. For *class*, substitute an alphanumeric character for the class ABARS uses for output. If *class* is not specified, the default is the SETSYS ACTLOGTYPE SYSOUT class. If ACTLOGTYPE has been set to DASD, the ABARSACTLOGTYPE class defaults to the SETSYS SYSOUT class, which in turn defaults to “A”.
- DASD specifies that an output data set is dynamically created and cataloged.
- SYSOUT is the initial default setting. If SETSYS ABARSACTLOGTYPE has not been issued, each time SETSYS ACTLOGTYPE is issued it makes the corresponding change to the setting of ABARSACTLOGTYPE.

  Note: Once SETSYS ABARSACTLOGTYPE is issued, any subsequent SETSYS ACTLOGTYPE command does not affect the ABARSACTLOGTYPE.

The DFSMShsm QUERY command, with the SETSYS or ABARS parameter specified, can determine and display the value specified in the ABARSACTLOGTYPE(*type*) parameter.

Conflict Avoidance and Conflict Resolution Processing Options

The RECOVERNEWNAMELEVEL(*olevel*,*nlevel*) and RECOVERNEWNAMEALL(*level*) parameters provide options that allow you to avoid known data set name conflicts when you issue the ARECOVER command. Specifying these parameters, however, does not guarantee that a data set name conflict for a newly generated name will not occur. If the use of either of these parameters results in a data set name conflict, then conflict resolution processing that involves the newly generated name is invoked as shown in the following table.

If you specify	And you have also specified	Then
RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL	DATASETCONFLICT(RENAMETARGET(*level*))	The data set that exists at the recovery site is renamed by using the high-level qualifier specified in the RENAMETARGET(*level*) parameter, and the ABACKUP data set is recovered using the newly generated name as specified by the RECOVERNEWNAME parameter.
	DATASETCONFLICT(RENAMESOURCE(*level*))	The data set is recovered with the data set name generated by using the high-level qualifier specified with the RENAMESOURCE(*level*) parameter. In this case, the high-level qualifier specified with the RENAMESOURCE parameter will override the high-level qualifier of the data set name that was originally backed up, along with the high-level qualifier generated by the RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL parameters.
	DATASETCONFLICT(REPLACE)	The existing data set at the recovery site is deleted, and the data set is recovered with a data set name that uses the high-level qualifier specified with the RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL parameters.

ARECOVER Data Set Name Conflict Resolution

DATASETCONFLICT Parameter

The optional DATASETCONFLICT parameter of the ARECOVER command allows more flexibility for conflict resolution options. This parameter applies to INCLUDE, ALLOCATE, and ACCOMPANY data sets, with some restrictions. Like-named conflicts are evaluated based on the new name for any data sets renamed by RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL.

The following subparameters of DATASETCONFLICT specify the action to be taken when a like-named conflict is found during ARECOVER verification.

- If you specify RENAMESOURCE(*level*), the ABACKUP data set is recovered with the new high-level qualifier specified by *level*.
- If you specify RENAMETARGET(*level*), the existing data set at the recovery site is renamed with the specified high-level qualifier, and the ABACKUP data set is recovered with its original name.

  RENAMETARGET(*level*) performs an ALTER NEWNAME for the existing data set name. Special care should be taken when you change the high-level qualifier

so that the existing data set does not become uncataloged, which may occur when no catalog alias exists for the new high-level qualifier.
- If you specify REPLACE, the existing data set at the ARECOVER site is deleted, and the ABACKUP data set is recovered.
- If you specify BYPASS, the ABACKUP data set is skipped and not recovered.
- BYPASS
- RENAMETARGET (if target data set restrictions are met)
- RENAMESOURCE (not supported for ACCOMPANY data sets)

If one or more verification errors occur (and are not resolved), the data sets receiving the verification errors are not recovered, but (when EXECUTE has been specified) processing continues. This causes a RESTART data set to be created. Data sets that did not receive verification errors are recovered.

Related reading: For additional information about the ALTER NEWNAME function, see *z/OS DFSMS Access Method Services for Catalogs*.

Conflict Resolution Data Set

The use of the IDCAMS command data set is no longer valid. A conflict resolution data set is generated by ARECOVER to simplify the resolution of like-named conflicts discovered during ARECOVER verification.

Each time ARECOVER verification processing cannot resolve a like-named conflict for a data set, the data set name is written to the conflict resolution data set with a default action of BYPASS. A message is issued indicating the name of the conflict resolution data set that was created. The name is generated as follows:

id.*agname*.CONFLICT.D*yyddd*.T*hhmmss*

Where:

id is the DFSMShsm authorized user ID.
agname is the name of the aggregate group being recovered.
CONFLICT is a constant.
D is a constant, followed by *yyddd*, indicating the Julian date the ABACKUP was performed.
T is a constant, followed by *hhmmss*, indicating the time the ABACKUP was performed.

The aggregate group name, ABACKUP date, and ABACKUP time are used to make this conflict resolution data set unique to the recovery of a specific ABACKUP version.

An example of a conflict resolution data set is shown in Figure 45 on page 363. It is a physical sequential data set with fixed-length 80 byte records.

```
/* CONFLICT RESOLUTION DATA SET                                    */
/* AGGREGATE GROUP=agname, VERSION=nnnn                            */
/* SPECIFY datasetname   action   ON EACH LINE                     */
/* VALID ACTIONS ARE BYPASS, REPLACE,                              */
/* RENAMESOURCE(level) or RENAMETARGET(level)                      */
DATASETNAME1                                BYPASS
DATASETNAME2                                BYPASS

⋮
```

Figure 45. Conflict Resolution Data Set

To indicate different conflict resolution actions taken by ARECOVER verification processing for each data set entry in the conflict resolution data set, you may edit the conflict resolution data set before issuing the next ARECOVER command. As shown in Figure 45, the format of one line of the conflict resolution data set is a data set name followed by the action to be taken. The syntax of the action is the same as that of the ARECOVER command DATASETCONFLICT subparameters. Valid actions are RENAMESOURCE(*level*), RENAMETARGET(*level*), REPLACE, and BYPASS. Not all actions are valid for all data sets, based on restrictions discussed in section "Order of Processing for Conflict Resolution" on page 367.

When a subsequent ARECOVER command is issued and the conflict resolution data set exists, it is read and used as input to the verification process. If the conflict resolution data set does not meet format restrictions, a message is issued and ARECOVER processing continues, but ends following verification processing, even if EXECUTE has been specified. The conflict resolution data set is ignored during the verification process when this error occurs.

The lines containing conflict resolution actions are parsed for valid syntax, and data set names or qualifiers are also parsed for validity. If invalid syntax is found, or a data set name entry in the conflict resolution data set does not match the name of a data set entry in the ARECOVER control file, or a data set name entry is a duplicate of another entry, a message is issued and ARECOVER processing continues, ignoring the erroneous entries during verification processing.

When a like-named conflict is found for a data set name that has an entry in the conflict resolution data set, the action specified in the conflict resolution data set is used to attempt to resolve the conflict.

When VERIFY is specified, the action is not actually performed, but restrictions are validated. If the action in the conflict resolution data set is invalid for the data set, a message is issued indicating this, and if the conflict is not resolved by other conflict resolution processing, the data set fails verification. The verification process continues with the next data set.

When you use the ARECOVER command with the VERIFY parameter, a conflict resolution data set may be created. This conflict resolution data set contains entries for data sets that had a name conflict during verification processing, and a default action of BYPASS is indicated for these data sets. If you determine that the conflicts can be handled by either the RECOVERNEWNAMEALL, RECOVERNEWNAMELEVEL, or DATASETCONFLICT parameters on a subsequent ARECOVER command, then you delete the conflict resolution data set before issuing the command. If this deletion is not done, ARECOVER processing uses the conflict resolution data set action of BYPASS, and the data sets are skipped.

When EXECUTE is specified, the conflict resolution action is validated. If valid, the action is performed during data movement. If the action in the conflict resolution data set is invalid for the data set, a message is issued indicating this. If the conflict is not resolved by other conflict resolution processing, the data set fails verification, and is not recovered. ARECOVER EXECUTE processing continues for the remaining data sets. If an error occurs while performing the action during EXECUTE processing, a message is issued, and the data set is bypassed and not recovered.

When EXECUTE is specified, if all conflicts are resolved, and if all data sets in the aggregate group have been recovered—not bypassed through use of the conflict resolution data set—the conflict resolution data set and the RESTART data set are deleted. Data sets which are bypassed by a conflict resolution data set action of BYPASS cause the RESTART data set and conflict resolution data set to be retained and a unique return code to be set. The bypass of these data sets may be intentional, but since the default action placed in the conflict resolution data set is BYPASS, and the default is also to use the conflict resolution data set if it exists, the data sets may be bypassed unintentionally. In this case, deletion of the conflict resolution data set and RESTART data set would make the recovery of the remaining data sets difficult. If your analysis at this point shows that all desired data sets have been recovered, you may then manually delete the RESTART data set and conflict resolution data set.

If the conflict cannot be resolved, an entry is added to the conflict resolution data set indicating a default action of BYPASS (if there is not an entry already, no duplicate entries are created and the action for a current entry is NEVER modified). The data set is not recovered and processing continues. An entry is *only* added to the conflict resolution data set if the conflict cannot be resolved by *any* of the preceding actions.

If the conflict has been found for a component of a VSAM data set being verified, the source VSAM cluster name is added to the conflict resolution data set (not the individual component names).

If the data set has been renamed before verification by a RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL subparameter, the original data set name is used for the entry in the conflict resolution data set.

Each data set successfully verified is recovered if the EXECUTE parameter is specified.

The following conflict resolution actions are supported.

- RENAMESOURCE(*level*)

  Verification checks to see if a data set is already cataloged with the new name at the recovery site. If the source data set is a GDS, neither a GDS generation nor a related GDG base may exist for the new name (a unique message is issued indicating GDS/GDG base conflicts). Before recovery of a renamed GDS generation, a GDG base is defined with the proper new name by using the attributes of the source GDG base. If no like-named conflict is found for the new data set name, the data set is renamed when it is recovered with the new high-level qualifier.

  If the new name would cause a like-named conflict, an appropriate message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

The RENAMESOURCE subparameter is not supported for ACCOMPANY data sets, or for migrated VSAM source data sets. In each case, a message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

- RENAMETARGET(*level*)

  Verification checks the attributes of the existing data set cataloged at the recovery site. If the target data set is one of the following, a message is issued, the target data set is not renamed, and an entry in the conflict resolution data set for the original data set name is created, indicating the default action of BYPASS.
  - tape data set
  - catalog
  - GDS
  - GDG base

  In addition, the target data set MUST be one of the following to be eligible for rename:
  - non-VSAM DASD
  - VSAM base cluster
  - VSAM alternate index (AIX)
  - Data component, index component, or path related to a VSAM cluster or AIX (but not related to a catalog)

  If the target object in conflict is a component or path related to a VSAM cluster or AIX, the entire target cluster, including component associations, is renamed.

  Verification checks to see if a data set is already cataloged at the recovery site with the new name. If so, an appropriate message is issued, the target data set is not renamed, and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

  If the data set meets restrictions for renaming and EXECUTE is specified, the existing data set at the recovery site is renamed. A message is issued indicating the action taken. When the conflict occurs for a component of a target VSAM data set, the entire target cluster, including associated components, is renamed. The indicated new high-level qualifier is used to rename the target objects. The ABACKUP data set is recovered with the original name. When the target data set is migrated, the RENAMETARGET action causes it to be recalled. If the rename action fails, a message is issued, and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

  RENAMETARGET(*level*) performs an ALTER NEWNAME for the existing data set name. Special care should be taken when you change the high-level qualifier so that the existing data set does not become uncataloged, which may occur when no catalog alias exists for the new high-level qualifier.

- BYPASS

  The data set is not recovered during data movement. When a conflict occurs for a component of a source VSAM data set, the entire source VSAM cluster is bypassed and not recovered. A message is issued indicating the action taken.

- REPLACE

  Verification checks to see if the existing data set at the recovery site meets restrictions for data sets that may be deleted. If restrictions are not met, a message is issued, and the data set is not recovered. If the data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

Note: When using the DSCONFLICT(REPLACE) parameter, the data set is deleted prior to attempting the recovery of the data set from the aggregate backup output files. If the recovery fails, the data set remains deleted. Causes of recovery failure include:

- I/O errors on the input tape
- Failure to allocate the new data set
- Recovering at a site that does not support the type of data set that is being recovered

If a GDG base exists and DSCONFLICT(REPLACE) has not been specified, recovery of the associated GDSs fails and an entry is placed in the conflict resolution data set with a default action of BYPASS.

If restrictions are met and EXECUTE is specified, the data set at the recovery site is deleted. When a conflict occurs for a component of a source VSAM data set, the entire target VSAM cluster related to the object in conflict is deleted. A message is issued indicating the action taken. If the delete for the REPLACE subparameter fails, a message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

ARCCREXT Installation Exit

Programming Interface Information

If the like-named conflict has not yet been resolved, and the installation exit ARCCREXT is active, the exit is called to attempt to resolve the conflict. If a conflict is found for a component of a source VSAM cluster being verified, only the source VSAM cluster name is passed to the installation exit. The action specified for the VSAM cluster name is used to resolve the conflict for the entire related VSAM cluster.

The installation exit passes back a return code to indicate what action to take to resolve the conflict. If the installation exit returns an invalid action for the data set, a message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

Return codes indicate the following actions:

- 0 = no action
- 4 = bypass the data set
- 8 = replace the existing data set
- 12 = rename the source data set (the exit specifies the new name)
- 16 = rename the existing data set at the recovery site with the returned new high-level qualifier before recovery. The source data set is recovered with the original name. This action is comparable to the DATASETCONFLICT RENAMETARGET(level) parameter.

End of Programming Interface Information

Migrated non-VSAM data sets can be renamed if only the high-level qualifier of the data set name is changed. Using this exit to rename more than just the data set's high-level qualifier causes an error (message ARC6325E) and the data set will not be processed during ARECOVER.

After conflict resolution processing for all data sets in the control file during verification and before actual data movement, a check is done to verify that no data sets to be recovered are renamed to the same name as another data set entry in the control file. When a data set name is renamed to the same name as an original source data set also being recovered, it is not renamed or recovered. When two or more data sets are renamed to the same name, only the first one is renamed and recovered. Error messages are issued for the other data sets that do not get recovered.

If all verification errors are resolved during ARECOVER EXECUTE processing, the conflict resolution data set is deleted.

DFSMShsm allows a single ARECOVER to invoke DFSMSdss multiple times to RESTORE all required data sets. Two hundred fifty-five data sets are renamed per invocation of DFSMSdss, until all data sets have been recovered.

Order of Processing for Conflict Resolution

A like-named conflict is identified when a catalog entry exists at the recovery site with the same name as the data set being recovered.

During like-named conflict checking, if the data set being recovered is a VSAM data set, each component of the source VSAM data set is used to locate a like-named conflict. The action taken to resolve the conflict is discussed below under each conflict resolution subparameter.

If a like-named conflict is detected, the conflict is resolved by the following order of conflict resolution actions:

1. If a conflict resolution data set exists and contains an entry for the data set in conflict, the requested action is validated.

   If valid, a message is issued indicating an action to take to resolve the conflict during EXECUTE processing. If the action is invalid, a message is issued indicating this, and conflict resolution processing is continued for the data set.

   If the conflict was found for a component of a VSAM data set, an entry must be found in the conflict resolution data set for the related source VSAM cluster name. If found, the action specified for the related source VSAM cluster name is used to resolve the conflict for the cluster and all the components.
2. If the conflict is not yet resolved, and the DATASETCONFLICT parameter is specified on the ARECOVER command, the requested action is validated.

   If valid, a message is issued indicating an action to take to resolve the conflict during EXECUTE processing. If the action is invalid, a message is issued indicating this, and conflict resolution processing is continued for the data set.

Programming Interface Information

3. If the conflict is not yet resolved and the installation exit ARCCREXT is active, the exit is called to resolve the conflict. The requested action is validated. If valid, a message is issued indicating an action to take to resolve the conflict during EXECUTE processing. If the action is invalid, a message is issued indicating this.

End of Programming Interface Information

RACF Facility Class Authorization Checking

RACF facility class checking supports the DATASETCONFLICT subparameters of REPLACE and RENAMETARGET. When either of these actions is specified

through the installation exit ARCCREXT, or through the conflict resolution data set, the user must have the same facility class authorization as when these subparameters are specified on the ARECOVER command.

When facility class authorization is denied for either REPLACE or RENAMETARGET specified as a subparameter of the DATASETCONFLICT parameter on the ARECOVER command, ARECOVER processing fails and a message is issued.

When facility class authorization is denied for REPLACE or RENAMETARGET specified as an action in the conflict resolution data set or in the installation exit ARCCREXT, ARECOVER verification processing fails for the individual data set, and a message is issued. Other data sets not receiving verification errors are processed as ARECOVER continues.

RACF data set authorization is checked before processing the REPLACE or RENAMETARGET action for individual data sets during ARECOVER verification processing, to obtain further security if restricted facility class authorization is indicated. Restricted facility class authorization is indicated when the user has READ or greater access to the following facility class profiles (where applicable):
- STGADMIN.ARC.ARECOVER.agname.REPLACE
- STGADMIN.ARC.ARECOVER.agname.RENTGT

When restricted facility class authorization is indicated, RACF ALTER access authority is required for individual data sets when conflict resolution indicates an action of REPLACE or RENAMETARGET for the existing data set at the ARECOVER site. If RACF access authority is denied, a message is issued and the data set fails verification processing, unless another conflict resolution action (BYPASS or RENAMESOURCE) is selected through the normal order of conflict resolution processing. Remaining data sets are verified and recovered if verification is successful.

The hierarchy of facility class command authorization is determined as follows:

1. Facility class must be active. If not active, the ARECOVER command fails if the user is not DFSMShsm-authorized.
2. If facility class is active but the appropriate profiles are not defined, processing continues as if facility class were inactive. The ARECOVER command fails if the user is not DFSMShsm-authorized.
3. ARECOVER processing checks for restricted authority. Restricted command authority is granted when the user is authorized to the following facility class profiles with an access authority of READ or greater.
   - STGADMIN.ARC.ARECOVER.agname
   - STGADMIN.ARC.ARECOVER.agname.REPLACE (for REPLACE)
   - STGADMIN.ARC.ARECOVER.agname.RENTGT (for RENAMETARGET)

   When restricted command authority is indicated, RACF data set authorization checking is performed each time a conflict is detected during ARECOVER verification processing, and a conflict resolution action of REPLACE or RENAMETARGET is selected. The user must have ALTER access to the data set to allow the conflict resolution action to be taken.
4. If restricted authority is not granted, ARECOVER processing checks for comprehensive command authority. Comprehensive authority is granted when a user is authorized to the STGADMIN.ARC.ARECOVER Facility Class profile, with an access authority of READ or greater.

When comprehensive authority is granted, RACF data set authorization checking is not performed during ARECOVER processing.

Holding and Releasing ABARS Functions

The DFSMShsm HOLD and RELEASE commands can be used to stop, hold, and release certain aggregate backup and aggregate recovery functions.

Related reading: For additional information about the HOLD and RELEASE commands, see the *z/OS DFSMShsm Storage Administration Reference*.

Holding ABARS Functions

You can use the DFSMShsm HOLD command to hold ABARS functions and to designate a specific aggregate group or control file data set name to be held. The commands HOLD ABACKUP or HOLD ARECOVER may be used to hold ABARS functions, and the commands HOLD ABACKUP(*agname*) or HOLD ARECOVER [DATASETNAME(*controlfiledsn*) | AGGREGATE(*agname*)] may be used to hold a specific aggregate group or control file within the ABARS functions.

If the HOLD EOD option is specified for ABACKUP or ARECOVER, the following processing takes place:

Note: If an aggregate group or control file data set name is specified, this processing takes place only when the specified name matches a currently running or pending ABACKUP or ARECOVER command.

- Currently executing ABACKUP or ARECOVER commands are stopped, at the end of processing a data set, and are not requeued for processing. Data sets and tapes are closed and deallocated, all memory is freed, and processing is immediately terminated.
- If the HOLD EOD option is specified for a currently running ABACKUP command, the ABACKUP output files are uncataloged and the tape volume serial numbers are removed from the HSMABR tape connected set if they are indicated to be RACF protected.
- Pending ABACKUP or ARECOVER commands that were submitted without using the WAIT option are held. They remain in the queue and are not executed until a RELEASE command is issued.
- Pending ABACKUP or ARECOVER commands submitted using the WAIT option are failed with an accompanying error message.

If the HOLD command is specified for the ABACKUP or ARECOVER function (without the EOD option), the following processing takes place:

Note: If an aggregate group or control file data set name is specified, this processing takes place only when the specified name matches a currently running or pending ABACKUP or ARECOVER command.

- Currently executing ABACKUP or ARECOVER commands run to completion. If an aggregate group or control file data set name has been specified, subsequent commands for that name are held.
- Pending ABACKUP or ARECOVER commands submitted without using the WAIT option are held. They remain in the queue and are not executed until a RELEASE command is issued.
- Pending ABACKUP or ARECOVER commands submitted using the WAIT option are failed with an accompanying error message.

When the HOLD command is specified for an ABACKUP or ARECOVER command using an aggregate group or control file data set name and the command is neither running nor pending, subsequent commands issued using that name are held.

Note: During ABACKUP processing, if the control file has already been written and a HOLD EOD is issued, ABACKUP completes normally. However, if ABACKUP has been issued with the MOVE option, ABACKUP will be held.

Releasing Held ABARS Functions

The RELEASE command, used with the ALL, ABACKUP, or ARECOVER parameters, releases a held ABARS function. The RELEASE command also can be used to specify the aggregate group name for ABACKUP and the aggregate group or control file name with the (ARECOVER AGGREGATE | ARECOVER DATASETNAME) parameter. The previously held ABARS function is released.

Querying Held ABARS Functions

You can use the QUERY ACTIVE command to perform the following tasks:

- List active ABACKUP or ARECOVER commands with their associated aggregate group or control file name
- Indicate if ABARS is active or inactive
- Indicate whether ABACKUP and ARECOVER is held, not held, or held EOD
- Indicate the specific ABACKUP(*agname*) and ARECOVER [AGGREGATE(*agname*) | DATASETNAME(*controlfiledsn*)] commands that are held

Displaying SMS Class Definitions during ABACKUP and ARECOVER

During DFSMShsm ABACKUP processing, a list of SMS construct names—along with individual class definitions for the SMS data, management, and storage classes associated with the data sets in the aggregate group being processed—are written to the control file.

During subsequent ARECOVER processing, a message is issued to the ARECOVER activity log listing the attributes of each SMS data class, management class, and storage class definition present at the ABACKUP site. This allows the recovery site to make any necessary changes to either existing SMS constructs or the SMS environment.

If, however, it is determined that the ABACKUP activity log has been recovered by the ARECOVER command specified with the ACTIVITY parameter (see "Backing Up and Recovering the ABARS Activity Log" on page 360), the SMS construct definitions are not listed in the ARECOVER activity log.

Notes:

1. Use the VERIFY or PREPARE option of the ARECOVER command before the EXECUTE option. By doing this, you can determine SMS class definitions, and any needed changes can be made to the SMS environment before actual data movement.
2. Use the ACTIVITY option for ARECOVER as an alternative method of retrieving the SMS construct information (see "Backing Up and Recovering the ABARS Activity Log" on page 360). By using this method, you can obtain SMS construct information from the ABACKUP activity log, if DASD logging has been specified.

ABARS User Catalog and Catalog Alias ARECOVER Support

If user catalogs have been specified in the allocate list during aggregate backup, the catalog data and alias definitions are stored in the control file and used by DFSMShsm during aggregate recovery to define the catalog and its aliases at the recovery location.

The RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL parameters do not apply to user catalogs in the ALLOCATE list. This restriction prevents conflicts in defining an alias that is the same as the new high-level qualifier.

If the catalogs specified in the ALLOCATE list have conflicts with existing data set names, aggregate recovery verification processing must ensure that these conflicts are resolved with either the conflict resolution data set, the DATASETCONFLICT parameter of the ARECOVER command, or through the invocation of the conflict resolution installation exit ARCCREXT.

All conflict resolution actions are supported to resolve a like-named conflict for a catalog to be recovered, provided the restrictions for each action are met, as discussed in "ARECOVER Data Set Name Conflict Resolution" on page 361. (The most notable restriction is that RENAMETARGET and REPLACE are not allowed when the existing data set is a catalog.)

If all catalog name conflicts are not resolved, verification fails. An error message is issued for each unresolved catalog conflict. A default entry (BYPASS) is *not* added to the conflict resolution data set, since a catalog recovery should not be bypassed as the default. (The user may, however, manually add an entry in the conflict resolution data set to resolve the catalog name conflict.) The remaining verification is completed, and the ARECOVER processing ends. No further processing takes place, even if EXECUTE was specified.

The alias definitions associated with the user catalogs in the ALLOCATE list must be checked against the current catalog structure at the aggregate recovery site, to ensure the alias is not already defined to an existing catalog. If the alias is already defined, an error message is issued during aggregate recovery verification processing. This error message indicates to which catalog this alias is already defined. Aggregate recovery verification is not failed, but the particular alias definition is not issued.

If any user catalogs have been renamed during conflict resolution RENAMESOURCE processing, the new high-level qualifier must be checked against any aliases not yet defined to ensure that no like-named conflicts exist. If conflicts exist, a message is issued and the renamed user catalog is not recovered unless the conflict can be resolved through means other than RENAMESOURCE.

The DEFINE of the user catalogs in the ALLOCATE list and the DEFINE for the associated ALIASES are performed before the data sets for the particular aggregate are recovered. This allows the data sets in the aggregate being recovered to be cataloged in these catalogs if the data sets have an alias defined to one of these catalogs.

Note: ABARS does not allow multiple defines of aliases at the recovery location. If an alias is defined to a catalog being recovered and a catalog already in existence, the catalog in existence takes precedence.

The aggregate recovery verification step lists the user catalogs contained in the control file and all of the aliases defined to those catalogs. This action is performed when the user specifies the ARECOVER command with the VERIFY parameter.

Data Sets That Are Dynamically Allocated during Aggregate Recovery Processing

In addition to the RESTART and conflict resolution data sets, DFSMShsm dynamically allocates the following data sets during aggregate recovery processing. These data sets are used while aggregate recovery is executing and are dynamically deallocated before completion of aggregate recovery processing. These data sets are allocated with a unit type of SYSALLDA. (If they are non-SMS-managed, you must have at least one STORAGE- or PUBLIC-type device online to use the system, and if they are SMS-managed, your ACS routine must assign them to a storage group that contains storage-type volumes.) You need to be aware of these data set names so that the high-level qualifiers used by ABARS can be defined to the system. You may want to modify your ACS filters to direct allocation of these data sets to specific volumes.

SYSIN data set: This data set is used to pass RESTORE parameters to DFSMSdss. The SYSIN data set name is:

authid.SYSIN.*agname*.T*hhmmss*

Where

authid is the DFSMShsm-authorized user ID,

agname is the aggregate group name being processed, and

hhmmss is hours, minutes, seconds.

FILTER data set: This data set is used to pass a list of data set names to DFSMSdss. The FILTER data set name is:

authid.FILTER.*agname*.T*hhmmss*

Where

authid is the DFSMShsm-authorized user ID,

agname is the aggregate group name being processed, and

hhmmss is hours, minutes, seconds.

Aggregate Recovery Output

The activity log, recovered data sets, FSR records, updated catalog entries, GDG base definitions, and updated DFSMShsm CDS records are the output of aggregate recovery.

The conflict resolution data set and the restart data set are the output of the ARECOVER command when data set conflicts or errors occur. These have been described in previous sections. See “Restart Data Set” on page 359.

Activity Log

An activity log is created each time you issue the ARECOVER command. You may find the activity log useful because it contains messages output during aggregate recovery processing.

The format of the activity log name is:

mcvtactn.H*mcvthost*.ARECOVER.*agname*.D*yyddd*.T*hhmmss*

where:

mcvtactn is the field in the MCVT that contains the value stored for the activity log high-level qualifier. The default value is HSMACT.

H*mcvthost* is the DFSMShsm host identification.

agname is the name of the aggregate group being processed.

ARECOVER is a constant for aggregate recovery.

D*yyddd* is the date.

T*hhmmss* is the time.

The activity log is written to either a SYSOUT or DASD data set as determined by the ABARSACTLOGTYPE parameter of the SETSYS command. If you specify DASD and the allocation attempt fails, an attempt is be made to allocate the activity log to SYSOUT. See "Defining SETSYS Parameters for Aggregate Recovery" on page 341 for more information on setting the activity log output type.

Figure 46 on page 374 is a partial example of the aggregate recovery activity log:

```
PAGE 0001  DFSMSHSM 1.4.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   97.238  18:16
ARC6102I AGGREGATE RECOVERY STARTING USING CONTROL FILE DATA SET PAY1.C.C01V0002, AT 18:16:00,
 STARTED TASK = DFHSMABR.ABAR0154
ARC6030I ACTIVITY LOG FOR CONTROL FILE DATA SET PAY1.C.C01V0002 WILL BE ROUTED TO SYSOUT=A
ARC6115I AGGREGATE RECOVERY USING CONTROL FILE DATA SET PAY1.C.C01V0002 WILL USE VOLUMES
  A00020
  A00016
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.NAMES DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.RETIRE DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.SOCSEC DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.STATUS.G0002V00 DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.INSTRUCT DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.CHECK DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.PGMLIB DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY002.LADDER DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY002.IRA DURING
AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.NAMES HAS BEEN UNCATALOGED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.RETIRE HAS BEEN UNCATALOGED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.SOCSEC HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.INSTRUCT HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.CHECK HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.PGMLIB HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY002.LADDER HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY002.IRA HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6004I 000{ ARECOVER PAGE 0001         5695-DF175  DFSMSdss         Data Set Services         97.238  18:16
ARC6004I 000{ ARECOVER ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 000{ ARECOVER  RESTORE DATASET(FILTERDD(SYS00016)) -
ARC6004I 000{ ARECOVER  INDDNAME(SYS00015) -
ARC6004I 000{ ARECOVER  OUTDYNAM( -
ARC6004I 000{ ARECOVER   (DUMPDS) -
ARC6004I 000{ ARECOVER   ) -
ARC6004I 000{ ARECOVER  PERCENTUTILIZED( -
ARC6004I 000{ ARECOVER  080 -
ARC6004I 000{ ARECOVER   ) -
ARC6004I 000{ ARECOVER  SPHERE  -
ARC6004I 000{ ARECOVER  TGTGDS(SOURCE) -
```

Figure 46. Example of an Aggregate Recovery Activity Log (Part 1 of 2)

```
PAGE 0002  DFSMShsm        DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   97.238  18:16
ARC6004I 000{ ARECOVER  CATALOG FORCE
ARC6004I 000{ ARECOVER ADR101I       RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
ARC6004I 000{ ARECOVER ADR109I       RI01 (01), 97238 18:16:41 INITIAL SCAN OF USER CONTROL STATEMENTS
COMPLETED.
ARC6004I 000{ ARECOVER ADR006I (001)-SETUP(01), 97238 18:16:42 EXECUTION BEGINS
ARC6004I 000{ ARECOVER ADR780I (001)-TDDS (01), THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL
DATA SET FORMAT
ARC6004I 000{ ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY2.PY002.SOCSEC ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000{ ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY2.PY002.SOCSEC HAS BEEN CATALOGED I
IN CATALOG TPCMAST1
ARC6004I 000{ ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY2.PY002.SOCSEC WAS RESTORED
ARC6004I 000{ ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY2.PY002.STATUS.G0002V00 ALLOCATED, ON
VOLUME(S): DUMPDS
ARC6004I 000{ ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 000{ ARECOVER DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000{ ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY2.PY002.STATUS.G0002V00 WAS RESTORED
ARC6004I 000{ ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY1.INSTRUCT ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000{ ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.INSTRUCT HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000{ ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.INSTRUCT WAS RESTORED
ARC6004I 000{ ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY1.PY001.CHECK ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000{ ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.PY001.CHECK HAS BEEN CATALOGED IN CATALOG
TPCMAST1
ARC6004I 000{ ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.PY001.CHECK WAS RESTORED
ARC6004I 000{ ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY1.PY001.PGMLIB ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000{ ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.PY001.PGMLIB HAS BEEN CATALOGED IN CATALOG
TPCMAST1
ARC6004I 000{ ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.PY001.PGMLIB WAS RESTORED
ARC6004I 000{ ARECOVER ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 000{ ARECOVER                          PAY2.PY002.SOCSEC
ARC6004I 000{ ARECOVER                          PAY2.PY002.STATUS.G0002V00
ARC6004I 000{ ARECOVER                          PAY1.INSTRUCT
ARC6004I 000{ ARECOVER PAGE 0002         5695-DF175  DFSMSdss         Data Set Services         97.238  18:16
ARC6004I 000{ ARECOVER                          PAY1.PY001.CHECK
ARC6004I 000{ ARECOVER                          PAY1.PY001.PGMLIB
ARC6004I 000{ ARECOVER ADR013I (001)-CLTSK(01), 97238 18:17:00 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 000{ ARECOVER ADR012I (SCH)-DSSU (01), 97238 18:17:00 DFSMSdss PROCESSING COMPLETE. HIGHEST
RETURN CODE IS 0000
ARC6109I REPLACE OPTION SPECIFIED,
 GDG DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN RESTORED USING EXISTING GDG ATTRIBUTES
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY RECOVERED USING AGGREGATE GROUP PAY1:
  PAY1.PY001.NAMES
  PAY2.PY002.RETIRE
  PAY2.PY002.SOCSEC
  PAY2.PY002.STATUS.G0002V00
  PAY1.INSTRUCT
  PAY1.PY001.CHECK
  PAY1.PY001.PGMLIB
  PAY1.PY002.LADDER
  PAY1.PY002.IRA
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY ALLOCATED USING AGGREGATE GROUP PAY1:
  PAY2.PY001.BENEFIT
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY CATALOGED USING AGGREGATE GROUP PAY1:
  PAY2.PY001.TAXES
ARC6103I AGGREGATE RECOVERY HAS COMPLETED FOR AGGREGATE GROUP PAY1, USING CONTROL FILE
DATA SET PAY1.C.C01V0002,
 AT 18:17:31, RETCODE = 000
```

Figure 46. Example of an Aggregate Recovery Activity Log (Part 2 of 2)

Recovered Data Sets at the Example Aggregate Recovery Site

At the example site, the following data sets have been recovered to the DASD volumes specified:

- Recovered to an SMS-managed volume:
  PAY2.PY002.SOCSEC

- Recovered to the level 0 volume PRIM02:
  PAY2.PY002.STATUS.G0002V00
  PAY1.PY001.CHECK
- Recovered to the migration volume MIG102:
  PAY1.PY002.IRA
  PAY1.PY001.BONUS
  PAY1.PY002.LADDER

The following data sets have been recovered to tape:
- PAY1.PY001.NAMES

Data set PAY2.PY001.BENEFIT has been allocated to the level 0 volume PRIM02.
Data set PAY2.PY001.TAXES has been cataloged.

ARECOVER CPU Time and Accounting Codes

ABARS records the CPU time for processing ABACKUP and ARECOVER requests. This information is written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANy system data sets.

The ABACKUP CPU time and ARECOVER CPU time are also recorded in the ABR record. There is a difference, however, between the ARECOVER CPU time recorded in the FSR control block and the ABR record. If an ARECOVER request fails and is reissued with a valid RESTART data set, the CPU time in the FSR only reflects the time to process the remaining data sets. The ABR record accumulates the CPU times of each restart until the recovery of the aggregate is successful.

The ISMF aggregate group definition panel allows you to specify an accounting code for an aggregate group. The accounting code gets written to the FSR control block (and associated SMF record where one exists) along with the ABACKUP and ARECOVER CPU processing time for the aggregate. Installations can then use the information to assist in calculating charges to users for ABARS requests.

The accounting code is also written to the ABR record, and the ABACKUP control file. It is written to the ABACKUP control file so the code can be used at the recovery site without requiring an aggregate definition at the recovery site.

The accounting code is optional, and can be up to 32 characters in length. Valid characters are alphanumeric characters, blanks, commas, periods, parentheses, hyphens, slashes, ampersands, and apostrophes (single quotes).

Using ISMF Panels to Perform Aggregate Backup

You can use ISMF panels to process aggregate backup instead of issuing the ABACKUP command. You can specify the DFSMSdfp Aggregate Group Application Selection panel from TSO and use either the LIST or the ABACKUP option for aggregate group backup.

Use the ABACKUP option when you know the aggregate group name you want to back up.

Use the LIST option when you are unsure of the aggregate group name you want to back up or when you want to back up more than one aggregate group.

In the example system, ACTIVE is used for the CDS NAME and the aggregate group name PAY1 is backed up to a 3490 tape unit. Two methods of performing aggregate backup, using either the ISMF LIST or the ISMF ABACKUP option, are shown.

Using the ISMF LIST Option to Perform Aggregate Backup

To use the LIST option for aggregate backup, use TSO ISPF panels to access the DFSMSdfp Aggregate Group Application Selection panel, shown in Figure 47.

```
                    AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===> _

TO PERFORM AGGREGATE GROUP OPERATIONS,SPECIFY:

 CDS NAME             ===> 'ACTIVE'
                                          (1 to 44 character dataset name)

 AGGREGATE GROUP NAME ===> *              (for Aggregate Group List, fully or
                                          partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 1

   1  LIST      - Generate a list of Aggregate Groups
   2  DISPLAY   - Display an Aggregate Group
   3  DEFINE    - Define an Aggregate Group
   4  ALTER     - Alter an Aggregate Group
   5  ABACKUP   - Backup an Aggregate Group
   6  ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
 RESPECIFY SORT CRITERIA            ===> N    (Y or N)

USE ENTER TO PERFORM SELECTION:
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 47. Aggregate Group Application Selection Panel

On the Aggregate Group Application Selection Panel:

1. Enter 'ACTIVE' (quotes required) for **CDS NAME**. You must specify ACTIVE when backing up data sets, because aggregate backup can be initiated only from an active CDS.
2. Enter an asterisk (*) for **AGGREGATE GROUP NAME** to display a list of aggregate group names.
3. Enter 1 for **SELECT ONE OF THE FOLLOWING OPTIONS** to use the LIST option.
4. Press the Enter key to display the Aggregate Group List panel, as shown in Figure 48 on page 378.

```
                              AGGREGATE GROUP LIST
COMMAND ===> _                                                SCROLL ===> PAGE

                                                        Entries 1-1 of 1
                                                        Data Columns 16-19 of 19
CDS NAME:  ACTIVE

ENTER LINE OPERATORS BELOW:

    LINE       AGGROUP   LAST DATE   LAST TIME   NUMBER     MGMTCLAS
    OPERATOR   NAME      MODIFIED    MODIFIED    OF COPIES  NAME
   ---(1)----  --(2)---  ---(16)---  ---(17)---  --(18)---  --(19)--
   abackup     PAY1      1995/01/08  12:45               1  ABARMC1
               REMOTE    1995/02/04  08:25               1  ABARMC2
               TUCSON    1995/05/11  13:50               1  ABARMC2
   ----------  --------  ------  BOTTOM  OF  DATA  ------  --------  ----------




USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 48. Aggregate Group List Panel

On the Aggregate Group List Panel, enter ABACKUP next to the aggregate group name PAY1 and press the Enter key to display the Aggregate Group Backup panel, as shown in Figure 49.

```
  Panel Utilities Help
-------------------------------------------------------------------------------
DFQDCAG4              AGGREGATE GROUP BACKUP                    Page 1 of 2
COMMAND ===> ____________________________________________________________________

CDS NAME: ACTIVE

  AGGREGATE GROUP NAME  . . PAY1


  UNIT NAME . . . . . . . . _________
  PROCESSING OPTION . . . . 2                 (1=VERIFY, 2=EXECUTE)
  WAIT FOR COMPLETION . . . N                 (Y or N)
  STACK / NOSTACK . . . . . _                 (S=STACK, N=NOSTACK or blank)
  OPTIMIZE  . . . . . . . . _                 (1 to 4 or blank)
  DELETE DATA SETS AFTER ABACKUP  . . _       (Y or N)

  FILTER OUTPUT DATA SET NAME (1 to 44 characters)
   ===> ____________________________________________

  Process only              _ L0       _ ML1       _ ML2       _ USERTAPE




USE ENTER CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 49. Aggregate Group Backup Panel

On the Aggregate Group Backup Panel, you do not need to enter the CDS NAME or the AGGREGATE GROUP NAME; they are carried over from the previous panel.

Note: You can change the aggregate group name and initiate backup for another aggregate group from this panel.

On the Aggregate Group Backup Panel:

1. Enter 3490 for **UNIT NAME**.
2. Enter a 1 for **PROCESSING OPTION**. to perform aggregate backup verification processing, If you specify VERIFY, no data movement will take place.

   Enter a 2 for **PROCESSING OPTION** to EXECUTE aggregate group backup. Data movement will take place.
3. Enter a Y for **WAIT FOR COMPLETION** to specify that you want to wait for aggregate backup to complete. If you specify Y, you will not be able to perform any other functions from your terminal while aggregate backup is processing.

   Enter an N for **WAIT FOR COMPLETION** to specify that you don't want to wait.
4. Enter an S for **STACK / NOSTACK** if you want to stack your ABARS output files on a minimum number of tape volumes. Be aware that you can only recover stacked output files on DFSMShsm hosts running DFSMS/MVS Version 1 Release 4 or subsequent releases.

   Enter an N for **STACK / NOSTACK** if you do not want to stack your ABARS output files. This allows you to recover your ABARS output files on hosts running DFSMS/MVS releases older than Version 1 Release 4.

   Enter a blank for **STACK / NOSTACK** to cause aggregate backup processing to default to the value specified by the SETSYS ABARSTAPES(STACK | NOSTACK) command. If SETSYS ABARSTAPES isn't specified, the default is STACK.
5. Enter one of the following values for OPTIMIZE, which ABACKUP processing passes to DFSMSdss when DFSMSdss is backing up the level 0 DASD data sets specified in the include list.

Value	Description
1	Causes DFSMSdss to read one track at a time.
2	Causes DFSMSdss to read two tracks at a time.
3	Causes DFSMSdss to read five tracks at a time.
4	Causes DFSMSdss to read one cylinder at a time.
blank	Causes this option to default to the setting specified in the SETSYS ABARSOPTIMIZE(*n*) command. If SETSYS ABARSOPTIMIZE is not specified, the default is 3 (DFSMSdss reads five tracks at a time).
6. Enter Y for **DELETE DATA SETS AFTER ABACKUP** if you want the data sets deleted following successful completion of ABACKUP processing.

   Enter N for **DELETE DATA SETS AFTER ABACKUP** if you don't want the data sets deleted.
7. Enter a slash (/) to choose the **PROCESS ONLY** selection or selections you want to include in your processing. If you don't choose any PROCESS ONLY selections, then the PROCESS ONLY option is not in effect. When the PROCESS ONLY option is not in effect, it is the same as choosing all of the selections; L0, ML1, ML2, and USERTAPE data sets are all eligible for processing.
8. Press the Enter key to initiate processing.

Using the ISMF ABACKUP Option to Perform Aggregate Backup

To use the ABACKUP option for aggregate backup, use TSO ISPF panels to access the DFSMSdfp Aggregate Group Application Selection panel, shown in Figure 50.

```
                  AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===> _

TO PERFORM AGGREGATE GROUP OPERATIONS,SPECIFY:

 CDS NAME             ===> 'ACTIVE'
                                          (1 to 44 character dataset name)

 AGGREGATE GROUP NAME ===> PAY1           (for Aggregate Group List, fully or
                                          partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 5

   1  LIST      - Generate a list of Aggregate Groups
   2  DISPLAY   - Display an Aggregate Group
   3  DEFINE    - Define an Aggregate Group
   4  ALTER     - Alter an Aggregate Group
   5  ABACKUP   - Backup an Aggregate Group
   6  ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY SORT CRITERIA             ===> N      (Y or N)

USE ENTER TO PERFORM SELECTION:
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 50. Aggregate Group Application Selection Panel

On the Aggregate Group Application Selection Panel:

1. Enter 'ACTIVE' (quotes required) for **CDS NAME**. You must specify ACTIVE when backing up data sets, because aggregate backup can be initiated only from an active CDS.
2. Enter PAY1 for **AGGREGATE GROUP NAME**.
3. Enter 5 for **SELECT ONE OF THE FOLLOWING OPTIONS** to use the ABACKUP option.
4. Press the Enter key to display the Aggregate Group Backup panel, as shown in Figure 51.

```
  Panel Utilities Help
------------------------------------------------------------------------------
DFQDCAG4              AGGREGATE GROUP BACKUP                    Page 1 of 2
COMMAND ===> _____________________________________________________________

CDS NAME: ACTIVE

  AGGREGATE GROUP NAME  . . PAY1


  UNIT NAME . . . . . . . . _______
  PROCESSING OPTION . . . . 2              (1=VERIFY, 2=EXECUTE)
  WAIT FOR COMPLETION . . . N              (Y or N)
  STACK / NOSTACK . . . . . _              (S=STACK, N=NOSTACK or blank)
  OPTIMIZE  . . . . . . . . _              (1 to 4 or blank)
  DELETE DATA SETS AFTER ABACKUP  . . _    (Y or N)

  FILTER OUTPUT DATA SET NAME (1 to 44 characters)
   ===> ____________________________________________

  Process only             _ L0       _ ML1       _ ML2       _ USERTAPE




USE ENTER CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 51. Aggregate Group Backup Panel

You do not need to enter the CDS NAME or the AGGREGATE GROUP NAME; they are carried over from the previous panel.

Note: You can change the aggregate group name and initiate backup for another aggregate group from this panel.

On the Aggregate Group Backup Panel:

1. Enter 3490 for **UNIT NAME**.
2. Enter a 1 for **PROCESSING OPTION**. to perform aggregate backup verification processing, If you specify VERIFY, no data movement will take place.

   Enter a 2 for **PROCESSING OPTION** to EXECUTE aggregate group backup. Data movement will take place.
3. Enter a Y for **WAIT FOR COMPLETION** to specify that you want to wait for aggregate backup to complete. If you specify Y, you will not be able to perform any other functions from your terminal while aggregate backup is processing.

   Enter an N for **WAIT FOR COMPLETION** to specify that you don't want to wait.
4. Enter an S for **STACK / NOSTACK** if you want to stack your ABARS output files on a minimum number of tape volumes. Be aware that you can only recover stacked output files on DFSMShsm hosts running DFSMS/MVS Version 1 Release 4 or subsequent releases.

   Enter an N for **STACK / NOSTACK** if you do not want to stack your ABARS output files. This allows you to recover your ABARS output files on hosts running DFSMS/MVS releases older than Version 1 Release 4.

   Enter a blank for **STACK / NOSTACK** to cause aggregate backup processing to default to the value specified by the SETSYS ABARSTAPES(STACK | NOSTACK) command. If SETSYS ABARSTAPES isn't specified, the default is STACK.

5. Enter one of the following values for OPTIMIZE, which ABACKUP processing passes to DFSMSdss when DFSMSdss is backing up the level 0 DASD data sets specified in the include list:

Value	Description
1	Causes DFSMSdss to read one track at a time.
2	Causes DFSMSdss to read two tracks at a time.
3	Causes DFSMSdss to read five tracks at a time.
4	Causes DFSMSdss to read one cylinder at a time.
blank	Causes this option to default to the setting specified in the SETSYS ABARSOPTIMIZE(*n*) command. If SETSYS ABARSOPTIMIZE is not specified, the default is 3 (DFSMSdss reads five tracks at a time).

6. Enter Y for **DELETE DATA SETS AFTER ABACKUP** if you want the data sets deleted following successful completion of ABACKUP processing.

   Enter N for **DELETE DATA SETS AFTER ABACKUP** if you don't want the data sets deleted.
7. Enter a slash (/) to choose the **PROCESS ONLY** selection or selections you want to include in your processing. If you don't choose any PROCESS ONLY selections, then the PROCESS ONLY option is not in effect. When the PROCESS ONLY option is not in effect, it is the same as choosing all of the selections; L0, ML1, ML2, and USERTAPE data sets are all eligible for processing.
8. Press the Enter key to initiate processing.

Using ISMF Panels to Perform Aggregate Recovery

This section demonstrates how, instead of issuing the ARECOVER command, you can use ISMF panels to process aggregate recovery. You can specify the DFSMSdfp Aggregate Group Application Selection panel from TSO and use option 6 for aggregate group recovery.

For this example ISMF session, PAY1.C.C01V0001 is specified as the control file data set name, a 3490 unit type is specified, and volume 111111 is allocated for the recovery files.

To use the ARECOVER option for aggregate recovery, use TSO ISPF panels to access the DFSMSdfp aggregate group application selection panel, shown in Figure 52.

```
                    AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===> _

TO PERFORM AGGREGATE GROUP OPERATIONS,SPECIFY:

 CDS NAME             ===> 'ACTIVE'
                                      (1 to 44 character dataset name)

 AGGREGATE GROUP NAME ===> PAY1       (for Aggregate Group List, fully or
                                      partially specified or &asterisk for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 6

   1  LIST      - Generate a list of Aggregate Groups
   2  DISPLAY   - Display an Aggregate Group
   3  DEFINE    - Define an Aggregate Group
   4  ALTER     - Alter an Aggregate Group
   5  ABACKUP   - Backup an Aggregate Group
   6  ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
 RESPECIFY SORT CRITERIA            ===> N          (Y or N)

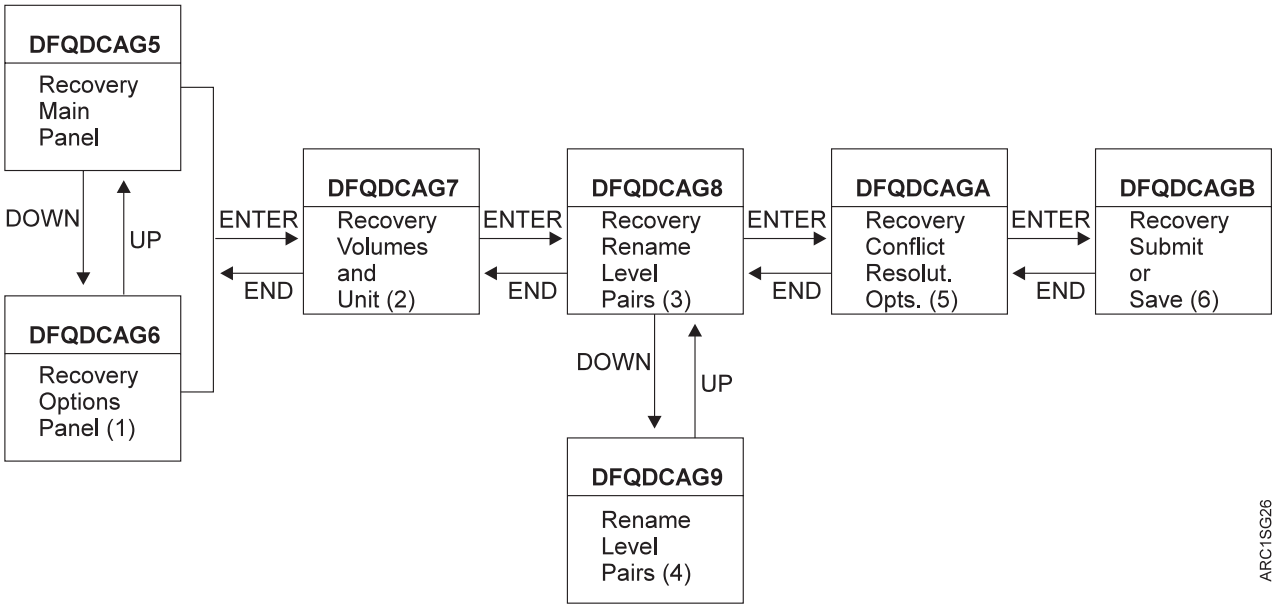
USE ENTER TO PERFORM SELECTION:
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 52. Aggregate Group Application Selection Panel

On the Aggregate Group Application Selection Panel:

1. Enter 6 in the SELECT ONE OF THE FOLLOWING OPTIONS field to select the ARECOVER option.
2. Press the ENTER key to display the first aggregate group recovery panel.

The following figure, with associated notes, maps the progression through the aggregate group recovery panels, and tells how each panel is accessed.



Notes:

1. Panel DFQDCAG6 is displayed only if DOWN is entered on panel DFQDCAG5.

2. Panel DFQDCAG7 is displayed only if a value is entered in the ABACKUP CONTROL DATASET field of panel DFQDCAG5, N is specified in the XMIT field of panel DFQDCAG5, and ENTER is entered on either panel DFQDCAG5 or panel DFQDCAG6.
3. Panel DFQDCAG8 is displayed only if Y is entered in the RENAME SELECTED DATA SETS field of panel DFQDCAG6, and ENTER is entered on panel DFQDCAG7.
4. Panel DFQDCAG9 is displayed only if DOWN is entered on panel DFQDCAG8.
5. Panel DFQDCAGA is displayed only if Y is entered in the SPECIFY CONFLICT RESOLUTION OPTION field of panel DFQDCAG6, and ENTER is entered on panel DFQDCAG8.
6. Panel DFQDCAGB is displayed only if all required fields are filled in on the other panels, and ENTER is entered on panel DFQDCAGA.

Panel DFQDCAG5

Figure 53 shows a representation of the ISMF DFQDCAG5 aggregate recovery main panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAG5                   AGGREGATE GROUP RECOVER               Page 1 of 7
Command ===> ____________________________________________________________________

  Abackup Control Dataset  . . 'PAY1.C.C01V0001'__________
                                          (1 to 44 Characters)
    Xmit . . . . . . . . . . . N          (Y or N)
    Stack / Nostack  . . . . . _          (S=STACK, N=NOSTACK or blank)

  Aggregate Group Name . . . . ________
    Date . . . . . . . . . . . __________  (yyyy/mm/dd)
    Version  . . . . . . . . . ____       (1 to 9999)

  Processing Option  . . . . . 3          (1=Prepare, 2=Verify, 3=Execute)
  Wait for Completion  . . . . Y          (Y or N)
  Target GDG Data Set Status . _          (A=ACTIVE, D=DEFERRED, R=ROLLEDOFF,
                                           S=SOURCE or blank)
  Volume Count . . . . . . . . _          (A=ANY, N=NONE or blank)
  Recover Instruction Data Set  . . N     (Y or N)
  Recover Activity Log  . . . . . . 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 53. Aggregate Recovery Panel DFQDCAG5

The following is a description of the fields of panel DFQDCAG5, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field generates a DATASETNAME(*dsname*) parameter.

  Enter 'PAY1.C.C01V0001' in this field
- The XMIT field indicates whether the ABACKUP files were independently transmitted to the ARECOVER site.

  Enter N in this field.
- The STACK / NOSTACK field indicates whether the ABACKUP output files were stacked on a minimum number of tapes, or not.

  Enter N in this field.
- The AGGREGATE GROUP NAME field generates an AGGREGATE(*agname*) parameter.

Leave this field blank.

- The DATE field generates a DATE(*yyyy/mm/dd*) parameter.

  Leave this field blank.
- The VERSION field generates a VERSION(*nnnn*) parameter.

  Leave this field blank.
- The PROCESSING OPTION field allows you to place either the PREPARE, VERIFY, or EXECUTE parameters in the ARECOVER command. A 1 generates the PREPARE parameter, a 2 generates the VERIFY parameter, and a 3 generates the EXECUTE parameter.

  Enter a 3 in this field.
- The WAIT FOR COMPLETION field, if Y is entered, generates an HSEND WAIT ARECOVER command, and the terminal is locked until the ARECOVER command completes. If N is entered, this field generates an HSEND ARECOVER command, and the terminal remains unlocked.

  Enter Y in this field.
- The TARGET GDG DATA SET STATUS field allows you to specify the status to which you want an SMS-managed generation data set returned. You can specify A for ACTIVE, D for DEFERRED, R for ROLLEDOFF, S for SOURCE, or leave the field blank. SOURCE indicates that you want the data set returned to the status it had when it was backed up. A blank will cause value specified by the SETSYS ARECOVERTGTGDS to be used, or its default value of SOURCE.

  Leave this field blank.
- The VOLUME COUNT field allows you to specify A for ANY, N for NONE, or a blank. If ANY is specified, the VOLCOUNT(ANY) parameter is passed to DFSMSdss during the restore, which results in the allocation of the target data set on as many volumes as required, up to a maximum of 59 volumes. If NONE is specified, the DFSMSdss VOLCOUNT parameter is not passed to DFSMSdss during the restore. If the field is left blank, the value specified by the SETSYS ABARSVOLCOUNT parameter, or its default value of NONE, is used.

  Leave this field blank.
- The RECOVER INSTRUCTION DATA SET field, when Y is entered, generates the INSTRUCTION parameter. If AGGREGATE GROUP NAME is specified with the PREPARE option, either the RECOVER INSTRUCTION DATA SET field, the RECOVER ACTIVITY LOG field, or both must be used.

  Enter N in this field.
- The RECOVER ACTIVITY LOG field, when Y is entered, generates the ACTIVITY parameter.

  Enter Y in this field.

Panel DFQDCAG6

Figure 54 on page 386 shows a representation of the ISMF DFQDCAG6 aggregate recovery options panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAG6                   AGGREGATE GROUP RECOVER               Page 2 of 7
Command ===> ____________________________________________________________

AGGREGATE GROUP NAME: 'PAY1.C.C0V0001'

  Model Entity . . . . _____________________________________________
                                    (1 to 44 Characters)
  Target Unit Name . . ________

  Target Utilization Percentage . . ___  (1 to 100)

  Backup Migrated Data Sets . . . . Y    (Y or N)
  New Migration Level . . . . . . . 3    (1=ML1, 2=ML2, 3=SOURCELEVEL)

  Rename All to High Level Qualifier  . . ________  (1 to 8 characters)
  Rename Selected Data Sets . . . . . . . Y         (Y or N)

  Specify Conflict Resolution Option  . . Y         (Y or N)


Use ENTER to Continue; Use UP to View Previous Options;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 54. Aggregate Recovery Panel DFQDCAG6

The following is a description of the fields of panel DFQDCAG6, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The MODEL ENTITY field allows the entry of a quoted, fully qualified data set name without a TSO *userid* prefix. This field generates, when filled in, a MENTITY(*dsname*) parameter.

  Leave this field blank.
- The TARGET UNIT NAME field generates, when filled in, a TARGETUNIT(*unitname*) parameter.

  Leave this field blank.
- The TARGET UTILIZATION PERCENTAGE field, if a percentage number is entered, generates a PERCENTUTILIZED(*nnn*) parameter.

  Leave this field blank.
- The BACKUP MIGRATED DATA SETS field generates, if Y is entered, a BACKUPMIGRATED parameter. If N is entered, a NOBACKUPMIGRATED parameter is generated.

  Enter Y in this field.
- The NEW MIGRATION LEVEL field indicates whether the MIGRATEDDATA parameter and associated subparameters are specified on the ARECOVER command. A 1 in this field generates a MIGRATEDDATA(ML1) parameter (the default), a 2 in this field generates a MIGRATEDDATA(ML2) parameter, and a 3 in this field generates a MIGRATEDDATA(SOURCELEVEL) parameter.

  Enter 3 in this field.

- The RENAME ALL TO HIGH LEVEL QUALIFIER field accepts a valid, 1 to 8 character data set name qualifier that generates a RECOVERNEWNAMEALL(*level*) parameter, with the specified *level* as the subparameter. A value in this field is mutually exclusive with a Y in RENAME SELECTED DATA SETS.

  Leave this field blank.
- The RENAME SELECTED DATA SETS field, if Y is entered, displays panel DFQDCAG8, shown in Figure 56 on page 388. A Y in this field is mutually exclusive with a value in the RENAME ALL TO HIGH LEVEL QUALIFIER field.

  Enter Y in this field.
- The SPECIFY CONFLICT RESOLUTION OPTION field, if Y is entered, displays panel DFQDCAGA, shown in Figure 58 on page 390.

  Enter Y in this field.

Panel DFQDCAG7

Figure 55 shows a representation of the ISMF DFQDCAG7 aggregate recovery volumes and unit panel.

```
   Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAG7                    AGGREGATE GROUP RECOVER              Page 3 of 7
COMMAND ===>

  ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'

    UNIT NAME ===> 3490

    VOLUMES   ===> 111111      ===>             ===>             ===>
              ===>             ===>             ===>             ===>
              ===>             ===>             ===>             ===>
              ===>             ===>             ===>




USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 55. Aggregate Recovery Panel DFQDCAG7

The following is a description of the fields of panel DFQDCAG7, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The UNIT NAME field generates the UNIT(*unitname*) parameter, where *unitname* is the unit name that you enter in the field.

  Enter 3490 in this field.
- The VOLUMES fields can accept up to 15 entries. These fields, when filled in, generate a VOLUMES(*volser1...,volsern*) parameter.

  Enter 111111 in the first of these fields.

Panel DFQDCAG8

Figure 56 shows a representation of the ISMF DFQDCAG8 aggregate recovery rename level pairs panel.

```
   Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAG8                    AGGREGATE GROUP RECOVER              Page 4 of 7
COMMAND ===>

ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'
  TO RENAME SELECTED DATA SETS, SPECIFY LEVEL PAIRS (1 to 8 characters):

    OLD LEVEL ===> LEVEL0      NEW LEVEL ===> LEVEL0N
    OLD LEVEL ===> LEVEL1      NEW LEVEL ===> LEVEL1N
    OLD LEVEL ===> LEVEL2      NEW LEVEL ===> LEVEL2N
    OLD LEVEL ===> LEVEL3      NEW LEVEL ===> LEVEL3N
    OLD LEVEL ===> LEVEL4      NEW LEVEL ===> LEVEL4N
    OLD LEVEL ===> LEVEL5      NEW LEVEL ===> LEVEL5N
    OLD LEVEL ===> LEVEL6      NEW LEVEL ===> LEVEL6N
    OLD LEVEL ===> LEVEL7      NEW LEVEL ===> LEVEL7N
    OLD LEVEL ===> LEVEL8      NEW LEVEL ===> LEVEL8N
    OLD LEVEL ===> LEVEL9      NEW LEVEL ===> LEVEL9N
    OLD LEVEL ===> LEVELA      NEW LEVEL ===> LEVELAN
    OLD LEVEL ===> LEVELB      NEW LEVEL ===> LEVELBN
    OLD LEVEL ===> LEVELC      NEW LEVEL ===> LEVELCN
    OLD LEVEL ===> LEVELD      NEW LEVEL ===> LEVELDN
    OLD LEVEL ===> LEVELE      NEW LEVEL ===> LEVELEN

USE ENTER TO CONTINUE; USE DOWN TO SPECIFY ADDITIONAL LEVEL PAIRS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 56. Aggregate Recovery Panel DFQDCAG8

The following is a description of the fields of panel DFQDCAG8, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The OLD LEVEL and NEW LEVEL fields can each contain a valid 1 to 8 character data set name. Values other than blanks in these fields generate the RECOVERNEWNAMELEVEL(*olevel1,nlevel1 [,oleveln,nleveln...]*) parameter. At least one LEVEL pair must be specified. Each OLD LEVEL name specified must have a corresponding NEW LEVEL name. The OLD LEVEL name must differ from the corresponding NEW LEVEL name. Up to 15 pairs of names can be entered on this panel.

  Enter the LEVEL names as shown in Figure 56.

  A second panel, DFQDCAG9, with fields for 15 more name pairs, can be displayed by using the DOWN command.

Panel DFQDCAG9

Figure 57 shows a representation of the ISMF DFQDCAG9 aggregate recovery rename level pairs panel, which is a continuation of the ISMF DFQDCAG8 panel.

```
   Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAG9                    AGGREGATE GROUP RECOVER              Page 5 of 7
COMMAND ===>

ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'
  TO RENAME SELECTED DATA SETS, SPECIFY LEVEL PAIRS (1 to 8 characters):

    OLD LEVEL ===> LEVELF      NEW LEVEL ===> LEVELFN
    OLD LEVEL ===> LEVELG      NEW LEVEL ===> LEVELGN
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>
    OLD LEVEL ===>             NEW LEVEL ===>

USE ENTER TO CONTINUE; USE UP TO VIEW PREVIOUS LEVEL PAIRS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 57. Aggregate Recovery Panel DFQDCAG9

This panel is a continuation of panel DFQDCAG8, and the explanation of the fields is the same.

Enter the LEVEL names as shown in Figure 57.

Panel DFQDCAGA

Figure 58 shows a representation of the ISMF DFQDCAGA aggregate recovery conflict resolution options panel.

```
   Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAGA                    AGGREGATE GROUP RECOVER              Page 6 of 7
COMMAND ===>

ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'

  SPECIFY CONFLICT RESOLUTION OPTION:

    RENAME SOURCE NEW LEVEL ===>           (1 to 8 characters)
    RENAME TARGET NEW LEVEL ===>           (1 to 8 characters)
    REPLACE                 ===> N         (Y or N)
    BYPASS                  ===> Y         (Y or N)









USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 58. Aggregate Recovery Panel DFQDCAGA

The following is a description of the fields of panel DFQDCAGA, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The RENAME SOURCE NEW LEVEL field can contain a valid 1-to-8 character data set name that generates the RENAMESOURCE(*level*) subparameter of the DATASETCONFLICT parameter, indicating the specified level.

  Leave this field blank.
- The RENAME TARGET NEW LEVEL can contain a valid 1-to-8 character data set name that generates the RENAMETARGET(*level*) subparameter of the DATASETCONFLICT parameter, indicating the specified level.

  Leave this field blank.
- The REPLACE field, when Y is entered, generates the REPLACE subparameter of the DATASETCONFLICT parameter.

  Enter N in this field.
- The BYPASS field, when Y is entered, generates the BYPASS subparameter of the DATASETCONFLICT parameter.

  Enter Y in this field.

Note: The RENAME SOURCE NEW LEVEL, RENAME TARGET NEW LEVEL, REPLACE, and BYPASS fields are mutually exclusive: only one field can have an entry.

Panel DFQDCAGB

Figure 59 on page 391 shows a representation of the ISMF DFQDCAGB aggregate recovery submit or save panel.

```
   Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DFQDCAGB                    AGGREGATE GROUP RECOVER          Page 7 of 7
COMMAND ===>


DFSMSHSM COMMAND AND PROCESSING OPTION:
  ARECOVER DATASETNAME(PAY1.C.C01V0001) EXECUTE


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 59. Aggregate Recovery Panel DFQDCAGB

The following is a description of the fields of panel DFQDCAGB, and the entries that are made for the example system:

- The DFSMSHSM COMMAND AND PROCESSING OPTION field indicates the DFSMShsm command and processing option currently being processed. In the case of the example, it appears as ARECOVER DATASETNAME(PAY1.C.C01V0001) EXECUTE.
- The SELECT OPTION field allows one of two choices: entering 1 saves the generated ARECOVER parameters and submits the ARECOVER command to DFSMShsm, and entering 2 saves the generated ARECOVER parameters but does not submit the ARECOVER command.

  Enter 1 in this field.

Using the NetView File Transfer Program to Transmit Aggregate Backup Files

You can use the NetView FTP to create file transfer requests for aggregate backup files and submit them to NetView FTP for processing. Requests can be created either by entering values in the input fields of the NetView FTP panels or by writing and running a batch job. After submitting one request, you can immediately submit another; you do not have to wait until NetView FTP has processed the first request.

Requests are submitted to a component of NetView FTP called the *queue handler*. The queue handler assigns a request number to your request and adds it to the request queue, where it waits to be processed. The request queue is located on DASD.

After your request has been added to the request queue, you can DELETE it from the queue or QUERY the queue for information about a request.

Before you can use NetView FTP to transmit your backup files, you will need to process the aggregate backup command. The ABACKUP command can create up

to four data sets that should be transmitted to a remote location. The naming convention for these four data sets is as follows:

- outputdatasetprefix.D.CnnVnnnn

  This data set contains data from the data file that originally resided on level 0 DASD before it was backed.
- outputdatasetprefix.O.CnnVnnnn

  This data set contains data from the data file that was migrated or resided on user tape volumes before it was backed up.
- outputdatasetprefix.C.CnnVnnnn

  This data set contains the control information from the control file.
- outputdatasetprefix.I.CnnVnnnn

  This data set may contain valid instructions to the recovery site, or the ABACKUP activity log, if it was directed to DASD.

Note: Any ACCOMPANY data sets must also be transmitted.

Related reading: For additional information about NetView FTP, refer to the *Tivoli NetView File Transfer Program* publications.

Using NetView File Transfer Program Panels

You can use NetView FTP panels to create, save, change, add, query, and delete your transfer requests. Which panels NetView FTP displays and the order in which it displays them, depend on the task you select and on the data you enter in the input fields of the panels. The following sections contain examples of NetView FTP panels and examples of how to use the panels to transfer copies of aggregate backup files to a receiving location.

Invoking NetView FTP Panels

Invoke NetView FTP from your TSO selection menu. The Main Task Selection panel, which is shown in Figure 60, is displayed.

```
DVGSPEPE-------------- NetView FTP Main Task Selection -------------------------
OPTION ===> 1


Select an option by number

   1  Create         - Create a new file-transfer request
   2  Control        - Query, delete, modify, hold, release submitted requests
   3  List Saved     - List saved requests
   4  Change Saved   - Change a saved request
   5  Submit Saved   - Submit a saved request
   6  Administer     - Administer the request queue


Specify the name of the request for option 4 or 5
   Request Name ===>




Press the END PF key to return, HELP PF key for help
```

Figure 60. NetView FTP Main Task Selection Panel

Option 1 from the NetView FTP Main Task Selection panel is selected to show you how to create a file transfer request for the data file data set that contains the data backed up by using DFSMSdss.

Creating a Request

Each request is made up of *parameters* that tell NetView FTP everything it needs to know to transfer this file.

In the example system, the aggregate backup file PAY1.D.C01V0001 is transferred to the *Remote Node ID* TKSMVS1. The file is stored on the labeled tape volume serial number ABRT01. You want the data set to be received at the remote location under the data set name PAY1.D.C01V0001 and stored on a 3490 tape volume, serial number ABRT01. You have decided that a data set transferred from tape will be given a *request class* of 3. To protect the request, you enter the *request password* ROSE. The password does not appear on the screen when typed. You do not want NetView FTP to process the request before 6:00 p.m. today. You want NetView FTP to notify the sending user ID, X999999 at node XYZMVS4, and the receiving user ID, Z000000 at node TKSMVS1, that the data set has been transferred.

This is how you use the NetView Main Transfer Parameters panel, which is shown in Figure 61, to tell NetView FTP how to conduct this file transfer.

```
DVGIT20E-------------------- Main Transfer Parameters --------------------------
COMMAND ===>


Specify whether you want to transfer a file to or from a remote node
   Transfer Mode       ===> TO          (TO,FROM)


Specify either a remote node ID or a remote LU name
   Remote Node ID      ===> TKSMVS1
   Remote LU name      ===>

Specify a request class, priority, and password
   Request Class       ===> 3           (0 to 9; A to Z)
   Request Priority    ===> 0           (0 to 9)
   Request Password    ===>             optional

Specify a running mode for each server
   Sending   Server    ===> CONT        (CONT,SINGLE)
   Receiving Server    ===> CONT        (CONT,SINGLE)

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 61. Main Transfer Parameters Panel

After the Main Transfer Parameters panel is filled in, press the Enter key. The NetView FTP Main Data Set Parameters panel shown in Figure 62 on page 394 is displayed.

```
DVGIT1SE------------------- Main Data Set Parameters ---------------------------
COMMAND ===>


Sending Data Set (Local):

   Data Set Name ===> 'PAY1.D.C01V0001'
   DD Name       ===>
   Member Name   ===>               (blank=list, *=all)
   Data Set Type ===> LAB           (blank,VSAM,PO,PS,LAB,UNLAB,USER)
   Volume        ===>


Receiving Data Set (Remote):

   Data Set Name ===> 'PAY1.D.C01V0001'
   DD Name       ===>
   Member Name   ===>
   Data Set Type ===> LAB           (blank,VSAM,PO,PS,LAB,UNLAB,USER)
   Catalog Name  ===>

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 62. Main Data Set Parameters Panel

After the Main Data Set Parameters panel is filled in, press Enter. NetView FTP displays the Additional Transfer Parameters panel, where you enter the not-before time and the names of the report recipients as shown in Figure 63.

```
DVGID30E-------------- Additional Transfer Parameters --------------------------
COMMAND ===>


Not-before   Date ===>    /    /         (YY / MM / DD)
             Time ===> 18 : 00           (HH : MM)
Not-after    Date ===>    /    /         (YY / MM / DD)
             Time ===>    :              (HH : MM)

Restart from Checkpoint    ===> NO       (YES,NO)
Automatic Transfer Restart ===> YES      (YES,NO)
Compression Method         ===> ADAPT    (SNA,ADAPT,NONE)
Hold Request               ===> NO       (YES,NO)

Recipient of Report from Sending   Program ===> X999999  at XYZMVS4
Recipient of Report from Receiving Program ===> Z000000  at TKSMVS1

Local LU Name              ===>          LU name of local server to be used

Installation Exit Routine Input
===> tape check

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 63. Additional Transfer Parameters Panel

After you complete your entries on the Additional Transfer Parameters panel and press Enter, NetView FTP displays the Additional Sending Data Set Parameters panel, where you specify the device type, volume serial number, user ID (user ID is required), and security parameters as shown in Figure 64 on page 395.

You do not need to fill in these fields when transferring any of the data sets, because they are cataloged during aggregate backup processing.

```
DVGID2RE Additional Parameters for the Sending Data Set (PO,PS,LAB,UNLAB) ------
COMMAND ===>


DCB Parameters:
   Record Format         ===>                  (F,FB,V,VB,U,VS,VBS)
   Logical Record Length ===>
   Physical Block Size   ===>

Security parameters:
   User ID  ===> X999999
   Password ===>
   Group ID ===>

Specify values for the following if data set not cataloged:
   Device Type           ===> 3490
   Volume Serial Numbers ===> ABRT01    ==>           ==>
                         ===>           ==>           ==>



Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 64. Additional Parameters for the Sending Data Set Panel

Press Enter after you fill in the additional parameters for the Sending Data Set panel. NetView FTP displays the additional parameters for the Receiving Data Set panel, where you specify the disposition, device type, volume serial number, user ID (user ID is required), and security parameters for the receiving data set, as shown in Figure 65.

Note: If CAT is not entered in the disposition field, the data set must be manually cataloged at the receiving location.

```
DVGID2RE Additional Parameters for the Receiving Data Set (PS,LAB,UNLAB) --------
COMMAND ===>


Disposition              ===> CAT     (NEW,OLD,CAT)

Security Parameters:
   User ID               ===> Z000000
   Password              ===>
   Group ID              ===>

Specify values for the following if data set is not cataloged
   Device Type           ===> 3490
   Volume Serial Numbers ===> ABRT01  ==>         ==>
                         ===>         ==>         ==>









Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 65. Example 2: Additional Parameters for the Receiving Data Set Panel

After you fill in the additional parameters for the Receiving Data Set panel and press Enter, NetView FTP has all the input it needs to conduct the file transfer.

Using a Batch Job to Transmit Aggregate Backup Files

You can transmit your data files, without using NetView FTP panels, by creating a batch job as shown in Figure 66 on page 396:

```
//STEP0012 EXEC PGM=DVGIFBI                          INVOKE NFT
          //*
          //DVGLOG    DD SYSOUT=A
          //SYSABEND  DD DUMMY
          //SYSIN     DD *
          FUNCTION=ADD
          XMODE=TO
          REMNODE=TKSMVS1
          SNOTIFY=(X999999.XYZMVS4)
          RNOTIFY=(Z000000.TKSMVS1)
          SSECURP=(X999999.ROSE)
          RSECURP=(Z000000.CLIFF)
          WAIT=YES
          SFTYPE=LAB
          SFILEID='PAY1.D.C01V0001'
          RFTYPE=LAB
          RFILEID='PAY1.D.C01V0001'
          RTDISP=CAT
          /*
```

Figure 66. Sample Batch Job Used to Transmit Aggregate Backup Files

Attention: Because you cannot specify a file sequence number with the volume serial number when you are using NetView FTP or transmitting a batch job, you must catalog the data set if you need to transmit a second file of the data file.

Related reading: For additional information about how to use the NetView FTP panels, see the *NetView File Transfer Program for MVS: Release 1.0 User's Guide*.

Issuing the ARECOVER Command When Using NetView FTP

When you use NetView FTP to transmit aggregate backup files, the following steps must be taken before you issue the ARECOVER command:

1. Ensure that all control file and data file data sets and any accompany data sets are cataloged before issuing the ARECOVER command.
2. Use the XMIT parameter on the ARECOVER command to indicate you are recovering data sets that have been transferred using NetView FTP.
3. *Do not* specify the UNIT and VOLUME parameters when XMIT is specified on the ARECOVER command, as shown in this example:

   ```
   ARECOVER PAY1.C.C01V0001 REPLACE XMIT
   ```

4. Do not use the AGGREGATE parameter with the XMIT parameter.

ABARS Considerations for Sequential Data Striping

DFSMShsm does aggregate back up and aggregate recover processing for single-stripe and multiple-stripe sequential extended format data sets. To process these data sets, the data mover must be DFSMSdss, and the data sets must be SMS-managed.

Aggregate Backup Processing

DFSMShsm ABACKUP processing invokes DFSMSdss to perform a logical data set dump when it encounters a sequential extended format data set—both single-stripe and multiple-stripe—in the INCLUDE list.

If a sequential extended format data set resides on a non-SMS-managed volume, the logical data set dump fails.

Aggregate Recovery Processing

DFSMShsm invokes dynamic allocation (SVC 99) to allocate the data sets in the ALLOCATE list:

- During ARECOVER processing, dynamic allocation allocates the data set as sequential extended format or not sequential extended format (regardless of the number of stripes at the backup site) according to what was specified at the recovery site for the DATA SET NAME TYPE attribute in the data class.
- Dynamic allocation also allocates the data set as single-stripe or multiple-stripe sequential extended format (regardless of the number of stripes at the backup site) according to what was specified at the recovery site for the SUSTAINED DATA RATE attribute in the storage class. If GUARANTEED SPACE = YES is specified, the number of volsers or VOLCNTs specified is the number of stripes; the value for SUSTAINED DATA RATE is ignored.
- Dynamic allocation allocates the data set as a data set that is not sequential extended format if the attribute DATA SET NAME TYPE in the data class is defined as (EXTENDED,C), and one of the following conditions is met:
  - SMS is not active at the recovery site.
  - None of the SMS-managed volumes is defined.
  - None of the SMS-managed volumes is connected to an ESCON 3990 model 3.
  - None of required volumes are available.
- Dynamic allocation does *not* allocate a data set if the attribute DATA SET NAME TYPE in the data class is specified as (EXTENDED,R), and one of the following conditions is met:
  - SMS is not active at the recovery site.
  - None of the SMS-managed volumes is defined.
  - None of the SMS-managed volumes is connected to an ESCON 3990 model 3.
  - Insufficient volumes are available to meet the allocation.
  - Insufficient space is available on the selected volumes.

  DFSMSdss performs a logical data set restore for a sequential extended format data set as follows:
  - A sequential extended format data set that was multiple-stripe when it was backed up can be restored as a multiple-stripe or single-stripe sequential extended format data set, depending on the storage class SUSTAINED DATA RATE value, and the availability of the required volumes.
  - Conversely, a sequential extended format data set that was single-stripe when it was backed up can be restored as a multiple-stripe or single-stripe sequential extended format data set, depending on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.
  - A data set that was sequential extended format when backed up is recovered as a non sequential extended format but still SMS-managed data set if no SMS volumes supporting sequential extended format data sets are available.
  - A data set that was not sequential extended format when it was backed up is always recovered as a data set that is not sequential extended format. DFSMShsm will not convert a data set that is not sequential extended format to a sequential extended format data set through ARECOVER processing.

  If the data class attribute DATA SET NAME TYPE is modified to indicate that the data set should be allocated as a data set that is not sequential extended format, the restore does not occur.

Aggregate Roll-Off Processing

When a new version of an aggregate group is created, ABARS processing ensures that the number of versions kept does not exceed the amount specified by the NUMBER OF VERSIONS attribute in the management class. If NUMBER OF VERSIONS is exceeded, ABARS rolls off (deletes) the oldest version of that aggregate.

Roll-off processing can occur each time a new ABR record is written to the BCDS. Since ARECOVER creates an ABR record if one does not exist, roll-off processing can occur there as well as during ABACKUP.

Roll-off processing does not occur when the VERIFY parameter is used, because no ABR record is created.

When an aggregate group is removed during roll-off processing, its control file name is located in the catalog and a list of volume serial numbers that the control file resides on is requested. When necessary, the volumes are deleted from the RACF HSMABR tape volume list. The control file is uncataloged after the volume serial number list has been exhausted. When the file does not exist in the catalog, it is not an error condition, and the ABR record can still be deleted.

If it is active, installation exit ARCTVEXT is called for each volume being processed, indicating to the exit that the volume being expired is an ABARS tape volume.

This process is repeated for the data files and the instruction and activity log files.

After the control file, data files, and instruction and activity log file have been uncataloged and removed from the HSMABR tape volume set, the ABR record is deleted and processing continues with the next ABR record for the group being processed.

Note: It is possible to code SMS ACS routines that direct ABARS files to DASD. If this is done, and the files get migrated to DASD, an HSM DELETE is issued to remove them during roll-off processing. If that delete fails, the file may remain cataloged with no ABR record pointing to it. If this happens, the data set must be manually deleted.

You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether or not you want DFSMShsm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs. The default is N, which specifies no automatic deletion.

Defining GDGs at the Recovery Site

An alternative way to have GDGs defined at the recovery site is to specify a GDG base name in the ALLOCATE statement and have ABARS predefine the GDG base name during ARECOVER processing. This can be done even if the installation has not specified an associated GDS name in the selection data set. During ARECOVER processing, all GDG base names are defined before restoring any GDSs specified in the aggregate. GDG base names specified in the ALLOCATE statement have the following restrictions:

- They are only defined if they do not already exist at the recovery site.

- The DSCONFLICT parameter of the ARECOVER command does **not** apply to these GDG base names.
- The ARCCREXT installation exit does **not** get control when active for conflicts that arise with GDG base names in the ALLOCATE statement.
- RECOVERNEWNAMELEVEL and RECOVERNEWNAMEALL **do** apply to these GDG base names.

Ensuring Backup Copies of ARECOVERed SMS-Managed Data Sets Exist at the Recovery Site

When ABARS is used to move data sets to a recovery site, DFSMSdss moves the catalog information for all data sets residing on primary volumes from the disaster site to the recovery site. If backup copies of these data sets are not also moved to the recovery site, the LBACKUP dates in the catalog are not correct when the data sets are ARECOVERed. If these data sets are not updated, then no backup is created for SMS-managed data sets because the LBACKUP dates in the catalog show that valid backup copies exist at the recovery site when, in fact, they do not. This creates a potential data integrity problem if the original data sets are ever lost or damaged.

Two methods can be used to ensure that backup copies of these data sets are made:

- Issue a BACKVOL TOTAL for ARECOVERed volumes. This creates backup copies unless the management class specifies that no backups should be made.
- Use CLIST to allow the ARECOVER activity log to process the data sets, opening them for output so that the change bit is set.

Chapter 9. Making Disaster Backup Copies of DFSMShsm-Owned Tape Volumes

The ABACKUP and ARECOVER functions described in Chapter 8, "Aggregate Backup and Recovery Support (ABARS)," on page 281 provide comprehensive recovery of your data in case of a disaster.

However, you also need protection against damage or loss of your backup and ML2 tape volumes. DFSMShsm allows you to make copies of your backup and ML2, single-file-format tape volumes for storage at a site different from the computer site or for use at the computer site to recover lost or damaged tape volumes.

In addition to considering tape-capacity usage, you should consider the tape-related implications of disaster backup described below, in deciding whether you want to handle partially full tapes by SETSYS PARTIALTAPE(MARKFULL) or SETSYS PARTIALTAPE(REUSE).

Related reading: For additional information about the SETSYS commands, see the *z/OS DFSMShsm Storage Administration Reference*.

Planning Disaster Recovery

The premise for disaster recovery is that you will have copies of all your vital data sets stored at a location separate from your computer site and that a computer system of approximately the same capabilities will be available to be used to resume operations. Consider what this implies:

- You must have a copy of all your system control data sets stored at the off-site location. System control data sets change everytime maintenance is applied, which for some installations is weekly or biweekly, so having automated scheduled backups are recommended. You can also dump the data sets by command whenever necessary.
- You must have copies of all your vital user data sets. You can use SMS to ensure that all of the vital user data sets are stored on the appropriate DASD volumes. You can provide these copies by using the DFSMShsm ABACKUP function.
- You must have copies of your ML1 volumes in order to recall any vital data sets that have migrated to ML1. You can use the DFSMShsm ABACKUP or automatic dump functions to provide these copies of DASD ML1 volumes.
- You must have copies of your ML2 volumes in order to recall any vital data sets that have migrated to ML2. If you use DASD for ML2, you can provide these copies by using the DFSMShsm ABACKUP or command dump functions. If you use cartridge-type, single-file-format tape for ML2, you can use the duplex tape option when creating the ML2 tapes, or you can use other DFSMShsm disaster backup facilities as discussed in this chapter. ABACKUP can be used for ML2 tape data sets as well.
- You should have copies of your backup volumes in order to reconstruct your user volumes. If you make copies of your backup volumes, you can make fewer automatic dump copies of entire user volumes. You reconstruct the user volumes from the combination of the latest dump copy and the incremental backup versions of data sets made after the dump copy was made. If you use DASD for backup, you can provide these copies by using the DFSMShsm ABACKUP or

© Copyright IBM Corp. 1984, 2004

command dump functions. If you use cartridge-type, single-file-format tape for backup, you can use the duplex tape option when creating backups, or you can use other DFSMShsm disaster backup facilities as discussed in this chapter.

Note: ABACKUP cannot be used for incremental versions stored on backup DASD. Only full volume dumps can provide offsite copies of data on backup DASD.

- You must have VTOC copy data sets for some cases. The VTOC copy data sets are required when you are restoring data from dump copies combined with incremental backup versions. By dumping ML1 volumes you have VTOC copy data sets available at the recovery site.
- You need a set of instructions for reconstructing the system and resuming operations.

Disaster Backup Facilities

To provide disaster backup, two things are necessary: (1) a way to make copies of DFSMShsm-owned tape volumes and to identify the copied volumes to DFSMShsm; and (2) 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.

Making Copies of DFSMShsm-Owned Tape Volumes

There are two methods of making copies of DFSMShsm-owned tape volumes. One method is by using the duplex tape option. The other method is by using the TAPECOPY command.

The Duplex Tape Option

The duplex tape option provides an alternative to TAPECOPY processing for backup and migration cartridge-type tapes. Two tapes are created concurrently with one designated the original, the other the alternate. The intent is that the original tape be kept onsite, while the alternate can either be created in a remote tape library, or taken offsite.

The alternate tape must have the same tape geometry as the original. For example, if the original tape is a 3590-1 tape, the alternate must be also.

The duplex tape option also requires one additional tape drive per writing task. Adequate numbers of drives must be available for the tasking level used by the functions using duplexing during the time periods those functions run. This includes recycle.

With the SETSYS DUPLEX command you can specify that the duplex tape option be used for backup tapes only, migration tapes only, or both backup and migration tapes. If you use duplex creation for both migration and backup and you have been auto scheduling tape copying, you probably want to remove the auto scheduling. If you are duplexing either migration or backup tapes but not both, you can auto schedule tape copies based on the type indicated in the ARC0421I message.

In an SMS environment, the ACS routines can direct the alternate tape to the same or to a different tape library. Different tape libraries can exist at the same location, or a remote location. ESCON technology defines how distant the remote location

can be. In a non-SMS environment, the output restrictor is used for both the original and the alternate. ACS filtering can be set up to cause the desired tape allocations to occur.

Tapes created by the duplex tape option are compatible with tapes created by TAPECOPY.

DFSMShsm initial tape selection selects tapes for the duplex option using the following selection process:

- A partial tape and its alternate are selected.
- If no partial tapes with alternates are found, an empty ADDVOLed volume with a scratch volume as an alternate are selected.
- If there are no empty ADDVOLed volumes, two scratch volumes are selected.

When problems occur that allow an original tape to be created, but prevent the alternate from being created by the duplex tape option, DFSMShsm automatically schedules internal TAPECOPY processing to ensure that valid alternate tapes exist. If the original tape is demounted before being filled it is marked as duplexed but unavailable for output until the alternate is successfully created. At either the EOV for the original volume or the end of the task, DFSMShsm schedules TAPECOPY processing to create an alternate tape. If that is unsuccessful, automatic scheduling of internal TAPECOPY processing occurs during secondary space management for migration volumes, and during automatic backup processing on the primary host for backup volumes.

As part of the automatic internal scheduling of TAPECOPY processing, a tape copy needed (TCN) record is written in the OCDS. When the automatic internal TAPECOPY completes successfully, the TCN record for the source volume (which is the original volume of the duplexed pair) is deleted, and the original tape volume is made available for reuse.

In a SETSYS PARTIALTAPE(REUSE) environment, both the duplexed original and its alternate can be reused (extended). If you have a non-duplexed original with an alternate created by TAPECOPY, only the original can be reused. The alternate is discarded.

A new flag has been added to the ARCTVEXT installation exit parameter list. When an alternate tape is being replaced and returned to scratch, and if the ARCTVEXT installation exit is active, the flag indicates if the action is the result of a user-issued TAPECOPY command. This information is provided so users are aware that an alternate tape is being returned to scratch, and that the original tape is not.

You can use the LIST and HLIST commands to show duplex tape status and alternate tape volume serial numbers.

Duplex tape does not support SETSYS TAPEUTILIZATION NOLIMIT set for the output unit being duplexed. This is because the alternate tape must match the original tape exactly, and with SETSYS TAPEUTILIZATION set to NOLIMIT this is impossible to guarantee, since either tape could be longer than the other. When TAPEUTILIZATION is set to NOLIMIT, the duplex tape option causes the tapes to reach a logical EOV when they are 97% full.

The duplex tape option does not support creating duplex copies of ML2 tapes during ARECOVER processing. Instead, one copy of the ML2 tape is recovered by

AURECOVER processing, and a TCN record is written in the OCDS. The TCN record causes a TAPECOPY to be scheduled during the next occurrence of secondary space management processing. The TAPECOPY is not scheduled immediately because a TAPECOPY running during the recovery process might not be wanted.

The duplex tape option handles errors in the following way:

- When an attempt is made to allocate and mount a partially full duplexed tape, but the associated alternate volume cannot be found, both the original and alternate are rejected and marked as unavailable for the duration of DFSMShsm processing.
- When an attempt is made to allocate and mount an empty original tape and a new alternate scratch tape, but the scratch tape cannot be allocated or mounted, the duplex tape option continues to write the original volume. When the original volume (ADDVOLed or scratch) is demounted, DFSMShsm marks the volume as duplexed and unavailable for output, and schedules an internal TAPECOPY for that volume. DFSMShsm makes the original volume available after internal TAPECOPY processing successfully creates an alternate tape volume.
- If the duplexed original encounters an I/O error, both the original and alternate tapes are marked full, demounted, and two different tapes are selected. The user data set related to the I/O error is restarted from the beginning on the different tapes.
- If the alternate tape volume encounters an I/O error, the alternate is be demounted and returned to scratch, but the original continues to be written. The original is marked as duplexed and unavailable for output, and an internal TAPECOPY is scheduled to automatically create an alternate tape. DFSMShsm makes the original volume available after internal TAPECOPY processing successfully creates an alternate tape volume.
- If RECYCLE processing writes a data set to the duplex tapes, and the alternate tape volume encounters natural EOV before the data set has been completely written to the tape, the alternate is demounted and returned to scratch, but the original continues to be written. The original is marked as duplexed and unavailable for output, and an automatic internal TAPECOPY is scheduled by DFSMShsm to create an alternate tape. DFSMShsm makes the original volume available after internal TAPECOPY processing successfully creates an alternate tape volume.
- When the original tape reaches its percent-full capacity, DFSMShsm must flush all the data to both tapes before performing a FEOV. If natural EOV is reached on either tape, DFSMShsm marks the tapes as *FULL* and restarts processing of the current data set from its beginning on two new tapes.

Systems running DFSMS/MVS releases prior to DFSMS/MVS V1R4 must not attempt to extend a backup or migration tape created in duplex. A PTF is available to prevent attempts to extend these data sets.

The TAPECOPY Command

The TAPECOPY command allows you to make copies of single-file-format, DFSMShsm-owned tapes. The copies are known as alternate tapes and are identified in the offline control data set (OCDS) as alternates to the tapes that are copied.

DFSMShsm provides commands to copy categories of DFSMShsm-owned tape volumes:

TAPECOPY BACKUP

TAPECOPY MIGRATIONLEVEL2

TAPECOPY ALL

These commands cause DFSMShsm to make alternate copies of full, single-file-format tapes that do not already have an alternate copy. The TAPECOPY MIGRATIONLEVEL2 and TAPECOPY ALL commands also cause DFSMShsm to make alternate copies of partially full ML2 tapes if the environment is PARTIALTAPE(MIGRATION(MARKFULL)).

You can make copies of all volumes categorized as either ML2, backup (or both), or of individual tape volumes. You can issue one TAPECOPY command per DFSMShsm host as long as the requests do not overlap. For example, you can issue a TAPECOPY ML2 command on one host and a TAPECOPY BACKUP command on another.

For individual tape volumes, you can make copies of either partial or full volumes, and of volumes with or without an alternate copy already existing.

The TAPECOPY command provides for copying only cartridge-type, single-file-format tapes. If you have data on multi-file format tapes (such as reel tapes), you can transfer the data onto new cartridge tapes by using the recycle function. These cartridge tapes can then be copied, and use of the reel tapes can be discontinued.

The output or alternate tape volumes used by the tape copy process begin as scratch tape volumes. These scratch tape volumes cannot be volumes that have been added to DFSMShsm with the ADDVOL command. Using scratch tape volumes not added to DFSMShsm eliminates backup, dump, or ML2 tape volumes from being used as alternate tape volumes.

DFSMShsm copies the security information (RACF and expiration date) of the original volume, or you can specify a different expiration date when an alternate volume is created. Because password protection is not copied to an alternate volume, you should use RACF or an expiration-date protection for the alternate volume. If the alternate volume is in the RACF DFSMShsm tape volume set, the RACF protection is removed from the alternate volume when it is returned to scratch status. If the tape resource class (TAPEVOL) is inactive during a recycle request for a RACF-protected original volume with an alternate, the RACF protection for the alternate will be deleted based on the TTOC information available for the original volume.

Information about the alternate tape volumes is stored in the tape table of contents (TTOC) records of the OCDS. While, over time, a given tape volume might be copied to several alternate volumes, only the most current alternate tape volume reference is stored in the TTOC record.

Attention: Be aware that some tape management products do not protect DFSMShsm data tapes from being mounted for output in response to a nonspecific mount PRIVAT request, and therefore they can allow DFSMShsm to write over some of its own tapes. This exposes alternate tapes because DFSMShsm does not maintain a way to quickly identify an alternate tape as a DFSMShsm tape, so there is no check to prevent an alternate from being overwritten. DFSMSrmm does not have this exposure.

Assigning expiration dates to tapes provides only partial protection from inadvertent rewriting over valid data. The general solution is to use policies defined with a function like "vital record specifications" in a tape management product like DFSMSrmm . DFSMShsm uses tape-volume exit ARCTVEXT to communicate to tape management products, except DFSMSrmm, when a tape can safely be returned to scratch status.

Related reading: For additional information about using DFSMShsm with DFSMSrmm, see Chapter 16, "Using DFSMShsm with DFSMSrmm," on page 537.

Copying Individual Volumes

You may want to make copies of individual volumes under the following conditions:

- Message ARC0421I, written to the console when a tape volume is marked full, allows making an alternate copy of that volume automatically, if you are not using the duplex tape option to create alternate tapes.
- You have replaced an original volume with an alternate volume. You would then immediately want to make a new alternate volume for protection against loss of what is now the only usable volume.
- An alternate volume is lost so that you must make a new alternate volume, even though it appears (in a TTOC record) that an alternate volume already exists.

When DFSMShsm makes an alternate volume, it records the alternate volume in the OCDS, thus associating the alternate volume with the original volume. For copying individual volumes, if an alternate volume already exists:

- DFSMShsm replaces the old alternate volume with the new alternate volume.
- DFSMShsm issues message ARC0436I.
- No record is kept of the old alternate volume.
- For duplex tapes only, the tape volume exit ARCTVEXT is invoked, if active, to release the previous alternate tape volume through the tape management system. A flag in the ARCTVEXT parameter list indicates that the tape being released is an alternate tape being replaced by a user TAPECOPY, and that the original tape is not being released.
- If SETSYS PARTIALTAPE(MARKFULL) is specified, the original volume is marked full.

The command to make alternate volumes for individual volumes is:

TAPECOPY ORIGINALVOLUMES(*ovol1*,*ovol2*,...,*ovoln*)

TAPECOPY ORIGINALVOLUMES(*ovol1*,*ovol2*,...,*ovoln*) -
ALTERNATEVOLUMES(*avol1*,*avol2*,...,*avoln*)

This command copies the volumes specified, creating an alternate tape volume for each. 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 be mounted for each alternate tape volume. However, if you use the second form of the command, DFSMShsm uses the volumes you specify with the ALTERNATEVOLUMES parameter to become the alternate volumes. If you use the ALTERNATEVOLUMES parameter, you must specify exactly the same number of volumes with the ALTERNATEVOLUMES parameter as you specified with the ORIGINALVOLUMES parameter. There is a one-to-one correspondence between the original volumes and alternate volumes that you specify. That is, *ovol1* is copied to *avol1*, *ovol2* is copied to *avol2*, and so forth.

An alternative exists for copying your existing DFSMShsm-owned volumes. You can issue a LIST command to obtain a list of your ML2 and backup volumes. You can then issue a LIST TTOC command to determine which of the volumes have alternate copies already. Send the LIST TTOC command output to a data set. Edit the data set to remove those volumes for which you do not want to make an alternate copy, convert the data set to the format required for the input data set, and issue the following command:

TAPECOPY INDATASET(*volcopy.list.dsname*)

DFSMShsm makes an alternate copy of each of the volumes contained in the list of volumes provided by the data set.

Related reading:

- For additional information about using the LIST command, see Chapter 14, "Obtaining Information from DFSMShsm," on page 517.
- For additional information about the TAPECOPY command, see the *z/OS DFSMShsm Storage Administration Reference*.

Converting Partial Tapes to Full Tapes

Because DFSMShsm automatically processes TAPECOPY requests for tape volumes marked as full, and because partial tapes without alternates are not selected for tape duplexing, it would be advantageous not to have any partially full tapes in the DFSMShsm inventory. There are three methods for removing all partially full tapes from the DFSMShsm inventory:

- Specify the SETSYS PARTIALTAPE(MARKFULL), PARTIALTAPE(MIGRATION(MARKFULL)), or PARTIALTAPE(BACKUP(MARKFULL)) parameter. This causes DFSMShsm to mark the appropriate tape volumes as full when they are only partially full at the end of a task (except at the end of a data set migration task).
- Because DFSMShsm always tries to start a task with a partially full tape, all the partially full tapes are used or marked as full after a few migration and backup cycles.
- Use the DELVOL BACKUP(MARKFULL) or MIGRATION(MARKFULL) command to mark each partially full tape.

Replacing an Original with an Alternate Tape Volume

If an original ML2 or backup volume is lost or damaged, the volume can be replaced with its alternate volume:

1. To determine if an alternate tape volume exists for the original volume, use the LIST TTOC(*volser*) command.
2. Issue the TAPEREPL ORIGINALVOLUMES(*ovol*) command to replace the original volume with the alternate. After the command completes, the alternate tape volume reference in the TTOC record replaces the original volume reference.
3. Issue the TAPECOPY ORIGINALVOLUMES(*newovol*) command to make a new copy of the volume (using the volume serial number of the old alternate volume) to provide a new alternate volume.

When the TAPEREPL command successfully replaces a tape volume, DFSMShsm issues the ARC0427I message and (with two exceptions) invokes the tape-volume exit ARCTVEXT to indicate to a tape management product that the original volume can be returned to scratch. The two exceptions are when you are using

TAPEREPL with the DISASTERALTERNATEVOLUME optional subparameter, or when your tape management product is DFSMSrmm.

For duplex tapes, after the original volume is replaced, it is marked full and is no longer a duplex tape.

Creating Duplex Tapes Off Site

With the duplex tape option, you can write the alternate volumes directly to a remote tape library. ESCON technology defines how distant the remote location can be.

When you process the RECYCLE command, DFSMShsm invalidates the alternate tapes at the same time that it invalidates the tapes from which they were copied. Invalidated alternate tapes are returned to the scratch pool at their respective sites.

Sending Alternate Tapes Off Site

To use alternate tapes for disaster backup by sending them to a safe site, you must determine what tapes to send away and determine when to bring the tapes back and reuse them. Use the following procedure to send alternate tapes offsite:

1. Issue the TAPECOPY command for your non-duplexed tapes (backup, migration, or all, as you desire).
2. Issue LIST TTOC to determine which volumes have alternate copies.
3. Send the alternate tapes off site.
4. When you process the RECYCLE command, DFSMShsm invalidates the alternate tapes at the same time that it invalidates the tapes from which they were copied. Both the original and the alternate tape are passed to the ARCTVEXT installation-wide exit.
5. Request the return from the offsite location of the alternate tapes that have been invalidated, and return the tapes to the scratch pool.

Managing Alternate Volumes

DFSMShsm provides full support for producing and managing one alternate tape for each volume in the categories of ML2 and backup tapes. As shown above, this includes using the duplex tape option, or using a generic TAPECOPY command to copy full tapes not already having a copy, and optionally interfacing to the DFSMShsm installation exit ARCTVEXT to return emptied tapes to scratch status.

Examples: Although it is possible for you to make multiple alternate tapes, that comes with the price of doing additional management of them, as detailed in the following two examples:

1. If you issue the TAPECOPY command to create an alternate tape for an original volume not marked full, then the original will likely be extended later.

   When entries are being added to a base TTOC or when a tape is marked full for reasons other than a DELVOL MARKFULL command, any existing alternate volume is no longer valid. If, in either case, the tape already has an alternate volume, then message ARC0441I is issued to the console and appropriate activity log, and the existing alternate is removed. DFSMShsm does not take exit ARCTVEXT, unless it was a duplexed alternate, since it is not known whether the alternate tape should be returned to scratch status. If it was a duplexed alternate, DFSMShsm takes the ARCTVEXT installation exit, doesn't get rid of the alternate, and doesn't issue the ARC0441I message.

2. If you issue a specific TAPEREPL command specifying the alternate volume to replace the original tape, and the specified alternate is different from the alternate volume DFSMShsm knows about, then message ARC0439I is issued to the submitter and the command activity log. The alternate previously known to DFSMShsm is not retained. It is also not passed to the ARCTVEXT exit (although the original is) because DFSMShsm does not know whether you wish it to be returned to scratch status.

For duplex tapes, after the original volume is replaced, it is marked full and is no longer a duplex tape.

The first example above is an instance where DFSMShsm directs a specific message to the console so that if automatic operations are desired, they can be implemented based on intercepting a console message.

If an ML2 or backup tape is deleted using the DELVOL command, DELVOL processing issues message ARC0420I for the alternate volume recorded by DFSMShsm.

Disaster Alternate Volumes

DFSMShsm provides a way—once you have created alternate tapes or have duplexed alternate tapes—to use those alternates in place of the original tapes for doing recalls or recovers.

If an actual disaster happens at your home site or if you are running a disaster recovery test, you should proceed as follows to establish operations at your recovery site:

1. Bring up the system by restoring the system packs, libraries, and DFSMShsm CDSs.
2. If your strategy includes dump tapes, set the DEFINE command at the recovery site to limit DFSMShsm recovery to those dump classes that are present there. It is likely that the CDS indicates the dump classes at the recovery site are those available at the home site.
3. Start DFSMShsm, and restore the required data sets from DASD.
4. If you have created alternate copies of DFSMShsm-owned tape volumes and the alternate volumes are present at the recovery site, run the TAPEREPL command to convert all of DFSMShsm's information about the original volume serial number to the alternate volume serial number. Since this process is very time-consuming, a significantly reduced form of the TAPEREPL command can be run during the earliest stages of bringing up a recovery site. This command consists of TAPEREPL, followed by required parameters, followed by DISASTERALTERNATEVOLUMES, abbreviated to DAVOLUMES. For example:

   TAPEREPL ALL DAVOLUMES

   When the DAVOLUMES parameter is used, only base OCDS TTOC records of ML2 and backup tapes that have alternate volumes are updated as having *disaster alternate volumes*.

   The designation of "disaster alternate volumes" can help distinguish between alternate volumes initially present at the recovery site, and alternate volumes created later at the recovery site.

Note: When the DAVOLUMES option is used, DFSMShsm does *not* invoke the tape-volume exit ARCTVEXT, since the original tape is not being totally replaced. For example, if this is done for a disaster recovery test, the resulting control data sets may be discarded after the test, and the original tapes again used at the home site.

DFSMShsm invokes ARCTVEXT when the complete replacement is done, using TAPEREPL without the DAVOLUMES option.

5. Before running customer jobs, issue the SETSYS command with the DISASTERMODE parameter. For example:

   SETSYS DISASTERMODE

   One approach would be to put the SETSYS DISASTERMODE command in a startup ARCCMDxx PARMLIB member that is used exclusively for starting DFSMShsm in disaster mode.

   While DFSMShsm is running in disaster mode, recalls and recovers from tape volumes that are marked as having disaster alternate volumes will substitute the volume serial number of the disaster alternate volume from the TTOC record for the original volume serial number that appears in the data set records.

6. If you use TAPEDELETION(HSMTAPE) or PARTIALTAPE(REUSE) options then the inventory will likely contain empty and partially filled backup and migration tapes that were destroyed at the home site. These tapes will be selected for output at the recovery site. To prevent DFSMShsm from requesting tapes that are not available, identify these tapes using the LIST TTOC SELECT(NOTFULL) command, then mark the tapes full with the MARKFULL option of the DELVOL command:

   DELVOL *volser* **BACKUP(MARKFULL)**
   DELVOL *volser* **MIGRATION(MARKFULL)**

7. As jobs run at the recovery site and new data sets are created, backup versions and migration copies are created. As tapes are filled, they may be copied. These “new” alternate copies are not considered “disaster alternate volumes” and will not be substituted for the original volumes during recall or recover.
8. If you have finished running a disaster recovery test or you are finished using the alternates, reset the Disaster Alternate Volume flag in the TTOCs by issuing the following command:

   TAPEREPL ONLYDISASTERALTERNATES(RESET)

   If you were performing a disaster recovery test, you may want to make scratch tapes out of any DFSMShsm output tapes created at the recovery site. You may leave the alternate volumes for future disaster recovery.

9. If you are recovering from a true disaster, you now need to plan for the return to your home site. You need to convert the disaster alternate volumes to original volumes. You can issue

   TAPEREPL ONLYDISASTERALTERNATES

   to replace all, and only, those original volumes identified as having disaster alternate volumes. The command HOLD TAPEREPL may be used if the TAPEREPL command needs to be stopped.

10. You can also replace specific original volumes with their disaster alternates. First, issue

```
LIST TTOC SELECT(DISASTERALTERNATEVOLUMES)
```

to identify those tape volumes with disaster alternate volumes, and then issue the TAPEREPL command without the DAVOLUMES parameter. For example, the command

```
TAPEREPL OVOLS(TKS001,TKS002,TKS003)
```

—where TKS001, TKS002, and TKS003 are volume serial numbers that were among those returned from the LIST TTOC command—will replace these specific original volumes with their disaster alternates.

These newly converted original tapes need to be copied now for future disaster recovery.

Note: The AUDIT MEDIACONTROLS processing will fail the audit of a tape volume that is marked as having a disaster alternate volume.

Related reading:

- For additional information about the TAPEREPL command, see the *z/OS DFSMShsm Storage Administration Reference*.
- For additional information about the DELVOL command, see the *z/OS DFSMShsm Storage Administration Reference*.

Considerations When Using Disaster Alternate Volumes

When doing a generic RECYCLE or TAPECOPY, a tape volume marked as having a disaster alternate volume that is otherwise selected is skipped. A specific volume RECYCLE or TAPECOPY for such a volume fails; message ARC0424I is issued for a TAPECOPY failure and message ARC0445I is issued for a RECYCLE failure.

Chapter 10. Other Considerations for Availability Management

The following additional topics are related to each of the aspects of availability management: backup, dump, recover, and restore.

Additional Considerations for Backup

Additional topics related to backup are:

- Possible tape volume contention in single and multiple DFSMShsm-host environments
- Recover tape takeaway contention
- Implications of redefining the backup cycle
- Inline backup of data sets in the middle of a job
- Preventing backup processing
- GUARANTEED SPACE attribute and VSAM data sets

Backup Volume Contention Resolution

DFSMShsm resolves volume contention among the different tasks that can request the same tape volume. The data set backup by command function may hold a tape for an extended period of time. Consequently, DFSMShsm allows a backup tape to be taken away from a backup or recycle task to satisfy a recover request. Table 24 summarizes the contention possibilities and scope of what DFSMShsm does to resolve volume contention for both single and multiple DFSMShsm-host environments.

Table 24. Volume Contention in Single and Multiple DFSMShsm Host Environments

Function That Has Volume	Function That Needs Volume	Result
Backup	Backup	Backup selects a new target volume.
Recover	Backup	Backup selects a new target volume.
Recycle (source)	Backup	Backup selects a new target volume.
Recycle (target)	Backup	Backup selects a new target volume.
Tapecopy (source)	Backup	Backup selects a new target volume
Backup (data set)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. Data set backup completes any data set that is currently being processed, releases the tape, and selects a different target tape.
Backup (volume)	Recover	Recover fails.
Recover	Recover	In a single DFSMShsm-host environment, there is no contention. In a multiple DFSMShsm-host environment, the Recover fails.

© Copyright IBM Corp. 1984, 2004

Table 24. Volume Contention in Single and Multiple DFSMShsm Host Environments (continued)

Function That Has Volume	Function That Needs Volume	Result
Recycle (source)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. When recycle processing is complete for the data set that is being processed, the allocated volume is released and a new volume is selected for recycle processing.
Recycle (target)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. When recycle processing is complete for the data set that is being processed, the allocated volume is released and a new volume is selected for recycle processing.
Tapecopy (source)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. If tapecopy does not complete before the maximum time, tapecopy fails. Either way, recover takes the tape.
Backup	Recycle source	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recover	Recycle source	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recycle (source)	Recycle source	In a single DFSMShsm-host environment, there is no contention. In a multiple DFSMShsm-host environment, Recycle skips the volume because another host is recycling it.
Recycle (target)	Recycle source	In a single DFSMShsm-host environment, Recycle selects a new target volume. In a multiple DFSMShsm-host environment, Recycle skips the volume, because another host is recycling it.
Tapecopy (source)	Recycle source	In a single DFSMShsm-host environment, Recycle skips the volumes, and recycle of other volumes continues. In a multiple DFSMShsm-host environment, Recycle skips the volumes, and recycle of other volumes continues.
Backup	Recycle target	Recycle selects a new target volume.
Recover	Recycle target	Recycle selects a new target volume.
Recycle (source)	Recycle target	Recycle selects a new target volume.
Recycle (target)	Recycle target	In a single DFSMShsm-host environment, there is no contention. In a multiple DFSMShsm-host environment, Recycle selects a new target volume.
Tapecopy (source)	Recycle target	Recycle selects a new target volume.
Backup	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.

Table 24. Volume Contention in Single and Multiple DFSMShsm Host Environments (continued)

Function That Has Volume	Function That Needs Volume	Result
Recover	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Recycle (source)	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Recycle (target)	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Tapecopy (source)	Tapecopy source	No contention.

Implications of Redefining the Backup Cycle

If you redefine the backup cycle, you might leave daily backup volumes unused.

Example: Assume the following conditions:

- You have specified the DEFINE BACKUP(YNNYNNY) command. Automatic backup occurs on days 1, 4, and 7.
- You have assigned volume A to day 1, volume B to day 4, and volume C to day 7.
- You later specify DEFINE BACKUP(YNYN). Automatic backup occurs on days 1 and 3.

These are the results:

- Volume A remains assigned to day 1.
- Volume B is still assigned to day 4, but automatic backup no longer runs on day 4. Therefore, DFSMShsm does not use volume B unless someone issues a BACKVOL command on day 4 in the backup cycle.
- DFSMShsm does not use volume C any more because it was assigned to day 7 and the new cycle has only four days.

To remove volume C (and volume B, if you want) from its day in the backup cycle and to make it an unassigned backup volume, use the DELVOL command. However, do not specify the PURGE parameter, because any backup version on the tape backup volume would be lost to DFSMShsm and the volume information for the tape or DASD volume would be deleted. The UNASSIGN parameter of the DELVOL command does not destroy any of the backup versions already contained on the volume or prevent DFSMShsm from accessing these backup versions during recovery, but it allows the volume to be used again if necessary.

In redefining the backup cycle, you can also use the CYCLESTARTDATE subparameter to redefine the starting date for the cycle.

Inline Backup

Inline backup allows you to request a back up of a data set in a batch environment. Inline backup is a way of overcoming the problems of executing the terminal monitor program (TMP) in the background in order to issue HBACKDS commands.

When using inline backup, you can use the JCL data set reference capabilities. This allows you to specify a relative generation number for a GDG. Also, in executing a

batch job with TMP, if a data set is created (DISP=NEW), an exclusive enqueue is done on the data set name, and the enqueue is not released until the last job step referring to that data set has ended. If a job step after the TMP job step backward-references a newly created data set, the HBACKDS command fails.

Inline backup allows you to back up data sets in the middle of a job. You add a new step to a job by specifying the inline backup module as the program to execute. You specify the data sets being backed up on specially named DD statements. Inline backup extracts the data set names associated with these DDnames and then performs the back up.

You can use the ARCINBAK program to make a quick backup that can be used in a later step (of the same job). This quick backup enables a restart in the middle of a job stream. Because it is mandatory that the target backup volume is available to perform a recovery even if a continuous backup load exists, DFSMShsm has a backup tape takeaway capability from data set backup.

The results of each attempted data set backup can be written to a specially named output data set. SNAP dumps of storage relating to inline backup can be written to a specially named output data set for certain error conditions.

The following data sets are not supported:

- Uncataloged data sets
- Backup-while-open data sets
- Non-SMS-managed VSAM data sets that are open for processing by another task at the time of inline backup processing

Related reading:

- For additional information about how to start the ARCINBAK program, see "Invoking Inline Backup."
- For additional information about how to start the ARCINBAK program, see *z/OS DFSMShsm Managing Your Own Data*.

Invoking Inline Backup

To use inline backup:

1. Include a step at the correct place in the job to execute a program called ARCINBAK.
2. Specify the name of the data set to be processed by including a valid DD statement with a DDNAME of BACK*xxxx* (where *xxxx* is 1 to 4 characters). Do this for each data set you want backed up.

Note: 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.

Any form of the data set name is acceptable to ARCINBAK. Specification of a relative generation number for a GDG or a reference to a DDname in a previous step is acceptable. A DD statement with a DDNAME of ARCPRINT can be optionally included in the ARCINBAK program job step. This DD statement defines a SYSOUT data set to contain messages about data sets for which a backup was attempted.

The possible messages are as follows:

```
BACKUP FOR (DSNAME) SUCCESSFUL
DDNAME (DDNAME) NOT BACKED UP, UNABLE TO GET ASSOCIATED JFCB
BACKUP FOR (DSNAME) FAILED, DATA SET NOT CATALOGED
BACKUP FOR (DSNAME) FAILED, RC = (RETURN CODE), REAS = (REASON CODE)
```

At the completion of the ARCINBAK program job step, one of the following return codes is displayed:

- 0 — All backups are successful.
- 4 — No DDnames prefixed with BACK are found or the JFCB associated with a DDname cannot be found.
- 8 — Backup of an uncataloged data set has been attempted. Uncataloged data sets are not supported by inline backup.
- 12 — Backup of a data set failed. One of the following reason codes is also displayed:

  RC100 : DFSMShsm is not active; request rejected.
  RC1–RC99 : For message return codes and reason codes, see message ARC13*XX*I, where *XX* is the return code.

A DD statement with a DDNAME of ARCSNAP can be included in the ARCINBAK program job step. This DD statement defines a SYSOUT data set for the output from a SNAP macro. The following conditions cause a SNAP dump of storage relating to inline backup:

- MWE storage can be dumped when an error occurs in issuing the service call.
- A complete image of the task storage can be dumped when no DDnames or data sets are processed.

A job stream example of data sets that can and cannot be backed up by ARCINBAK in a batch environment is shown in Figure 67 on page 418.

```
//JOBNAME  JOB . . . ,USER=USERID,PASSWORD=USERPSWD
//STEP1    EXEC PGM=USERPGM
//SYSPRINT DD SYSOUT=A
//DSET1    DD DSN=USERID.N03.GDG(-1),DISP=OLD
//DSET2    DD DSN=USERID.N03.PSFB,DISP=OLD
//DSET3    DD DSN=USERID.N04.PSFB,DISP=OLD
//DSET4    DD DSN=USERID.N03.KSDS,DISP=OLD
/*
//STEP2    EXEC PGM=ARCINBAK, PARM='TARGET(TAPE)
//                     CC=(PREFERRED,PHYSICALEND)'
//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.N01.GDG.G0001V00,DISP=SHR
//* ---------------------------------------------------------------
//*        BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ---------------------------------------------------------------
//BACK05   DD  DSN=USERID.N01.PSFB,DISP=SHR
//* ---------------------------------------------------------------
//*        BACKUP OF UNCATALOGED DATA SET SHOULD FAIL.
//* ---------------------------------------------------------------
//BACK06   DD  DSN=USERID.N02.UNCAT,VOL=SER=VOL003,UNIT=3390,DISP=SHR
//* ---------------------------------------------------------------
//*        BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ---------------------------------------------------------------
//* ---------------------------------------------------------------
//BACK07   DD  DSN=USERID.N01.KSDS,DISP=SHR
//* ---------------------------------------------------------------
//*        BACKUP OF OPEN IN-USE VSAM DATA SET SHOULD BE SUCCESSFUL.
//* ---------------------------------------------------------------
//BACK08   DD  DSN=USERID.N02.KSDS,DISP=SHR
//* ---------------------------------------------------------------
//*        BACKUP OF RACF PROTECTED NON-VSAM DATA SET
//*        BY AN UNAUTHORIZED USER SHOULD FAIL.
//* ---------------------------------------------------------------
//BACK09   DD  DSN=USERXX.N02.PSFB,DISP=SHR
//* ---------------------------------------------------------------
//*        DDNAME PREFIX OTHER THAN BACK IS NOT PROCESSED.
//* ---------------------------------------------------------------
//BAK01    DD  DSN=*.STEP1.DSET3,DISP=SHR
//* ---------------------------------------------------------------
//*        DDNAME PREFIX OTHER THAN BACK IS NOT PROCESSED.
//* ---------------------------------------------------------------
//BAK02    DD  DSN=USERID.N02.PSFB,DISP=SHR
/*
```

Figure 67. Example of Inline Backup Job Stream

The return code from executing the inline backup job stream example of the ARCINBAK program is 12, and the resulting messages for each DDname prefixed with BACK are shown in Figure 68 on page 419:

```
BACKUP FOR USERID.N03.GDG.G0001V00        SUCCESSFUL
BACKUP FOR USERID.N03.PSFB                SUCCESSFUL
BACKUP FOR USERID.N03.KSDS                SUCCESSFUL
BACKUP FOR USERID.N01.GDG.G0001V00        SUCCESSFUL
BACKUP FOR USERID.N01.PSFB                SUCCESSFUL
BACKUP FOR USERID.N02.UNCAT               FAILED, DATA SET NOT CATALOGED
BACKUP FOR USERID.N01.KSDS                SUCCESSFUL
BACKUP FOR USERID.N02.KSDS                SUCCESSFUL
BACKUP FOR USERXX.N02.PSFB                FAILED, RC = 039, REAS = 008
```

Figure 68. Example of Messages Resulting from Inline Backup Job Stream

Preventing Backup Processing When Using the HOLD BACKUP Command

You can use the HOLD BACKUP command to prevent all backup processing, prevent only automatic backup processing, or prevent fast replication backup processing. If you do not want any backup processing at all (for volume, data sets, and auto backups), issue the HOLD BACKUP command by itself. You can specify the AUTO subparameter to prevent only automatic backups from running. You can specify the DSCOMMAND subparameter to prevents only data set backups from running.

Notes:

1. The DSCOMMAND option applies to the BACKDS, HBACKDS, ARCHBACK, and ARCINBAK data set backup requests and to backups that are performed as INUSE RETRY from automatic backup.
2. The HOLD ALL command prevents all DFSMShsm functions (except logging) from running.
3. If you are using SETSYS BACKUP parameters or a data set backup installation exit (ARCBDEXT) to direct DFSMShsm to retry backup requests that fail because data sets are in use (see "Specifying How DFSMShsm Should Respond to a Data Set in Use" on page 175), and if DFSMShsm has scheduled any such retries during automatic backup, those backup retries will not be affected by the HOLD BACKUP(AUTO) command. You will need to use HOLD BACKUP(DSCOMMAND).

The following command prevents processing of all data set backups (BACKDS, HBACKDS, ARCHBACK, and ARCINBAK), and it prevents the processing of all backups that would be the result of INUSE RETRY from automatic backup. All WAIT requests fail, and control returns to the requester.

```
HOLD BACKUP(DSCOMMAND)
```

The following command prevents processing of data set backups that are directed to ML1 DASD. All WAIT requests that target DASD fail, and control returns to the requester.

```
HOLD BACKUP(DSCOMMAND(DASD))
```

The following command prevents processing of data set backups that are directed to tape. All WAIT requests that target tape and all WAIT requests for large data sets that do not specify a target fail. Control returns to the requester.

```
HOLD BACKUP(DSCOMMAND(TAPE))
```

The following command prevents processing of automatic backups, data set backups that are directed to tape, and all WAIT requests for large data sets that do not specify a target. However, data set backups to ML1 DASD are not affected.

```
HOLD BACKUP(AUTO DSCOMMAND(TAPE))
```

The following command prevents processing of automatic backups and data set backups that are directed to ML1 DASD. Data set backups to tape are not affected.

```
HOLD BACKUP(AUTO DSCOMMAND(DASD))
```

During automatic or volume backup, you can specify whether the holding action occurs immediately at the end of processing each data set or at the end of processing each volume currently being backed up. If you hold backup at end of volume, it can take some time for all of the volumes that are being processed by the different backup tasks to complete and the hold to take full effect. If you hold backup at the end of data sets, DFSMShsm resumes data set backup with the next queued entry when the hold is released. However, for automatic backup, the volume interrupted by the HOLD at end of data set remains eligible for processing if automatic backup is restarted.

The holding action begins when an MWE is selected for processing, not when the HOLD command is entered. If you enter a wait-type backup request and the backup function is held, DFSMShsm fails the wait-type backup request because the hold backup function gives no indication when the backup function will be available.

When a non-wait-type backup request is issued and the appropriate backup function is held, the request remains on the queue and is processed when the appropriate backup request function is released. The held backup processing function can be released with the RELEASE BACKUP command.

Related reading: For additional information about other ways to prevent DFSMShsm from processing, see "Preventing the Processing of Data Sets during Space Management" on page 131.

GUARANTEED SPACE Attribute

The storage class attribute of GUARANTEED SPACE allows the volumes on which an SMS data set is to reside to be chosen explicitly when the data set is created. If GUARANTEED SPACE is used and if any of the volumes specified are not part of one of the storage groups to which the storage class maps, the data set creation fails. The use of GUARANTEED SPACE requires some special considerations if DFSMShsm is to manage data sets assigned to such a storage class.

DFSMShsm is used as the data mover on partitioned data sets, catalogs, and other data sets that are not supported by DFSMSdss. When DFSMShsm is the data mover, the Access Method Services functions of EXPORT and IMPORT back up and recover VSAM data sets. The IMPORT function redefines a VSAM data set by using a copy of the catalog information saved in the EXPORT copy of the data set. This define request explicitly specifies the volumes from which the data set was exported. The explicit specification of the volume affects the way data sets are

imported with the GUARANTEED SPACE attribute. Likewise, for non-VSAM data sets, VTOC Data Set Services uses the volume serial number passed by DFSMShsm if the storage class is GUARANTEED SPACE.

Results of Recovering GUARANTEED SPACE Data Sets

With DFSMShsm as the data mover, data sets recovered with the GUARANTEED SPACE attribute are normally placed on the volume from which they were backed up. Consider the effects on a backed up, multiple-volume VSAM data set. A subsequent recover preserves the volume-to-component relationship that originally existed as long as the data set remains SMS-managed. For a data set in a GUARANTEED SPACE storage class, the relationship includes returning the data set to the same volumes from which it was backed up. Thus, if a data set has been originally defined with the data component on one volume and the index component on another, a backup and recover sequence preserves the relationship between the components and the volumes. Similarly, if a key range data set has three key ranges on three separate volumes, the relationship is preserved.

Problems That Can Occur

In an environment in which the storage groups and storage classes do not change, the preceding results are adequate to allow DFSMShsm to manage the data sets. However, you can expect problems in an environment that is subject to storage group reconfiguration or subject to changes in the mapping of data sets to storage classes. These problems can discourage the use of DFSMShsm space management for these data sets and might require special contingency procedures for availability management. The problems that occur relate to:

- Changes in the storage group
- Changes in the storage-class-to-data-set mapping
- Conversions of data sets to the GUARANTEED SPACE storage class

Changes in the Storage Group: Consider the consequence of making a specific volume request in a storage group configuration that has changed. Assume that a data set with the GUARANTEED SPACE storage class has been backed up. Sometime later, one or more of the volumes on which the data set has resided are removed from the storage groups that contain GUARANTEED SPACE data sets. A subsequent recover of the data set fails because at least one of the requested volumes is no longer part of a storage group for a GUARANTEED SPACE request. The result is the same for both single-volume data sets and multiple-volume VSAM data sets if the specific volume required is unavailable in the necessary storage group.

If a storage group is changed to remove volumes that data sets migrated from, you must do one of the following:

- Change the ACS storage class routine so that a storage class with GUARANTEED SPACE is not assigned to the data set.
- Change the volume configuration again so the storage group contains the volumes from which the data set was backed up.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This step applies only if the volumes are already SMS-managed.
- Use the RECOVER command with the FORCENONSMS and the VOLUME parameters. This RECOVER command causes the data set to be recovered to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

Note: If the data set was multiple-volume when it was backed up, it is recovered to a single volume.

Changes in the Storage-Class-to-Data-Set Mapping: Consider the consequence of making a specific volume request in a storage class mapping that has changed. Assume that a data set is defined and one of the following conditions occur:

- The ACS routines do not assign the data set to any storage class.
- The ACS routines assigns the data set to a storage class that does not have the GUARANTEED SPACE attribute.

Later, assume one of the following changes occur:

- The ACS routine is changed so that it assigns the data set to a storage class with the GUARANTEED SPACE attribute.
- The original storage class is changed to include the GUARANTEED SPACE attribute.

Now, when you recover this data set, the recovery might fail because the requested volumes are not in the storage groups that receive GUARANTEED SPACE requests.

If an SMS-managed data set's original storage class is changed to specify the GUARANTEED SPACE attribute before the data set is recovered, you must do one of the following:

- Change the ACS storage class routine so that a storage class without GUARANTEED SPACE is assigned to the data set.
- Change the storage class of the data set to a storage class without GUARANTEED SPACE, without recalling the data set, by using the IDCAMS ALTER command.
- Change the volume configuration in the appropriate storage groups to contain the volumes from which the data set was backed up. The volumes might need to be converted to SMS management first. It can be difficult to determine all of the volumes involved. DFSMShsm does not save all of the original volume serial numbers, only that of the volume on which the base data component resided.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This step applies only if the volumes are already SMS-managed.
- Use the RECOVER command with the FORCENONSMS and the VOLUME parameters. This RECOVER command causes the data set to be recovered to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

  Note: If the data set was multiple-volume when it was backed up, it is recovered to a single volume.

Conversions of Data Sets to GUARANTEED SPACE: If the data set was non-SMS-managed when it was backed up and is assigned a storage class with GUARANTEED SPACE during recover, it is assigned to a volume in a storage group that contains GUARANTEED SPACE data sets.

If the data set was SMS-managed when it was migrated but did not have GUARANTEED SPACE, the volumes from which it was backed up might not be in a storage group that receives GUARANTEED SPACE requests. To avoid errors, you must do one of the following before the recover:

- Change the target storage class with GUARANTEED SPACE to map to a storage group that contains the original volumes.
- Add the original volumes to a storage group mapped to by the target storage class with GUARANTEED SPACE.

Data Set Backup with Concurrent Copy

The data set backup function allows the use of either concurrent copy or a standard backup method for copying both SMS-managed and non-SMS-managed data sets. Use the concurrent copy (CC) keyword to specify that you want concurrent copy, rather than standard backup methods, to back up your data sets by command.

Note: For SMS data sets, you do not have to use the CC keyword. Use the CC keyword only if you want to override SMS concurrent copy specifications.

To control the use of concurrent copy for your data set, several options are available for the data set backup function. If you omit the options, DFSMShsm backs up the data set using the management class attributes for SMS-managed data. The issuer of the backup will not receive notice until physical completion. For non-SMS managed data sets, DFSMShsm uses the standard backup method.

The following keywords control concurrent copy usage:

REQUIRED
DFSMShsm uses concurrent copy to backup the data set. If concurrent copy is unavailable, the backup fails.

PREFERRED
DFSMShsm uses concurrent copy to backup the data set. If concurrent copy is unavailable, the data set backup continues as if this keyword were not specified.

STANDARD
DFSMShsm backs up the data set without using concurrent copy (the default). Control returns to the requester at the physical end of the copy.

PHYSICALEND
Users receive data set backup results when the backup operation is complete (the default). For a WAIT-type request, the job receives control only after the physical end of the copy.

LOGICALEND
DFSMShsm releases serialization for the data set and returns control to the data set backup WAIT requester after concurrent copy initialization successfully completes. If concurrent copy initialization fails and PREFERRED was specified, notification is delayed until the end of the backup operation. If concurrent copy initialization fails and REQUIRED was specified, the request fails. Because users are not notified at the physical end of the backup operation, use caution if you use this option.

Authorizing Concurrent Copy with RACF

By default, all users can use the CC options. However, if RACF indicates a lack of authority, DFSMShsm fails the data set backup request if the specified concurrent copy request was specified with the REQUIRED subparameter. If the CC option was used without the REQUIRED subparameter and RACF indicates a lack of authority, DFSMShsm continues to back up the data set as if the CC keyword were not specified on the backup command.

Additional Considerations for Dump

Several additional topics are related to dump:

- Dumping DFSMShsm-owned volumes
- Dumping control data sets
- Dumping the journal data set
- Preventing dump processing
- Invoking full-volume dump compression
- Bypassing enqueue processing between full-volume dump and DFSMSdfp functions

Dumping DFSMShsm-Owned Volumes

DFSMShsm-owned DASD backup and migration level 2 volumes are not automatically dumped. You must issue a DFSMShsm BACKVOL command for each migration level 2 volume you want to dump. The activity against the volumes should be at an absolute minimum while they are being dumped. The volumes do not have any internal DFSMShsm serialization performed for them to prevent migration copies or backup versions from being created on them while they are being dumped, so it is imperative to dump these volumes when no migration or backups are being done at the same time. Also, no VTOC copy data sets are created for DFSMShsm-owned volumes regardless of the VTOCCOPIES subparameter of the target dump classes.

Dumping Control Data Sets

If you dump the volumes that contain the control data sets, *you will destroy your control data sets if you do a restore of the volume*. If you restore the control data sets, you must immediately recover the control data sets. If you dump the volume containing the BCDS, no concurrent dump activity should be processing on another host. A lockout can occur.

Related reading: For additional information about how to recover control data sets, see "Reasons For Recovering the Control Data Sets" on page 462.

Dumping the Journal Data Set

If you dump the volume containing the journal data set, if your environment consists of multiple hosts with a global resource serialization product, and if the journal resource QNAME=ARCGPA and RNAME=ARCRJRN and the resource QNAME=SYSVTOC and RNAME=volser-containing-journal, both resources must be treated as either local resources or as global resources.

If the resource QNAME=SYSVTOC and if RNAME=volser-containing-journal is named in the systems exclusion resource name list, the resource QNAME=ARCGPA and RNAME=ARCRJRN must also be named in the systems exclusion resource name list. In this case, global resource serialization treats both as local resources. The reserve conversion resource name list is not searched, and the system is allowed to issue a hardware reserve for these resources.

If the journal data set shares a volume with a data set that has a higher access priority requirement than the journal data set and if the resource QNAME=ARCGPA and RNAME=ARCRJRN is named on the reserve conversion resource name list, then the resource QNAME=SYSVTOC and RNAME=volser-containing-journal must also be named on the reserve conversion

resource name list. In this case, global resource serialization treats both as global resources to be processed around the ring and suppresses any hardware reserves for these resources.

Preventing Dump Processing

You can use the DFSMShsm HOLD command to prevent all dump processing or to prevent only automatic dump processing. The following command prevents automatic dump processing without preventing command dump processing:

```
HOLD DUMP(AUTO)
```

The following command prevents both command and automatic dump processing:

```
HOLD DUMP
```

You can specify whether you want the holding action to occur immediately at the end of processing each data set currently being dumped or to occur at the end of each volume being dumped. If you hold dump at end of volume, it can take a long time for all of the volumes that are being processed by the different dump tasks to be completed and the hold to take full effect. If you hold dump at the end of data sets, DFSMShsm resumes command dump with the next queued entry when the hold is released. However, for automatic dump, the volume that was being processed when the HOLD at end of data set occurred remains eligible for processing if automatic dump is restarted.

The holding action begins when an MWE is selected for processing, not when the HOLD command is entered. If you use an HSENDCMD command to enter a wait-type dump request and the dump function is held, DFSMShsm fails the wait-type dump request because the hold dump function gives no indication of when the dump function is available.

When a non-wait-type dump request is issued and the dump function is held, the request remains on the queue and is processed when the dump function is released. The held dump function is released with the RELEASE DUMP command.

Related reading: For additional information about other ways to prevent DFSMShsm from processing a dump function, see “Preventing the Processing of Data Sets during Space Management” on page 131.

Dumping All Allocated Space

Programming Interface Information

DFSMShsm’s full-volume dump invocation of DFSMSdss specifies the ALLEXCP and ALLDATA(*) parameters. If the field in the DSCB that shows the last-used track and the block on that track contain zeros, the ALLEXCP option dumps all allocated space. When ALLDATA(*) is specified, all allocated space of SAM and PAM data sets, and data sets with a null DSORG field and whose last-used track and block on that track is greater than zero, are dumped. Use the following PATCH commands to suppress these parameters:

To suppress ALLEXCP:

```
PATCH.MCVT.+3C3 BITS(1.......)
```

To suppress ALLDATA(*):

```
PATCH.MCVT.+3C3  BITS(01......)
```

To suppress both ALLEXCP and ALLDATA(*):

```
PATCH.MCVT.+3C3  BITS(11......)
```

End of Programming Interface Information

Invoking Full-Volume Dump Compression

You can compress all volume dumps by specifying the compress parameter at the DFSMShsm full-volume dump invocation of DFSMSdss. Use the following PATCH command to specify the compress parameter:

Programming Interface Information

```
PATCH.MCVT.+3C3  BITS(001.....)
```

End of Programming Interface Information

Note: If DFHSM Version 2 Release 5.0 or later is installed and this patch is on, it also applies to DFSMSdss as the data mover and to aggregate processing.

Bypassing Enqueue Processing between Full-Volume Dump and DFP Functions

To avoid contention problems between DFSMShsm full-volume dump and DFP scratch, rename, and allocate services, major resource name ADRLOCK and minor resource name NONSPEC are allocated EXClusive when the volume is not known.

Since this process can have an impact on dump performance, you can bypass it by using the following patch:

Programming Interface Information

```
PATCH  .MCVT.+38F  X'10'  VERIFY(.MCVT.+38F  X'00')
```

End of Programming Interface Information

Volume Recovery and Restore Considerations

The following topics discuss items to consider when doing volume recovery and restore:

- Restoring data sets or volumes directly with DFSMSdss
- Requesting a list of dump volumes for a dump copy
- Determining the dump copy data set name
- Invoking DFSMSdss's RESTORE function
- Preventing recovery and restore processing
- Use of the date-last-referenced field for recovery

Restoring Data Sets or Volumes Directly with DFSMSdss

It is possible to invoke DFSMSdss directly to restore a data set or volume from a DFSMShsm-owned dump copy. This requires you to perform the following steps:

1. Request from DFSMShsm a list of the tape volumes that comprise the dump copy.
2. Determine the data set name that is used for the dump copy.
3. Invoke the DFSMSdss RESTORE function directly.

Requesting a List of Dump Volumes for a Dump Copy

Use the DFSMShsm LIST command to request the set of dump volumes that compose the required dump copy. You must know the volume serial number of the volume you need to restore or the volume serial number of the volume on which the data set you want to restore resided when it was dumped.

Determining the Dump Copy Data Set Name

DFSMShsm generates a unique data set name for each dump copy that it has created through DFSMSdss. This data set name must be known in order to allocate the dump volumes. If you need to perform a restore using DFSMSdss directly without DFSMShsm, you need the DFSMShsm-generated data set name. This name can be obtained by issuing the following command:

```
LIST DUMPVOLUME(volser) DUMPCONTENTS
```

Related reading: For additional information about the format of these data set names and how to determine the name of a specific dump copy, see "Names of Dump Copy Data Sets" on page 181.

Invoking the DFSMSdss RESTORE Function

You can invoke DFSMSdss in one of two ways: either write the necessary JCL to submit a batch job, or if you have ISMF, use the menu-driven panels of ISMF to build a job stream that will perform a DFSMSdss RESTORE.

If you received an invalid track format error (ADR367E) during a DFSMSdss full-volume restore, you can restore this volume by issuing the following DFSMSdss restore command in your JCL or ISMF job stream:

```
RESTORE FULL INDDNAME(tapename) OUTDDNAME(dasdname)
     PURGE COPYVOLID
```

This restore restriction exists because DFSMShsm uses the CANCELERROR keyword in its full-volume dump restore. Volumes that receive invalid track format errors (ADR367E) can only be restored outside of DFSMShsm by using the DFSMSdss restore command, as previously explained.

Attention: DFSMSdss cannot directly restore dump tapes that are password protected by DFSMShsm. Such tapes require special consideration during restore processing. Because DFSMSdss uses EXCP for reading, OPEN requires that the data be protected from destruction. Therefore dump tapes with password protection must be write-protected (ring removed or cartridge knob set) before they are mounted for restore processing.

Related reading: For additional information about using DFSMSdss JCL code or the ISMF panels for DFSMSdss, see the *z/OS DFSMSdss Storage Administration Guide* and *z/OS DFSMSdss Storage Administration Reference*.

Preventing Recovery and Restore Processing When Using the HOLD RECOVER Command

You can use the HOLD RECOVER command to prevent all recovery, to restore processing, or to prevent recovery alone from tape. The following command prevents recovery processing from tape without preventing recovery processing from DASD:

```
HOLD RECOVER(TAPEDATASET)
```

Note: Holding tape recovery has the effect of holding all restores, because all dump copies managed by DFSMShsm are on tape. Holding tape recovery does not hold the data set recovery from a mounted backup tape already in use.

The following command prevents both tape and DASD recovery processing:

```
HOLD RECOVER
```

The holding action begins when an MWE is selected for processing, not when the HOLD command is entered. If you enter a wait-type recovery or restore request and the recovery function is held, DFSMShsm fails the wait-type recovery request because the hold recovery function gives no indication of when the recovery function will be available.

When a non-wait-type recovery or restore request is issued and the appropriate recovery or restore function is held, the request remains on the queue and is processed when the appropriate recovery function is released. The held recovery or restore function can be released with the RELEASE command.

Using the Date-Last-Referenced Field for Recovery

When a data set is recovered or restored by an unauthorized user (HRECOVER), the date-last-referenced field in the data set VTOC entry of the data set is updated with the current date. When a data set is recovered or restored by an authorized user, the date-last-referenced field in the data set VTOC entry is updated with the date-last-referenced field from the data set's VTOC entry that was saved at the time of backup or dump. When a data set is recovered or restored with the NEWNAME parameter of the RECOVER command, the date-last-referenced field in the data set VTOC entry is updated with the current date regardless of the authorization of the user.

Deferred Generation Data Sets

Deferred generation data sets are generation data sets for which a failure has occurred before the data set is correctly associated with the generation data group. These data sets are temporary in the sense that at some time they may become associated with the GDS. DFSMShsm treats deferred generation data sets as permanent non-GDS data sets.

Security Considerations for Availability Management

DFSMShsm does not check data set security during automatic backup and dump. DFSMShsm also bypasses security checking when it processes operator commands entered at the system console or commands issued by a DFSMShsm-authorized user.

DFSMShsm checks security for data sets when a user who is not DFSMShsm-authorized issues an HBACKDS, HBDELETE, HALTERDS, or HRECOVER command. Security checking is not done when DFSMShsm-authorized users issue the user commands. If users are not authorized to manipulate data, DFSMShsm does not permit them to back up data sets, delete backup versions, change the conditions for backup versions, or recover data sets. Table 25 shows the RACF authority required to perform each availability management function.

Table 25. Resource Access Authority Required for DFSMShsm Functions

DFSMShsm Function	RACF Resource Access Authority Required
Back up a data set	Update
Recover a backup version without specifying the NEWNAME parameter	Alter If profile recovery is required, authority is needed to create a discrete RACF profile for the recovered data set.
Recover a backup version and specify the NEWNAME parameter	Read authority to the data set being recovered. Alter authority to the newly named data set if one exists. If profile recovery is required, authority is needed to create a discrete RACF profile for the recovered data set.
Changing backup parameters	Alter
Deleting backup versions	Alter

Authorization Checking during Recovery

During recovery, DFSMShsm makes a first authorization check against the original profile. If it cannot find either a generic profile or a discrete profile for a cataloged data set that was RACF-indicated at the time of backup or recovery, DFSMShsm determines whether a backup profile exists. If a backup profile exists, DFSMShsm makes a second authorization check with the backup profile name. If the second authorization check fails, the recovery fails.

When DFSMShsm is the data mover, a new RACF access control environment element (ACEE) is created that describes the user's RACF status. This ACEE exists for the duration of the cataloging operation and is effective only for the recovery task. After the cataloging operation has completed, DFSMShsm removes the ACEE.

For a non-VSAM data set, DFSMSdfp catalog management is invoked to catalog the data set. DFSMSdfp does its own security checking of the requester against the catalog using this ACEE. If security checking against the catalog fails for lack of correct authority, DFSMShsm sets a return code to reflect the failure of the catalog operation. RACF issues messages describing the reasons for denying access to the catalog.

For a VSAM data set, a new RACF ACEE is created when the target data set of the Access Method Services IMPORT has not been cataloged at the start of the recovery operation. This ACEE is effective for the duration of DFSMShsm's invocation of Access Method Services to perform the IMPORT operation.

DFSMSdfp catalog management will use this ACEE to verify the requester's authority to catalog the data set. If the catalog operation fails, the IMPORT fails and generates an appropriate return code. RACF issues messages describing the reasons for denying access to the catalog.

When DFSMSdss is the data mover, the authorization process is the same, except that DFSMSdss does the cataloging.

Recovering or Restoring RACF-Indicated Data Sets

When you use the RECOVER or HRECOVER command to recover or restore a RACF-indicated, cataloged, non-VSAM data set and the target volume is different from the source volume, DFSMShsm changes the volume serial number and device type information in the RACF discrete data set profile to indicate the volume where DFSMShsm has recovered the data set.

If the attempt to change a profile fails because the profile cannot be found, DFSMShsm attempts to recover a backed up profile for the recovered data set.

When a RACF naming conversion table converts a data set name to a different profile name, DFSMShsm will not be able to recover the discrete profile. The RACF administrator will need to enable access.

Recovering RACF Discrete Profiles

If a recovered data set was cataloged and RACF-indicated when it was backed up or when it was recovered or restored, DFSMShsm tries to recover the RACF discrete profile for the data set even if you specified the NOPROFILEBACKUP parameter of the SETSYS command or if the profile already exists.

DFSMShsm creates a RACF discrete profile during recovery by performing the following actions:

1. Modeling the new profile after the original profile, and if that isn't possible,
2. Modeling the new profile after the backup profile and if that isn't possible,
3. Creating a minimal discrete profile.

DFSMShsm tries to recover the RACF discrete profile by using the profile of the original data set as a model for creating a profile. If the original data set's profile no longer exists, DFSMShsm tries to create a profile by using a backup profile for the model.

If a backup profile does not exist, DFSMShsm tries to create a minimal discrete profile. A minimal discrete profile is one for which DFSMShsm supplies no access list and no model profile. The access authority with which users can address the data set is taken from the defaults defined for the user (or group if the high-level qualifier is a group name). If a DFSMShsm-authorized user recovers the data sets, defaults are taken from the user ID associated with the DFSMShsm-started procedure.

DFSMShsm creates a minimal discrete profile only if neither the original profile nor a backup profile exists for the recovered data set. If the profile creation fails because a profile already exists for the recovered data set, DFSMShsm considers the profile recovery to be successful, and normal processing continues. When this occurs, RACF may issue message ICH408I at the security console to indicate that the resource is already defined.

Note: DFSMShsm recovers a discrete profile if one existed at the time of backup or recovery. If you change from RACF discrete profile protection to generic profile protection, DFSMShsm still recovers the discrete profiles when it recovers the backup versions of data sets that were once protected with discrete profiles. So, discrete profiles can reappear even if you now use generic profile protection.

Related reading: For additional information about the installation of generic profile options, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Authorization Checking during Profile Recovery

When a DFSMShsm-authorized user issues a RECOVER or HRECOVER command, no security checking is performed. If the recover request originates with a non-DFSMShsm-authorized user, the recover is done with the user's ACEE, which is a control block for RACF. DFSMShsm informs RACF that security checking is to be done, and the recover request fails if RACF determines that the user is not authorized to create a discrete profile for the data set.

Characteristics of the Recovered Profile

When a DFSMShsm-authorized user issues a RECOVER or HRECOVER command, DFSMShsm recovers profiles using DFSMShsm's ACEE. Consequently, the recovered profile shows the group name from the user ID associated with the DFSMShsm-started procedure. If a minimal profile is created when a DFSMShsm-authorized user issues a recover request, additional information from this user ID is placed in the recovered profile. This information includes levels of auditing, access authority required by other users (universal access), and profile ownership.

If a non-DFSMShsm-authorized user issues the recover request, the user's ACEE is used to create a new profile. The recovered profile reflects the group name of the requester. If a minimal profile is created, the profile also reflects the requester's user ID in the owner field and the requester's defaults for audit level and universal access authority.

Note: For steps 2 on page 430 and 3 on page 430, if the DFSMShsm-started procedure is not associated with a RACF user ID, the user and group names of the recovered profile appear as an asterisk (*) if the data set is recovered by a DFSMShsm-authorized user.

Effects of Profile Modeling on Profile Recovery

Profile modeling is a RACF function that merges information from a *model profile* into a new profile when it is created. Profile modeling is available at both the user ID and group level. If the user ID associated with the DFSMShsm-started procedure has a model profile, information from the model is put into all backup profiles. Model profiles can provide information for universal access, auditing level, profile owner, level field, installation-defined data, and an access list. If a profile is subsequently recovered, the additional information from the profile model can be carried along and inserted into the recovered profile.

When a non-DFSMShsm-authorized user issues a recover request that results in profile recovery, the user's ACEE is used to create the profile. Consequently, if profile modeling is active for the user or the user's group, information from the appropriate profile is merged into the newly created, recovered profile.

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

Accessing a Data Set When the Profile Cannot Be Recovered

Even if DFSMShsm cannot recover a discrete profile, you can access the recovered data set. Table 26 shows how to access data sets using the DFSMShsm RECOVER command when recovering data sets that are RACF-protected with discrete profiles and DFSMShsm is unable to recover a discrete profile. These data sets were generally backed up by DFHSM before Version 2 Release 2.0.

Table 26. Recovering RACF-Protected Data Sets Backed Up under DFHSM before Version 2 Release 2.0

Type of Data Set	Parameter	Result	How to Access the Data Set
Cataloged Non-VSAM	RECOVER NEWNAME	If the RACF indicator was on when the backup version was created for the current data set, the RACF indicator is on in the recovered data set.	If a RACF profile does not already exist, create a generic profile or use the RACF ADDSD command with the NOSET parameter to create a RACF discrete profile for the data set with the new name.
Cataloged VSAM	RECOVER NEWNAME	DFSMShsm scratches the newly named data set. The recovered data set replaces the newly named data set, including the catalog record.	If the original data set was protected with a discrete profile and the newly named data set was not, use the ADDSD NOSET command to create a new discrete profile. Or, you can create a generic profile for the newly named data set.
Uncataloged	RECOVER TOVOLUME (Volume different from the one the data set was backed up from)	If the RACF indicator was on when the backup version was created or the newly named data set is RACF indicated, the RACF indicator is on in the recovered data set.	If a RACF profile does not already exist, use the RACF ADDSD command with the NOSET parameter to create a RACF profile for the data set.
Cataloged Non-VSAM or VSAM	RECOVER (NEWNAME is not specified)	If the RACF indicator was on when the backup version was created for the current data set, the RACF indicator is on in the recovered data set.	If a RACF profile does not already exist, create a generic profile or use the RACF ADDSD command with the NOSET parameter to create a RACF discrete profile for the data set.

Maintaining Password Protection for Recovered Data Sets

Password protection in MVS is implemented differently for non-VSAM and VSAM data sets. Password protection is not applicable to SMS-managed data sets. The passwords for all non-VSAM data sets are stored in one system password data set. The password for each VSAM data set is stored in a catalog record for the specific VSAM data set. Therefore, VSAM passwords are maintained during DFSMShsm processing as a part of regular catalog record creation and updating.

If you are recovering an existing VSAM data set, the password is the current password. If you are recovering a VSAM data set that has been deleted, the password is the password that existed when DFSMShsm backed up the data set.

If you specify the NEWNAME parameter of a RECOVER command for a password-protected, non-VSAM data set, DFSMShsm marks in the DSCB of the recovered data set the same password protection as the original data set had. Enter a password for the recovered data set in the system password data set. Then, use the password to access the recovered, non-VSAM data set.

For VSAM data sets, whether or not you specify the NEWNAME parameter, DFSMShsm marks in the catalog that the recovered data set is protected and enters the old password in the catalog record of the recovered data set. Use the old password when you want to access the recovered VSAM data set.

If DFSMShsm migrated a password-protected VSAM data set whose backup versions are to be deleted, you must specify the password or DFSMShsm fails the request. When deleting backup versions of VSAM data sets that no longer exist or that were migrated by DFSMShsm Version 1 Release 3, DFSMShsm does not check the password.

Be aware of the following password issues when you attempt to change password protection while data sets are backed up:

- You can change the password of a backed up non-VSAM data set by using the IEHPROGM utility. However, to change the password of a backed up VSAM data set you must use the Access Method Services facility. For a backed up VSAM data set, the password required to recover a backup version is the one that existed when the backup version was created if the data set is not currently cataloged.
- If you try to add password protection to a non-existing non-VSAM data set that has backup versions, the password is added to the system password data set. However, the backup versions do not have protection, and when DFSMShsm recovers the data set, the DSCB does not indicate password protection. If an existing non-VSAM data set is password protected at the time it is recovered or restored, even if it was not protected when the data set was backed up or dumped, the data set is password indicated after the recovery or restore.
- If you try to add password protection to a backed up VSAM data set, you must use the Access Method Services facility. If you recover the backup versions, DFSMShsm does not protect the recovered data set if the data set was not protected when DFSMShsm backed it up.

Note: If a catalog has update protection, you are prompted for the password when DFSMShsm tries—during an unauthorized DFSMShsm function from an unauthorized DFSMShsm user—to create a catalog entry in the protected catalog.

DFSMShsm Retrieval of Management Classes

If you are changing the definitions of management classes, you must know when the new definitions will take effect. When DFSMShsm begins automatic availability management or availability management from a volume command, it retrieves a list of all currently defined management classes before it begins processing unless volume-type processing is operating or has just completed. Volume-type processing consists of the following functions:

- Automatic backup
- BACKVOL command

If volume-type processing is in progress or has just been completed when a new volume-type processing is to begin, DFSMShsm uses the list of management

classes that exists from the prior processing. If you are changing the management classes, DFSMShsm must complete its volume-type processing before you can expect the changes in the management classes to become effective.

Availability Management Considerations for Extended SAM and Extended and Compressed VSAM Data Sets

When DFSMShsm does backup and recovery processing for extended format SAM and VSAM compressed data sets, be aware of the following results:

- When DFSMShsm backs up an extended format VSAM compressed data set, it recovers that data set only as an extended format compressed data set.
- When DFSMShsm backs up an extended format SAM compressed data set, it recovers that data set as either an extended format compressed or nonextended format SAM data set.
- When DFSMShsm backs up a compressed SAM or VSAM data set, it suspends software compression for that data set.
- When a compressed SAM or VSAM data set is backed up, DFSMShsm records in the BCDS that it is a compressed data set. It also records both the compressed size of the data set and the size the data set would be if it were not compressed. This information is passed to DCOLLECT.

Availability Management Considerations for Sequential Data Striping

DFSMShsm does backup and recovery processing for single-stripe and multiple-stripe sequential data sets. To process these data sets, the data mover must be DFSMSdss, and the data sets must be SMS managed.

Backup Processing

Incremental backup processing backs up a single-stripe or multiple-stripe data set when the data set's name is specified in the BACKDS command, and other required conditions are satisfied.

During volume incremental backup processing, DFSMShsm backs up a sequential extended format data set when the first stripe (volume) is being processed. Backup is skipped when subsequent stripes (volumes) are being processed.

If a sequential extended format data set is very large (greater than 64K tracks) and cannot fit on one backup DASD volume, backup fails.

The space used on all volumes is reported (message ARC0734I) for the sequential extended format data set backed up.

During full physical volume dump processing, DFSMShsm dumps only the stripe on the volume being dumped. You are responsible for performing full physical volume dumps for each volume containing a stripe of a sequential extended format data set. It is recommended that you use incremental backup versions for multiple-stripe sequential extended formatdata sets.

Incremental backup processing does not occur under the following conditions:

- A sequential extended format data set is not backed up if it is on a non-SMS-managed volume.
- A multiple-stripe sequential extended format data set is not backed up if a SETSYS USERDATASETSERIALIZATION is not presented to DFSMShsm.

- A multiple-stripe sequential extended format data set is not backed up if it is protected by a RACF discrete profile.

Recover and Restore Processing

DFSMShsm recovers or restores a sequential extended format data set (indicated in the MCC record) using DFSMSdss as the data mover.

During full volume restore with APPLYINCREMENTAL, DFSMShsm does not apply the incremental backup version that is more recent than the full physical dump copy to a currently cataloged multiple-stripe sequential extended format data set. The stripe restored from the dump copy is scratched from the volume if the data set is no longer cataloged on the volume being processed.

During data set recovery of a multiple-stripe sequential extended format data set, without FROMDUMP(DUMPVOLUME(*volser*)) being specified, DFSMShsm always recovers an incremental backup version made by DFSMSdss, even if a physical full-volume dump copy is more recent than the incremental backup version. A message is issued to indicate that an incremental backup version has been recovered instead of a more recent dump copy.

During a recovery attempt of a multiple-stripe sequential extended format data set, when FROMDUMP(DUMPVOLUME(*volser*)) has not been specified, recovery fails if the data set has only been backed up by physical full-volume dump.

DFSMSdss restores a sequential extended format data set as follows:

- A sequential extended format data set that was multiple-stripe when it was backed up or dumped can be recovered or restored as a multiple-stripe or single-stripe sequential extended format data set, regardless of the existence of the target data set before the RECOVER command is issued.

  The number of stripes depends on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.
- A sequential extended format data set that was single-stripe when it was backed up or dumped can be recovered or restored as a multiple-stripe or single-stripe sequential extended format data set, regardless of the existence of the target data set before the RECOVER command is issued.

  Again, the number of stripes depends on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.

A sequential extended format data set that was either single-stripe or multiple-stripe when it was backed up or dumped is recovered or restored as a sequential extended format data set or non sequential extended format data set, regardless of the existence of the target data set before the RECOVER command is issued.

A data set that was not sequential extended format when it was backed up is always recovered or restored as a data set that is not sequential extended format. DFSMShsm does not convert a data set that is not sequential extended format to a sequential extended format data set through recovery or restore processing, even if the data set is preallocated as sequential extended format before the RECOVER command is issued.

If ACS filter routines are modified to direct a data set that was sequential extended format when it was backed up to a non-SMS-managed volume, the recovery or restore fails.

If FORCENONSMS is specified on the RECOVER command to force a sequential extended format data set to be recovered or restored to a non-SMS-managed volume, the data set will be recovered as a non sequential extended format data set. The maximum size of the non-SMS SAM data set is 1 volume.

If FROMVOLUME is specified on the RECOVER command to recover a sequential extended format data set, the recovery or restore fails.

If reblocking is required for a sequential extended format data set, DFSMShsm passes the requirement to DFSMSdss, which performs the reblocking.

To restore a single-stripe sequential extended format data set from a physical full volume dump tape, you can issue the RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) command for the stripe dumped to the specified dump tape.

To restore an existing multiple-stripe sequential extended format data set from physical full volume dump tapes, you must issue the RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) command for each stripe of the data set. If the sequential extended format data set no longer exists, you must catalog the data set after the restores, since DFSMSdss and DFSMShsm do not catalog a multiple-stripe sequential extended format data set during physical data set restore processing.

DFSMShsm fails the recovery of a sequential extended format data set if the desired backup version indicates that the data set was sequential extended format when it was backed up, and the data mover was DFSMShsm.

Note: DFSMShsm recovery or restore processing does not honor the "stripedness" specified in the data class DSNTYPE in the ACS filter routine. Changing the DSNTYPE definition or allocating the target data set before issuing the recover command has no effect on converting to or from striped data during DFSMShsm recovery or restore processing.

DFSMShsm does not recover a multiple-stripe sequential extended format data set during incremental full volume recovery.

If the target data set is preallocated and the REPLACE keyword is specified, DFSMShsm data set recovery, restore, incremental volume recovery, and full volume restore with APPLYINCREMENTAL processing delete the non-VSAM target data set, both SMS-managed and non-SMS-managed, before recovering or restoring the backup or dump copy.

DFSMShsm command data set recovery always lets DFSMSdfp select the target SMS-managed volume for a non-VSAM data set, even if the target data set is cataloged before the RECOVER command is issued. The volume the data set is cataloged on is still passed to DFSMSdfp, but the final target volume is selected by DFSMSdfp.

Chapter 11. Availability Management Procedures

This chapter details specific backup, recovery, and full-volume dump procedures associated with DFSMShsm availability management.

Backup and Recovery Procedures

This section tells you how to perform the following functions:

- Interrupt and restart backup processing
- Define a new automatic backup start window
- Back up and recover a catalog

Interrupting and Restarting Backup Processing

To stop the first phase of automatic backup, which is the backup of the control data sets, 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 DFSMShsm-managed volumes having the automatic backup attribute

There are four methods you can use to interrupt automatic backup processing during its last three phases:

- **Holding backup processing**—You can hold backup processing by specifying the BACKUP, BACKUP(AUTO), or ALL parameter of the HOLD command. 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 has finished 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 has completed. 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 NOBACKUP parameter of the SETSYS command. If you disable backup processing while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after DFSMShsm has finished processing the current backup version. If you disable backup processing while DFSMShsm is backing up DFSMShsm-managed volumes, 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 EMERGENCY parameter of the SETSYS command. If you place DFSMShsm in emergency mode while DFSMShsm is moving backup versions or backing up migrated data sets, processing ends after DFSMShsm has finished processing the current data set.

© Copyright IBM Corp. 1984, 2004

If you are in a multiple DFSMShsm-host environment and have enabled the secondary host promotion function, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you have put in emergency mode. Those functions include automatic movement of backup versions from ML1 to tape, and 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.

If you place DFSMShsm in emergency mode while DFSMShsm is backing up DFSMShsm-managed volumes, 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.

- **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 has finished processing the current data set. If you shut down DFSMShsm while DFSMShsm is backing up DFSMShsm-managed volumes, 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 control data sets or the journal data set, DFSMShsm finishes processing the current data set and does not start backing up any other control data sets or the journal data set.

  If you are in a multiple DFSMShsm-host environment, have enabled secondary host promotion, and you have issued a STOP PROMOTE command, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you have stopped. Those functions include automatic movement of backup versions from ML1 to tape, and 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, 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 ones assigned to the day in the backup cycle that was in effect when automatic backup started.

After you restart automatic backup processing, DFSMShsm continues with the next DFSMShsm-managed volume under automatic backup control. However, there are a few considerations:

- If you issued a DFSMShsm STOP command and then a START DFSMShsm command, DFSMShsm can start automatic backup at the point of interruption only:
  - 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 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 issued for one volume.

Related reading: For additional information about using the secondary host promotion function, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Defining a New Automatic Backup Start Window

As you increase the number of volumes that DFSMShsm backs up, you might want to change the automatic backup start window. In a test environment, you might want daily automatic backup to run more than once on a single day. This section describes how to define a new start window.

Defining a New Start Window When Automatic Backup Is Running

You can change the automatic backup start window at any time, even when automatic backup is currently running on the affected DFSMShsm host.

If automatic backup is running but completes before the newly defined start time is reached, automatic backup processing starts again at the newly defined start time.

If automatic backup is running and processing continues beyond the newly defined start time, automatic backup processing does not start again until the next time the newly defined start time occurs.

If you specify a quiesce time for the new automatic backup start window while automatic backup is running and the current time is outside of the newly specified window, automatic backup processing stops.

If automatic backup is released or DFSMShsm is restarted during the automatic backup start window, automatic backup starts, unless prevented by one of the commands discussed in “Interrupting and Restarting Backup Processing” on page 437, or by denying permission to the SETSYS REQUEST command.

Defining a New Start Window When Automatic Backup Is Not Running

If you define a planned start time that is later than the time that automatic backup last ended, automatic backup starts at the new planned start time if it has not already processed all the requested volumes on this day. Otherwise, automatic backup starts at the new planned start time on the next day.

If you define a planned start time that is earlier than the time that automatic backup last ended, automatic backup starts only if it has not run to completion on this day.

Backing Up and Recovering a Catalog

There are no special requirements to back up a catalog. Either the catalog is backed up during volume backup of the volume it is on, or it can be backed up by the BACKDS command. Although the catalog is open when the backup occurs, DFSMShsm enqueues on the catalog to prevent it from being updated while DFSMShsm makes the backup version.

Because this enqueue prevents access to the catalog by any other user, for the duration of the backup, it is imperative that the elapsed time of the backup be kept as short as possible.

Defining the user catalog with as large a data control interval size as possible dramatically reduces the elapsed time to back up a user catalog. Therefore, the user catalogs should be defined with a data control interval size of one half track.

Use the following procedure to recover a backup version of a catalog:

1. Purge all jobs that allocate the volume.
2. Issue the F CATALOG,OPEN(*volser*) command to determine which catalogs on the target volume are allocated.
3. Issue, for each catalog that is open:

   F CATALOG,UNALLOCATE(*catalogname*.)

   to cause the catalog address space to close the catalog and deallocate it.
4. Issue the RECOVER command to DFSMShsm.

If DFSMSdfp is installed, aliases of the catalog are backed up and recovered. When DFSMShsm attempts to recover a catalog, the following rules apply:

- If the catalog does not exist, it is defined, along with its aliases, from the backup version.
- If the catalog is empty, any aliases for that catalog in the system catalog are not deleted. Aliases that were backed up are defined for the recovered catalog if they do not already exist in the system catalog.
- If the catalog is not empty, it is deleted and redefined from the backup version. Any aliases for that catalog in the system catalog are not deleted. Aliases that were saved with the backup version are not added to the system catalog.

The recovery of a catalog has the following restrictions:

- A catalog cannot be recovered with a new name.
- Aliases in the catalog are not backed up or recovered unless DFSMSdfp is installed on the system.

- If you need to recover the master catalog, you must recover it on a different z/OS image.
- You must be a DFSMShsm-authorized user.
- If you do not use RACF discrete profiles for VSAM data sets, you can recover the catalog to the volume of your choice. If, however, you *do* use RACF discrete profiles for some VSAM data sets, the following restrictions apply:
  - If the catalog contains entries for VSAM data sets that are protected by RACF discrete profiles, we recommend that you recover the catalog to the same volume it was on when it was backed up. If the catalog is recovered to a different volume, all VSAM data sets that were protected by a discrete profile are now either unprotected or protected by a generic profile that might not provide the level of access and security desired.

    The RACF profile of a VSAM data set reflects the catalog volume. DFSMShsm does not update these profiles when it recovers a catalog. To restore the level of protection that a discretely-protected VSAM data set had before the catalog was recovered to a different volume, you have to update the data set's discrete profile to indicate the new volume. The task of deciding whether to update a discretely-protected VSAM data set's discrete profile, and making the update, can be left to the owner of the data, if an installation so desires.

    When recovering non-SMS-managed catalogs, you can use the TOVOLUME parameter to specify the volume from which the catalog was backed up.

    When recovering SMS-managed catalogs, you can assign a storage class with the GUARANTEED SPACE attribute as a means of requesting that the catalog is recovered to its original volume.
  - The HRECOVER command will fail if an ALIAS is substituted for the data set name of a catalog, even if the user is DFSMShsm-authorized.

Restoring a Catalog from a Dump Copy

You cannot use DFSMShsm to restore a catalog from a dump copy. DFSMShsm issues a LOCATE for the catalog name before invoking DFSMSdss, which causes the catalog to be opened. DFSMSdss does not restore a catalog that is open. You can, however, use DFSMSdss directly to restore a catalog from a DFSMShsm-owned dump copy.

Use the following procedure to restore a catalog:

1. Use the procedure in section "Restoring Data Sets or Volumes Directly with DFSMSdss" on page 427 to find the appropriate dump copy and to invoke DFSMSdss.
2. Ensure that all dump tape volumes are available at the time of the restore.
3. Follow the procedure, with one exception, described in "Backing Up and Recovering a Catalog" on page 440 for recovering a backup version of a catalog. The exception is in the last step. Instead of starting DFSMShsm and recovering the catalog, use the DFSMSdss RESTORE command to restore the catalog.

Note: You cannot restore a catalog that was dumped with compression, as described in "Invoking Full-Volume Dump Compression" on page 426.

Full-Volume Dump Procedures

This section explains:

- Interrupting and restarting dump procedures
- Defining a new automatic dump start window

Interrupting and Restarting Automatic Dump Processing

You can use the following methods to interrupt automatic dump processing during the three phases of automatic dump. These three phases are:

- Automatic deletion of expired dump copies
- Automatic dump of DFSMShsm-managed or owned volumes
- Deletion of excess volume table of contents (VTOC) copy data sets

There are four methods you can use to interrupt automatic dump processing:

- **Hold dump processing** — You can hold dump processing by specifying the DUMP, DUMP(AUTO), or ALL parameter of the HOLD command. 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 has finished 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 has completed. 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 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 NOBACKUP parameter of the SETSYS 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 has finished processing the current dump generation. If you disable dump processing while DFSMShsm is dumping DFSMShsm-managed volumes, DFSMShsm dump processing ends the next time 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 EMERGENCY parameter of the SETSYS 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 has finished processing the current dump generation.

  If you are in a multiple DFSMShsm-host environment, and have enabled the secondary host promotion function, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you have 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 DFSMShsm-managed volumes, DFSMShsm dump processing ends the next time 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 has finished processing the current dump generation. If you shut down DFSMShsm while DFSMShsm is dumping

DFSMShsm-managed volumes, DFSMShsm dump processing ends the next time DFSMSdss reads or writes a record. DFSMShsm does not start any new volume dump tasks.

If you are in a multiple DFSMShsm-host environment, have enabled secondary host promotion, and you have issued a STOP PROMOTE command, another DFSMShsm host takes over the unique functions of the primary DFSMShsm host that you have stopped. Those functions include automatic movement of backup versions from ML1 to tape, and 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 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 DFSMShsm-managed volume under automatic dump control. Automatic dump restarts at the point of interruption if less than 24 hours have passed since automatic dump was last started from the beginning and if the time is still within the dump start window.

Related reading: For additional information about using the secondary host promotion function, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Defining a New Automatic Dump Start Window

As you increase the number of volumes that DFSMShsm dumps, you might want to change the automatic dump start window. In a test environment, you might want automatic dump to run more than once on a single day.

This section describes:

- Defining a new start window when automatic dump is running
- Defining a new start window when automatic dump is not running

Defining a New Start Window When Automatic Dump Is Running

You can change the automatic dump start window at any time, even when automatic dump is currently running on the particular DFSMShsm host. Automatic dump runs when the new planned start time occurs if the currently running automatic dump completes before the planned start time of your newly defined start window. If the currently running automatic dump runs past the new planned start time, automatic dump will not start automatically until the next time the planned start time occurs. However, if automatic dump is released or DFSMShsm is restarted during the newly defined automatic dump start window, automatic dump starts unless it is prevented by one of the commands discussed in

"Interrupting and Restarting Automatic Dump Processing" on page 442 or by denying permission to the SETSYS REQUEST command.

Defining a New Start Window When Automatic Dump Is Not Running

If you define a planned start time that is later than the time that automatic dump last ended, automatic dump starts at the planned start time.

If you define a planned start time that is earlier than the time that automatic dump last ended, automatic dump starts only if it has not run to completion on this day.

Part 4. DFSMShsm Support Activities

In addition to the activities that cause the DFSMShsm program to perform its space management and availability management functions, certain other activities maintain the security and efficiency of DFSMShsm itself. For example, backup copies of the control data sets should be kept, and the DFSMShsm-owned volumes must be maintained.

Organization of This Part

This part is divided into the following chapters.

Task	See Page
Chapter 12, "Maintaining DFSMShsm Control Data Sets"	447
Chapter 13, "Managing DFSMShsm Media"	483
Chapter 14, "Obtaining Information from DFSMShsm"	517
Chapter 15, "DFSMShsm Tape Libraries"	535
Chapter 16, "Using DFSMShsm with DFSMSrmm"	537
Chapter 17, "Operating Procedures for DFSMShsm"	539
Chapter 18, "Purposes of DFSMShsm Commands"	567
Chapter 19, "RACF Considerations"	573

© Copyright IBM Corp. 1984, 2004

Chapter 12. Maintaining DFSMShsm Control Data Sets

DFSMShsm maintains three control data sets (CDSs):
- Backup control data set (BCDS)
- Migration control data set (MCDS)
- Offline control data set (OCDS)

These control data sets and the DFSMShsm journal are what make it possible for you to rely on DFSMShsm's record keeping to perform all of its activities. Therefore, you need backup copies of them to allow you to recover data should the active control data sets become lost or damaged.

DFSMShsm provides for automatic backup of the control data sets and the journal each time that automatic backup is performed for the system. In addition, you can back up the control data sets by command.

The primary host is the only host that normally performs an automatic CDS backup. If the primary host fails, a secondary host can take over the functions of the failed primary host.

To enable you to recover a lost or damaged control data set, DFSMShsm maintains a journal of each transaction that occurs to the control data sets. Should a control data set be damaged or lost, you can recover a backup version of the control data set and apply the transactions recorded in the journal to bring the control data set back to current status.

In addition to showing how to manage the control data sets automatically, the method of issuing commands necessary to determine the condition of the data sets and to repair them if necessary is discussed.

Related reading:
- For more information about secondary host promotion, see the sysplex chapter in *z/OS DFSMShsm Implementation and Customization Guide*.
- For more information about control data set placement and sizing, see *z/OS DFSMShsm Implementation and Customization Guide*.

Tasks for Maintaining the Control Data Sets

The tasks for controlling automatic backup of the control data sets require adding DFSMShsm commands to the ARCCMD*xx* member. The tasks for retrieving the backup control data sets require the invocation of CDS recovery utilities. The following tasks maintain the control data sets:
- Code SETSYS CDSVERSIONBACKUP to tell DFSMShsm how to back up the data sets automatically.
- Manually back up the control data sets using the BACKVOL CDS command, when automatic backup is not used.
- Recover the control data sets by using the enhanced CDS recovery utilities.
- Issue a DFSMShsm command to determine if the control data sets match the contents of the computing system catalogs and the VTOCs of the DASD volumes.
- Issue commands to repair errors in the control data sets.

© Copyright IBM Corp. 1984, 2004

Specifying How DFSMShsm Backs Up the Control Data Sets

You can use the CDSVERSIONBACKUP parameter of the SETSYS command to control backup of the control data sets. Within the CDSVERSIONBACKUP parameter, subparameters specify the number of backup versions to keep for the control data sets, the device type on which to store the backup versions, whether to run CDS backup in a parallel or nonparallel mode, the names of the backup data sets, and the data mover.

Note: If you are converting from a SETSYS NOCDSVERSIONBACKUP environment, you no longer need the MIGCATBU, BAKCATBU, and OFFCATBU DD statements in your DFSMShsm startup procedure.

Allocating Backup Data Sets

Whether you choose DFSMSdss or IDCAMS as the data mover when copying the control data sets, the backup data sets are allocated in the following manner:

- If backing up to DASD, you must preallocate the backup data sets.
- If backing up to tape, the data sets are dynamically allocated by DFSMShsm.

The backup data set names are given a low-level qualifier of either D*xxxxxxx* (if DFSMSdss is used for copying) or V*xxxxxxx* (if IDCAMS is used for copying), where *xxxxxxx* is the version number. Backup data set names for multicluster CDS data sets are given a low-level qualifier of DS*y*.D*xxxxxxx*, where *y* is the volume number and *xxxxxxx* is the version number. When CDS version backup fails, the data set name of the failed backup is given a low-level qualifier of X*xxxxxxx*, where *xxxxxxx* is the version number. This data set naming convention supports the use of automated CDS-recovery procedures that determine the latest good control data set backup versions from catalog information.

Choosing the Data Mover to be Used

The SETSYS CDSVERSIONBACKUP subparameter DATAMOVER allows you to indicate which of the following datamovers that DFSMShsm can use to back up the control data sets. (The DFSMShsm journal is always backed up by sequential I/O.)

- DATAMOVER(DSS) — tells DFSMShsm to use DFSMSdss logical dump to back up the control data sets
- DATAMOVER(HSM) — tells DFSMShsm to use IDCAMS EXPORT to back up the control data sets

When you use DFSMSdss—rather than IDCAMS—to copy the data, concurrent copy can be used, and the CDSs will be validated as they are backed up. Concurrent copy allows the CDS backup to complete as soon as the journal has been backed up, so that DFSMShsm can begin processing requests before the CDS backups are completed.

Rule: If you access the CDSs in RLS mode, you must use DFSMSdss as the data mover.

If you use IDCAMS, you need to examine the number of records copied each day (check the record count in message IDC0005I from IDCAMS EXPORT), because a broken VSAM index for a CDS will likely result in significant data loss in the backup copy. This loss can go unnoticed for a considerable period of time.

Recommendation: DFSMShsm development strongly recommends that you specify **DATAMOVER(DSS)**, which provides advantages of validation during backup as well as (with the concurrent copy function) reduced serialization of the control data sets. DATAMOVER(HSM) is retained as an option for purposes of coexistence with previous releases.

DFSMSdss as Data Mover

You can use DFSMSdss and the concurrent copy function to back up the control data sets. Concurrent copy lets you make a *point in time* copy of a control data set even while it is being updated. DFSMSdss also validates the CDS while the CDS is being backed up. This function is similar to AMS EXAMINE.

The use of the concurrent copy function with DFSMSdss significantly reduces the duration of complex-wide serialization of the DFSMShsm control data set resources when performing CDS version backup. This reduction increases the availability of DFSMShsm functions. To achieve maximum DFSMShsm availability, you should place the control data sets on SMS-managed devices that support the concurrent copy function, and you should use DFSMSdss to back up the control data sets. If DFSMSdss backs up to tape, each resource control data set is copied to an individual tape drive (in parallel).

If a control data set is on an SMS-managed device that supports the concurrent copy function, and the management class attribute BACKUP COPY TECHNIQUE is specified as PREFERRED or REQUIRED, DFSMSdss establishes a concurrent copy session. (See "Concurrent Copy Management Class Support" for information about the BACKUP COPY TECHNIQUE management-class attribute.) Otherwise, DFSMShsm directs DFSMSdss to dump the data set without the benefit of concurrent copy. All control data sets, if they are SMS-managed, should be specified with the same SMS backup copy technique.

After all resources have been set up (data sets allocated and tapes mounted), the control data sets are reserved. When using RLS in a multihost environment, DFSMShsm does an exclusive enqueue rather than a reserve. The reserves, or enqueues, prevent other DFSMShsm functions that require the control data sets from running.

DFSMShsm removes the CDS serialization after the following actions have been done:

1. The journal has been backed up.
2. DFSMSdss indicates that it has established a concurrent copy session for each control data set to be backed up.
3. The journal has been nulled (emptied).

All current DFSMShsm-supported backup DASD and tape devices are supported by the concurrent copy function.

Concurrent Copy Management Class Support: When using DFSMSdss as the data mover, the BACKUP COPY TECHNIQUE management-class attribute that is assigned to the control data sets informs DFSMShsm whether or not to direct DFSMSdss to use the concurrent copy function when backing up those data sets. (The attribute is used for a similar purpose during automatic or volume backup processing.)

The BACKUP COPY TECHNIQUE attribute is used as follows:

- CONCURRENT REQUIRED (R) — Specifies that the concurrent copy facility is required. If a concurrent copy session cannot be established, the data sets will not be backed up.
- CONCURRENT PREFERRED (P) — Specifies that the concurrent copy function is preferred. If a concurrent copy session can be established, the data sets are backed up using the concurrent copy function. Otherwise, DFSMSdss logical dump (without concurrent copy) is used.
- STANDARD (S) — Specifies that the concurrent copy function is not to be used. DFSMSdss logical dump without concurrent copy processing is used.

If the management class associated with a data set specifies that the concurrent copy BACKUP COPY TECHNIQUE (either REQUIRED or PREFERRED) be used, DFSMSdss is invoked specifying the CONCURRENT parameter.

If the management class BACKUP COPY TECHNIQUE is specified as CONCURRENT REQUIRED and a concurrent copy session cannot be established, the backup for that data set fails.

If the management class BACKUP COPY TECHNIQUE is specified as CONCURRENT PREFERRED and a concurrent copy session cannot be established, the data set is backed up without using the concurrent copy feature.

DFSMShsm as Data Mover

When DFSMShsm (IDCAMS) is used as the data mover, the CDSs are not validated during backup, and concurrent copy is not available. Thus you need to examine the number of records copied each day, because a broken VSAM index for a CDS will likely result in significant data loss in the backup copy. This loss can go unnoticed for a considerable period of time.

To specify DFSMShsm as the datamover, add the following command to the ARCCMD*xx* member for processing unit 1:

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

Specifying How Many Backup Versions to Keep

DFSMShsm keeps the backup copies of the control data sets as versions, where each version is a set of CDS backup data sets and a journal backup data set. The version number of each backup data set is shown in the low-level qualifier of the data set names. Unless all backups fail, the CDS backup version number is incremented by one, and the new backup data sets are created and cataloged each time you back up the control data sets.

Note: Previous to DFSMShsm, a CDS backup version was considered invalid unless *all* the data set backups were successful. With DFSMShsm, you may have one or more invalid backup data sets in a backup version. If you back up your control data sets to DASD and fail to attend to any CDS backup failures, you can eventually lose the ability to recover the control data sets.

Whenever a new CDS backup version is created, the oldest backup version is eliminated from the system (rolled off). If you back up to tape, the oldest backup version is uncataloged. If you back up to DASD, the oldest backup version is reused and renamed to the new version.

You must decide how many versions to keep on your system, and then tell DFSMShsm that number by using the BACKUPCOPIES subparameter of the SETSYS CDSVERSIONBACKUP command. You should keep a minimum of three CDS backup versions. Specify that number to DFSMShsm with the following command:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(3))
```

Specifying the Names of the Backup Data Sets

Use subparameters of the SETSYS CDSVERSIONBACKUP command to specify the data set names that are to be assigned to the backup data sets. You can use up to 35 characters in the data set name, including periods, but you must not end the name with a period. DFSMShsm appends a qualifier to the data set name that you specify. The form of this qualifier depends on the data mover used to perform the backup:

- DS*y*.V*nnnnnnn* — V specifies that IDCAMS was the data mover
- DS*y*.D*nnnnnnn* — D specifies that DFSMSdss was the data mover
- DS*y*.X*nnnnnnn* — X specifies that the backup failed, regardless of the data mover

Note: *nnnnnnn* is the version number. *y* is the volume for a multicluster CDS. DS*y* does not exist if the CDS is only one volume.

If you do not specify the data set names, DFSMShsm generates names of *uid*.BCDS.BACKUP, *uid*.MCDS.BACKUP, *uid*.OCDS.BACKUP, and *uid*.JRNL.BACKUP for the data sets. DFSMShsm substitutes the UID from the startup procedure for *uid*.

The following commands, used in the example system and added to the ARCCMD*xx* member in processing unit 1, demonstrate DFSMShsm processing when the data set names are not specified:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM) -
  BACKUPCOPIES(3) -
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
  JRNLBACKUPDSN(BHSM.JRNL.BACKUP))
```

Specifying the Device Type for Backup of the Control Data Sets

The control data sets and journal data set can be backed up to either DASD or tape. Both DASD and tape have certain advantages in relation to each other. In deciding whether to use DASD or tape for backing up your control data sets and journal data set, consider the following factors:

- **Mounting** —You do not have to mount DASD. For tape backup, you must mount a tape for each control data set and the journal.
- **Allocation** —With DASD, you do not have allocation problems that are caused by no available drives. With tape, the allocation request for a scratch tape fails if no tape drives are available.
- **SMS-managed** — With DASD, the backup copies of your control data sets can be SMS managed.
- **Preallocation** — With tape, you need not preallocate backup data sets.

Whether you back up the control data sets to DASD or to tape, DFSMShsm issues messages ARC0742I and ARC0743I for each data set that is backed up. If the backups are in nonparallel mode, the messages are issued in pairs. However, if the backups are in parallel mode, all of the ARC0742I messages may be issued before the first ARC0743I message is issued. Parallel processing makes it possible to start all the backups before the first backup is completed.

If you decide to back up the control data sets and journal data set to DASD, you must preallocate the data sets. Also, allocate the different versions of the backup data sets on different volumes. This better ensures that you can recover your backup versions.

In the example system, backup is shown going to DASD. Therefore, the command that is added to the ARCCMD*xx* member in processing unit 1 is modified as follows:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
   BACKUPCOPIES(3) -
   BACKUPDEVICECATEGORY(DASD) -
   BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
   MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
   OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
   JRNLBACKUPDSN(BHSM.JRNL.BACKUP))
```

Unlike backup of user data sets, backup of the control data sets must be to only one kind of I/O device. You cannot use a mixture of DASD and tape.

Specifying DASD for Backup

If you specify BACKUPDEVICECATEGORY(DASD), parallel processing is always performed.

Because the backup of the control data sets to DASD is in parallel mode, all of the resources necessary for the backup are allocated before the control data sets and the journal are reserved. This means that DFSMShsm can not be perform its other functions until either backup is complete or (if concurrent copy is used) concurrent copy sessions have been established and journal backup is complete.

Specifying Tape for Backup

Parallel backup CDS processing to tape allows the control data set and journal backups to be performed concurrently, thereby reducing the time required for the CDS backup function.

If you specify BACKUPDEVICECATEGORY(TAPE), your datamover is DFSMShsm, and you want parallel backup processing, then you must specify PARALLEL, because NOPARALLEL is the default. When using DFSMSdss as the data mover and going to tape, parallel processing must be specified.

If CDS backup is invoked while the CDSs are accessed in RLS mode, DFSMSdss must be the datamover. If the backup is directed to tape, the PARALLEL parameter of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY) must be specified.

When backing up to tape in parallel mode, one tape device for each data set to be backed up is required. For instance, if all three control data sets and the journal are to be backed up, then four tape drives are required.

When the backup of the control data sets is in parallel mode, all of the resources necessary for the backup are allocated before the control data sets and the journal are reserved.

Protecting Control Data Set Backup Tapes

The DFSMShsm CDS backup tapes are supported for expiration protection only, as specified by the EXPIRATIONDATE or RETENTIONPERIOD subparameters of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE)) command. The SETSYS TAPESECURITY command does not affect the CDS backup tapes, and cannot be used to assign RACF or password protection as it can with other DFSMShsm-owned tapes. However, the DFSMShsm ARCCBEXT installation exit can be used to assign RACF protection to the CDS backup tapes, and the DFSMShsm ARCTVEXT installation exit can be used to remove RACF protection from the tapes when you want to return them to scratch.

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

Invoking the Control Data Set Backup Exit (ARCCBEXT)

After CDS backup completes, the *control data set backup exit* (ARCCBEXT) is invoked, if it is active.

Related reading: For additional information about the ARCCBEXT exit, see *z/OS DFSMS Installation Exits*.

Backing Up to Multiple DASD Data Sets

If you decide to back up the control data sets and journal data set to DASD, you must first allocate and catalog a backup data set for each backup version of the control data sets. If the CDS is multicluster, additional backup version data sets are required. See "Multicluster CDS Considerations" on page 456.

The following example describes the process of backing up the control data sets and journal data set to multiple DASD backup data sets. The example assumes you have single-cluster control data sets.

1. The following command has been specified to indicate the number of preallocated data sets for each control data set and the journal data set:

   ```
   SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
     BACKUPCOPIES(3) -
     BACKUPDEVICECATEGORY(DASD) -
     MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
     BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
     OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
     JRNLBACKUPDSN(BHSM.JRNL.BACKUP))
   ```

   BACKUPCOPIES(3) indicates the number of preallocated data sets for each backup data set.
2. Three backup data sets are allocated for each control data set and the journal data set backup copy. Assume the following 12 data sets are allocated:

Copy Number	Names of Allocated Backup Data Sets
1	BHSM.MCDS.BACKUP.V0000001 BHSM.BCDS.BACKUP.V0000001 BHSM.OCDS.BACKUP.V0000001 BHSM.JRNL.BACKUP.V0000001

Copy Number	Names of Allocated Backup Data Sets
2	BHSM.MCDS.BACKUP.V0000002 BHSM.BCDS.BACKUP.V0000002 BHSM.OCDS.BACKUP.V0000002 BHSM.JRNL.BACKUP.V0000002
3	BHSM.MCDS.BACKUP.V0000003 BHSM.BCDS.BACKUP.V0000003 BHSM.OCDS.BACKUP.V0000003 BHSM.JRNL.BACKUP.V0000003

Note: The final qualifier of each backup data set must start with V0000001 and increase by 1 for each required backup data set. Because you chose "DSS" as the data mover in the SETSYS CDSVERSIONBACKUP command, DFSMShsm changes the final qualifier of its output data sets to begin with the letter D (D0000001). The renaming of the backup data set final qualifier happens automatically when you change data movers.

3. DFSMShsm backs up the control data sets and journal data set by doing the following:
   a. It issues an exclusive enqueue against the control data set resource and, if it is in a multiple DFSMShsm-host environment, reserves the volumes containing the journal data set and each control data set being used.

      When using RLS in a multiple DFSMShsm-host environment, DFSMShsm does an exclusive enqueue rather than a reserve.
   b. It invokes DFSMSdss to logically dump each of the control data sets, using concurrent copy if available. DFSMSdss informs DFSMShsm when a concurrent copy session has been established, or when it has finished the dump of a control data set.
   c. It copies the journal to the backup data set and nulls the journal data set. When the journal has been backed up and all the concurrent copy sessions have been established or all the control data sets have been backed up, DFSMShsm releases the volume reserves (if using RLS there are no reserves), and dequeues the control data set resource.

      When DFSMShsm backs up the control data sets and journal data set, it determines the final qualifier of the backup data sets. If the version number in the final qualifier matches the version number in the final qualifier of an existing data set, DFSMShsm uses the existing data set. In this example, DFSMShsm finds an existing backup data set the first three times it backs up the control data sets, and changes the first letter of the last data set name qualifier from V to D (V0000001 becomes D0000001). The fourth time DFSMShsm backs up the control data sets, it cannot find an existing version 4 backup data set. Therefore, DFSMShsm backs up the control data sets to the backup data set with the final qualifier of D0000001.
   d. If DFSMShsm backs up the control data sets to DASD volumes and needs to rename the backup data set, it does so. In this example, after DFSMShsm successfully backs up these data sets for the fourth time, it renames the backup data set by changing the final qualifier of D0000001 to a final qualifier of D0000004. Even if the renaming fails, the backup copy is still valid.
   e. DFSMShsm optionally calls the control data set backup user exit ARCCBEXT, allowing you to process the backup version after DFSMShsm creates it.

Note: If you change the specification for the number of backup copies kept to *reduce* the number of copies, DFSMShsm does not delete the excess older backup versions that were made before you reduced the number of copies.

If you change the specification for the number of backup copies kept to *increase* the number of copies, you must first allocate and catalog additional backup data sets. Start with a final qualifier that is one more than the last final qualifier of your backup data sets on DASD.

Related reading: For additional information about allocating the MCDS, BCDS, OCDS, and journal data set, and their backup copies, see *z/OS DFSMShsm Implementation and Customization Guide*.

Backing Up to Multiple Tape Data Sets

If you decide to back up the control data sets and journal data set to multiple tape data sets, the process is a little different. When you specify the following command, you do not have to allocate and catalog backup data sets:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(copies) -
  BACKUPDEVICECATEGORY(TAPE))
```

You can choose to specify a unit name, density, and retention period or expiration date.

The following example describes the process of backing up the control data sets and journal data set to multiple tape backup data sets.

1. The following command is specified:

   ```
   SETSYS CDSVERSIONBACKUP (BACKUPCOPIES(3) -
     DATAMOVER(DSS) -
     BACKUPDEVICECATEGORY(TAPE))
   ```

   BACKUPCOPIES(3) indicates the maximum number of cataloged backup data sets for each control data set and the journal data set.
2. DFSMShsm backs up the control data sets and the journal data set. DFSMShsm puts only one backup data set on a specific tape. Because you do not have to preallocate backup data sets when you use tape, DFSMShsm dynamically allocates a new backup data set. The default backup data set name is the same for tape as for DASD. Therefore, if you have not specified the backup data set name with the SETSYS CDSVERSIONBACKUP command, DFSMShsm uses the default backup data set name and the correct final qualifier for the new backup data set. In this case—the first time DFSMShsm backs up the four data sets—the backup data sets have the following names:

   HSM.MCDS.BACKUP.D0000001
   HSM.BCDS.BACKUP.D0000001
   HSM.OCDS.BACKUP.D0000001
   HSM.JRNL.BACKUP.D0000001

   The second time DFSMShsm backs up the four data sets, it uses a final qualifier of D0000002.

   HSM.MCDS.BACKUP.D0000002
   HSM.BCDS.BACKUP.D0000002
   HSM.OCDS.BACKUP.D0000002
   HSM.JRNL.BACKUP.D0000002
3. The third time DFSMShsm backs up the four data sets, it dynamically allocates a third set of backup data sets and uses a final qualifier of D0000003.

   After DFSMShsm successfully backs up the control data sets and the journal data set for the fourth time, it uncatalogs the set of backup data sets whose

final qualifier is D0000001 and invokes installation exit ARCTVEXT for the rolled-off version of each successfully backed-up data set.

Note: If the backup is not successful for a particular data set, the oldest version of that data set is not rolled off, and DFSMShsm invokes the tape installation exit, ARCTVEXT, that allows you to reuse the bad backup tapes.

4. DFSMShsm optionally invokes the control data set backup user exit ARCCBEXT, allowing you to process the backup version after DFSMShsm creates it.

Notes:

1. The maximum number of tapes DFSMShsm uses for one backup version of one control data set or the journal is 18.
2. If you change the specification for the number of backup copies kept to reduce the number of copies, DFSMShsm does not delete or uncatalog the excess older backup versions that were made before you reduced the number of copies.

Multiple DFSMShsm-Host Considerations

The following are considerations in a multiple DFSMShsm-host environment:

- All hosts sharing DFSMShsm control data sets must specify the same options for the SETSYS CDSVERSIONBACKUP command, as the information is stored in the MCDS. The last DFSMShsm host to issue the SETSYS CDSVERSIONBACKUP command controls how all the DFSMShsm hosts back up the CDSs.

  An alternative to having all DFSMShsm hosts specifying the same options for the SETSYS CDSVERSIONBACKUP command is to have only one DFSMShsm host (usually the host identified as the primary DFSMShsm host) specifying the command.
- The NOCDSVERSIONBACKUP parameter of the SETSYS command is no longer valid. If SETSYS NOCDSVERSIONBACKUP is specified, message ARC0103I is issued, indicating an unsupported parameter.
- If backing up the control data sets to DASD, make the preallocated data sets accessible to any DFSMShsm host doing the backups.
- If backing up to tape, some of your tape data sets won't be rolled off if they were created on one z/OS image, deleted on another z/OS image, and cataloged other than on a shared user catalog.

Multicluster CDS Considerations

A multicluster MCDS or BCDS can be defined when the CDS size increases beyond one volume. If you have defined a multicluster CDS, you must make the following changes to your CDS backup procedure:

- When backing up your CDSs to DASD, preallocate each volume within the multicluster CDS using a different naming convention. Each low-level qualifier (version number) must be preceded by .DS*y*, where *y* is a number from 1 to 4 and represents the number of volumes in the multicluster CDS. For instance, if you have split the MCDS into four volumes, split the BCDS into two volumes, and are keeping only one backup copy, the following data sets would be preallocated (assuming the choice of DFSMSdss for the data mover):

  HSM.MCDS.BACKUP.DS1.D0000001
  HSM.MCDS.BACKUP.DS2.D0000001

HSM.MCDS.BACKUP.DS3.D0000001
HSM.MCDS.BACKUP.DS4.D0000001

HSM.BCDS.BACKUP.DS1.D0000001
HSM.BCDS.BACKUP.DS2.D0000001

HSM.OCDS.BACKUP.D0000001

HSM.JRNL.BACKUP.D0000001

- The interface to the ARCCBEXT installation exit has been extended to include the names of the backup data sets for the multicluster CDS.
- When you back up your CDSs to tape, the new naming convention (.DS*y*) applies to the created tape data set names.

  Note: The CDS backup tapes created under the old naming convention are not rolled off automatically. Message ARC0747I is issued to indicate that roll-off processing cannot be performed.
- Messages ARC0742I, ARC0743I, and ARC0744E indicate which volume in the multicluster CDS was being processed.
- When you back up your CDSs to tape in parallel, a tape drive is needed for each cluster in the multicluster CDS. For instance, if you split the MCDS into four volumes, split the BCDS into two volumes, have defined an OCDS on one volume, and are journaling, eight tape drives are required to back up the CDSs and the journal.

Related reading:

- For additional information about how to implement a multicluster CDS, see*z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the ARCCBEXT installation exit, see *z/OS DFSMS Installation Exits*.

Nulling the Journal

If the journal data set fails to back up during the CDS version backup function, the journal will not be nulled. Conversely, the journal is always nulled if the backup of the journal data set is successful, regardless of the success or failure of the backup of the control data sets.

You can null the journal by using the following command:

```
BACKVOL CDS(NULLJOURNALONLY)
```

where NULLJOURNALONLY is an optional subparameter telling the CDS version backup function to only null the journal. No backup copies of the control data sets or journal are made.

Changing the Names of the Backup Data Sets

You can change the naming conventions for the backup data sets by performing the following steps:

- Issue the SETSYS CDSVERSIONBACKUP command using the subparameter for the data set names assigned to the backup data sets. Specify the new naming convention used to back up the control data sets.

Rule: You must change the naming convention for the backup data sets in all procedures that DFSMShsm uses. For example, update the ARCCMD*xx* member of the DFSMShsm startup procedure if you have specified the SETSYS CDSVERSIONBACKUP command.

- Rename any current backup versions of the control data sets to the new naming convention if the backup data sets are on DASD. Also, you must rename any "unused" backup versions.
- Uncatalog the old backup data sets so the tape management function can reuse the tape if the backup data sets are on tape. During backup of the control data set, you might receive message ARC0747I, which indicates DFSMShsm could not uncatalog the oldest version using the new backup names. The tape backup copies are successfully made.

Changing BACKUPDEVICE from Tape to DASD

If you started your backup of the control data sets and the journal data set to tape and want to change to DASD, do the following:

1. Uncatalog the backup versions that reside on tape. If this is not done, attempts to back up the control data sets to DASD will fail.

   The expiration date specified by the EXPIRATIONDATE or RETENTIONPERIOD parameter is still in effect on your backup tapes. Your tape management system returns the tapes to scratch according to the expirations date. Note that some expiration dates, such as 99365 (which prevents the tape from ever expiring), may require manual deletion through your tape management system.
2. Specify the following command:

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

   You can also change the number of backup copies. However, this example uses the same number as in the tape example; that is, three backup copies.
3. Allocate your backup data sets on DASD. The sample job ALLOCBK1, which is located in the DFSMShsm starter set, allocates and catalogs the control data set backup version data sets on DASD volumes.

   The final qualifier must again increase by 1 for each set of backup data sets. However, do not start with a final qualifier of D0000001. Instead, start with a final qualifier that is 1 more than the last final qualifier of your backup data sets on tape.

   Issue the following command to determine the last final qualifier for the backup data sets:

   ```
   QUERY CDSVERSIONBACKUP
   ```

   In this example, the last final qualifier is D0000050.

   When you allocate your backup data sets, you decide to use the default backup data set names. Because you are backing up all three control data sets and the journal data set, you must allocate 12 backup data sets:

Data Set	Names of Allocated Backup Data Sets
MCDS	HSM.MCDS.BACKUP.D0000051 HSM.MCDS.BACKUP.D0000052 HSM.MCDS.BACKUP.D0000053
BCDS	HSM.BCDS.BACKUP.D0000051 HSM.BCDS.BACKUP.D0000052 HSM.BCDS.BACKUP.D0000053
OCDS	HSM.OCDS.BACKUP.D0000051 HSM.OCDS.BACKUP.D0000052 HSM.OCDS.BACKUP.D0000053

Data Set	Names of Allocated Backup Data Sets
Journal	HSM.JRNL.BACKUP.D0000051 HSM.JRNL.BACKUP.D0000052 HSM.JRNL.BACKUP.D0000053

DFSMShsm now backs up the control data sets and journal data set to backup data sets on DASD volumes.

Related reading: For additional information about the sample job ,ALLOCBK1, and the DFSMShsm starter set, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Changing BACKUPDEVICE from DASD to Tape

If you started your backup of the control data sets and journal data set to DASD and want to change to tape, specify the following command:

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

If you are using DFSMSdss as the data mover, specify:

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

You may change the number of backup copies. If you do not change the number of backup copies, the backup copies that are currently cataloged on DASD are uncataloged by DFSMShsm.

Note: DFSMShsm does not delete the DASD backup copies. The only function performed by DFSMShsm is to uncatalog these data sets. If you feel that you do not need the backup copies on DASD, you may scratch these data sets from your system after the appropriate number of backup copies have been made on a tape device media.

If you have already uncataloged the DASD backup copies, or if you change the number of backup copies, the attempt to uncatalog the DASD backup copies may fail during the backup of the control data sets and journal data set. An ARC0747I message is issued for each data set that could not be uncataloged, and the ARC0741I message at the completion of the backup of the control data sets and journal data set indicates that the backup was UNSUCCESSFUL. However, this unsuccessful status indicates only that backup versions were not successfully uncataloged. The tape backup copies will have been successfully made, and the final qualifier for these backup copies will have been successfully saved to be used for the next backup of the control data sets and journal data set. This scenario requires no action on your part.

Using the QUERY Command to Get CDS Information

Issue the DFSMShsm QUERY CDSVERSIONBACKUP or the QUERY SETSYS commands to determine whether DFSMShsm or DFSMSdss is being used to back up the control data sets.

Issue the QUERY CONTROLDATASETS command to display the CDS serialization technique in use, and the key boundaries being used for non-key-range multicluster CDSs.

Backing Up the Control Data Sets Manually

Occasionally, it may become necessary for you to manually back up the control data sets. This manual backup could be needed if CDS version backup fails to get a good copy of the control data sets, or if you receive a message that the journal is full. DFSMShsm provides parameters for the BACKVOL command that allow you to back up all the control data sets manually:

```
BACKVOL CDS(DATAMOVER(HSM | DSS)
  BACKUPDEVICECATEGORY(DASD | TAPE(NOPARALLEL | PARALLEL)))
```

where

DATAMOVER(HSM | DSS) specifies which CDS backup data mover should be used when backing up the control data sets.

BACKUPDEVICECATEGORY(TAPE | DASD) specifies the device on which the backup copies are recorded.

TAPE(NOPARALLEL | PARALLEL) specifies whether one or more tape devices are allocated. If DATAMOVER(DSS) is specified, TAPE(NOPARALLEL) cannot be specified.

Notes:

1. If you do not specify DATAMOVER with the BACKVOL CDS command, the last DATAMOVER specified by SETSYS CDSVERSIONBACKUP is the default.
2. If you do not specify BACKUPDEVICECATEGORY with the BACKVOL CDS command, the last BACKUPDEVICECATEGORY specified by SETSYS CDSVERSIONBACKUP is the default.
3. If you specify BACKVOL CDS DATAMOVER(DSS) without BACKUPDEVICECATEGORY, and the DFSMShsm CDS version backup environment indicates TAPE(NOPARALLEL), the BACKVOL command terminates with an error message indicating that the CDSVERSIONBACKUP options of DATAMOVER(DSS) and TAPE(NOPARALLEL) are in conflict.
4. Use caution if you change device types with the BACKUPDEVICECATEGORY parameter. See the sections on “Changing BACKUPDEVICE from Tape to DASD” on page 458 and “Changing BACKUPDEVICE from DASD to Tape” on page 459.

Recommendation: DFSMShsm development strongly recommends that you specify **DATAMOVER(DSS)**, which provides advantages of validation during backup as well as (with the concurrent copy function) reduced serialization of the control data sets. DATAMOVER(HSM) is retained as an option for purposes of coexistence with previous releases.

In a single DFSMShsm-host environment, DFSMShsm obtains an exclusive enqueue on the control data sets. In a multiple DFSMShsm-host environment, DFSMShsm reserves the volumes, or if using RLS, does an exclusive enqueue on the volumes on which the control data sets reside. Any other DFSMShsm activity is prevented from occurring during the backup of the control data sets. Therefore, do not back up the control data sets by command unless it is absolutely necessary.

Possible Failures during Backup

If a backup fails, DFSMShsm documents that failure with message ARC0744E. This section documents how to proceed after the most likely type of error, and more general suggestions for other error conditions.

Growth of Your Control Data Sets

Over time, the number of data sets migrated and backed up, the number of tape volumes owned by DFSMShsm, and so on, typically increase, causing your control data sets to grow in size. A control data set may grow such that the space allocated to its DASD backup copy is no longer enough, producing return code 12, 13, 18, 47, or 48 in message ARC0744E during an attempted backup.

Assuming from analyzing messages from DFSMSdss or Access Method Services that a backup copy is full, proceed as follows:

1. Issue the QUERY CDSVERSIONBACKUP command. The resulting message ARC0376I, for example,
   ARC0376I
   BACKUPCOPIES=3,
   BACKUPDEVICECATEGORY=DASD,
   LATESTFINALQUALIFIER=V0000170,
   DATAMOVER=HSM
   gives you the latest version qualifier, and how many backup copies need to be reallocated.
2. Examine the message for the control data set that is identified in the ARC0744E message and determine which backup copies (MCDS, BCDS, OCDS, or JRNL) must be made larger. Assume that the backup copies of the BCDS must be reallocated.
3. Rename the backup data sets

   *uid.*BCDS.BACKUP.V0000168 through *uid.*BCDS.BACKUP.V0000170

   so they can be kept until there are successful backups.
4. Compare the (used) size of the current BCDS to its size when the backup data sets for the BCDS were last allocated. This allows you to get a rough idea of how much larger the backup data sets must be.
5. Use the ratio from step 4, and reallocate backup data sets

   *uid.*BCDS.BACKUP.V0000168 through *uid.*BCDS.BACKUP.V0000170

   to their new larger sizes.
6. Issue the commands

   RELEASE BACKUP
   BACKVOL CDS

   If the failed backup was for the journal, DFSMShsm will have inhibited journaling. A successful BACKVOL CDS command should back up the journal, then null the online journal.
7. Issue commands to RELEASE the other DFSMShsm functions that you are using once the BACKVOL command is successful.
8. Delete the backup data sets that were renamed in step 3 when DFSMShsm makes three successful backups. A successful backup generates the following message as part of the output from a QUERY CDSV:
   ARC0376I . . .

LATESTFINALQUALIFIER=V0000173,

Other Failures

By considering the return code in message ARC0744E, you can determine which type of failure occurred:

- If there is an indication of an error in the structure or contents of the control data set itself, perform the following steps:
  1. Take the action to correct the error as indicated by the error message number and the associated return code. This information is contained in the MVS System Messages document.
  2. You might need to recover the CDS to a previous level.
- If there is an indication of an error in the system environment outside of the control data set (for example, an output tape could not be allocated), perform the following steps:
  1. Stop DFSMShsm if necessary (if, for example, the error was fragmented storage such that IDCAMS could not be loaded).
  2. Correct the environmental condition, if necessary.
  3. Restart DFSMShsm, if you stopped it.
  4. Issue the **BACKVOL CDS** command to make up for the backup that failed.
- If there is an indication of a probable program error, gather the necessary data for contacting the IBM Support Center.

Related reading:

- For additional information about the action to take regarding error messages and return codes, see *z/OS MVS System Messages, Vol 1 (ABA-AOM)*.
- For additional information about how to recover the CDS to a previous level, see "Recovering Control Data Sets Using the Enhanced CDS Recovery Function" on page 463).

Reasons For Recovering the Control Data Sets

Occasionally a control data set becomes unusable by DFSMShsm. For example, a hardware failure or a software logical failure may occur after which DFSMShsm is unable to use a control data set.

When the control data sets are backed up, they constitute a snapshot of the conditions in the system at that time. After the control data sets are backed up, as DFSMShsm continues to process data sets and volumes, the control data sets are continually changing. To make it possible for you to return the control data sets to their condition just before the error, DFSMShsm makes a journal record of each transaction that occurs in the control data sets. If a control data set becomes lost or damaged, you can return it to its correct condition by recovering a backup version of the control data set and applying the appropriate journal.

The CDS recovery process has variations based on which data mover has been used to back up the control data sets. The low-level (final) qualifier of the name of the control data set backup data set indicates which data mover has been used to back it up. A low-level qualifier of Vnnnnnnn indicates that DFSMShsm backed up the data set. A low-level qualifier of D*nnnnnnn* indicates that DFSMSdss backed up the data set, and a low-level qualifier of X*nnnnnnn* indicates that CDS version backup of the data set failed, regardless of whether DFSMShsm or DFSMSdss was doing the backup.

I/O Errors

When an I/O error occurs on the control data sets, the following situations can exist:

- If the I/O error is caused by an equipment check on the real DASD containing the affected data set, recover the data set to a different volume. Call your service representative to repair the drive on which the equipment check has occurred.
- If the I/O error is caused by a data check on the real DASD containing any control data set, you can use the enhanced CDS recovery function to recover the backup version of the affected data set to a different volume. You can use Device Support Facilities (a program to perform operations on disk volumes) to analyze, inspect, and assign alternate tracks and then recover the affected control data set to the same volume. Or, you can rename the affected control data set and define a new control data set on the same volume.

Recovering Control Data Sets Using the Enhanced CDS Recovery Function

The recommended procedure for recovery is called the "enhanced CDS recovery function", which allows you to recover a control data set to the point of failure (to the point where the hardware failure occurred or to the point where the software failure was recognized).

The enhanced CDS recovery function consists of the following two utility programs:

ARCBJRNL
The journal copy utility that backs up the journal—using as input either the active journal or the active journal plus multiple journal backup copies—and prepares the resultant backed up copy that is sorted and used as input to the import utility.

Rule: Because there may still be residual data in the journal, you must run the sort job regardless of how many records that ARCBJRNL reports were processed.

ARCIMPRT
The import utility that uses a sorted version of the backed up and sorted journal or journals from the journal copy utility and an exported control data set to recover the control data set to the point of failure.

Requirements for Using the Enhanced CDS Recovery Function

To successfully use the enhanced CDS recovery function, you must be sure that:

- You have sufficient DASD to contain the rebuilt control data set and the backed up journal.
- DFSMShsm is not active on any z/OS image in the complex while the control data set is being recovered.
- You have specified SETSYS JOURNAL for normal DFSMShsm operation, because an up-to-date journal is necessary for the enhanced CDS recovery function.
- You run the ARCIMPRT utility separately for each specific control data set to be recovered because the parameter passed to ARCIMPRT is for a specific control

data set to be recovered. Also, the EXPORT DD statement must be changed to the particular exported control data set being used.

If you are recovering a multicluster CDS, you need recover only the specific volume which failed; you do not need to recover all the volumes within the multicluster CDS.

In other words, if a migration control data set is to be recovered, then the parameter passed to ARCIMPRT must be MCDS or MIGRATIONCONTROLDATASET, and the EXPORT DD statement *dsname* must refer to the portable-exported-migration CDS (your backup copy of the control data set).

Implementing Enhanced CDS Recovery using JCL

This section describes the steps necessary to recover a CDS. You can recover a CDS either from the most recent backup copy or from an older backup copy. An older backup copy is needed if your most recent backup copy is unusable. Both types of recoveries are essentially the same except for Step 1 (Create the Backup Journal).

Steps 1, 2, and 4 can be run separately as shown or can be combined into one job.

The following sample JCL shows the steps required to recover a control data set:

1. Create the backup journal.

   The backup journal can be created by using either of the two following JCL examples.

   This example uses only the active journal as input and is used if recovering from the most recent backup copy of your CDS. If the most recent backup copy is a failed copy (last qualifier is Xnnnnnnn) then follow the procedures which use an active journal and one or more backup journals.

```
//ARCBJRN JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K
//* ------------------------------------------------------
//* CREATE DATA SET CONTAINING RECORDS FROM THE
//* ACTIVE JOURNAL
//* ------------------------------------------------------
//STEP01  EXEC PGM=ARCBJRNL
//STEPLIB   DD DSNAME=location where program ARCBJRNL resides,
//             DISP=SHR
//SYSPRINT  DD SYSOUT=*
//JRNLO     DD DSNAME=online journal data set name,
//             DISP=OLD
//JRNLBK    DD DSNAME=backup journal name,
//             UNIT=unit name,
//             VOL=SER=volume where backup journal will reside,
//             SPACE=adequate space to hold journal,
//             DISP=(,CATLG,DELETE)
```

   The following example uses the active journal and three existing journal backup copies as input. This example is used if recovering from an older backup copy of your CDS. In this case, the backup CDS is from 4 backups ago.

   Notes:

   a. You can indicate what input—the active journal, journal backup copies, or both—is used during ARCBJRNL utility processing by using the PARM keyword on the EXEC statement. The keywords and their meanings are:
      - JRNLONLY — Only the active journal is used as input
      - JRNLBACKUP — Only journal backup data sets are used as input

- JRNLBACKUPS — Only journal backup data sets are used as input
- BOTH — Both the active journal and journal backup data sets are used as input

If PARM is not specified, only the active journal is used as input to ARCBJRNL processing.

b. When using multiple journal backup data sets for input to ARCBJRNL processing, list these data sets in ascending version order in the ARCBJRNL BKUPJRNL DD DSNAME statements.

```
//ARCBJRNL JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K
//* ------------------------------------------------------
//* CREATE DATA SET CONTAINING JOURNAL FROM JOURNAL
//* BACKUP COPY DATA SETS AND THE ACTIVE JOURNAL
//* ------------------------------------------------------
//STEP01    EXEC PGM=ARCBJRNL,PARM='BOTH'
//STEPLIB   DD DSNAME=location where program ARCBJRNL resides,
//             DISP=SHR
//SYSPRINT  DD SYSOUT=*
//JRNLO     DD DSNAME=online journal data set name,
//             DISP=SHR
//BKUPJRNL  DD DSNAME=backup copy name and version,DISP=SHR
//          DD DSNAME=backup copy name and version,DISP=SHR
//          DD DSNAME=backup copy name and version,DISP=SHR
//JRNLBK    DD DSNAME=backup journal name,
//             UNIT=unit name,
//             VOL=SER=volume where backup journal will reside,
//             SPACE=adequate space to hold journal,
//             DISP=(,CATLG,DELETE)
```

2. Sort the journal backup file.

   After you have created the journal backup file using the JCL in step 1, sort the file by KEY using the EQUALS parameter, as in the following JCL. This technique *must* be used when sorting journal records created by DFSMS/MVS Version 1 Release 4 and subsequent releases. It can also be used by prior DFSMS/MVS releases.

   Note: The following JCL assumes you have a SORTD cataloged procedure. The SORTD cataloged procedure can be created by running the DFSORT SORTPROC job that DFSORT distributes in SYS1.SAMPLIB. If you do not have a SORTD procedure, modify the EXEC card in the JCL to identify the sort program you want to use. Ensure that the program is logically equivalent to DFSORT.

```
//SORT JOB
//* ---------------------------------------------------
//* SORT THE BACKED UP JOURNAL
//* ---------------------------------------------------
//STEP02  EXEC SORTD
//SORTLIB   DD .................
//SYSOUT    DD SYSOUT=*
//SORTWK01  DD ........................
//SORTWK02  DD ........................
//SORTWK03  DD ........................
//SORTWKnn  DD ........................
//SORTIN    DD DSNAME=backup journal name from STEP 1,
//             DISP=SHR
//SORTOUT   DD DSNAME=sorted backup journal name,
//             UNIT=unit name,
//             VOL=SER=volume where sorted backed up journal will reside,
//             SPACE=adequate space to hold sorted backed up journal,
//             DCB=(RECFM=VBS,BLKSIZE=0,LRECL=6560),
//             DISP=(,CATLG,DELETE)
//SYSIN     DD *
  SORT  FIELDS=(17,44,CH,A)
        OPTION EQUALS
/*
```

3. The next step depends on the data mover used to create the backup copy. If the backup copy has been made by DFSMSdss:
   a. Restore the backup version (either the most recent backup version or the last good backup version) using the following example JCL:

```
//JOB1 JOB accounting information,REGION=nnnK
//STEP1     EXEC  PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE      DD    DSN=uid.CDS.BACKUP.D0000106,DISP=(OLD,KEEP)
//DASD      DD    UNIT=3390,VOL=SER=222222,DISP=OLD
//SYSIN     DD    *
             RESTORE DATASET(-
             INCLUDE(uid.dsname))-
             INDDNAME(TAPE)-
             OUTDDNAME(DASD)-
             REPLACE
/*
```

   b. Export the data set restored in step 3a using the following example JCL:

```
//EXPORT JOB accounting information,REGION=nnnnK
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTDD1   DD DISP=(,CATLG,DELETE),BUFNO=26,BLKSIZE=28332,
//         SPACE=(CYL,(200,20),RLSE),
//         UNIT=3390,VOL=SER=222222,
//         DSN=temp_dsname
//SYSIN    DD *
           EXPORT dsname -
           OUTFILE(OUTDD1) -
           TEMPORARY
/*
```

   If the backup copy was made by DFSMShsm, that backup copy is already in the form of a portable exported data set.
4. Import and merge the control data set.

After you have sorted the journal backup file by doing step 2 and you have an exported copy of the subject CDS, run ARCIMPRT to create a control data set by using the following JCL.

Notes:

a. If you need to increase the size of the control data set, issue the DEFINE command before the IMPORT command.

b. Any valid IDCAMS command may be specified in the SYSIN stream. These commands are passed directly to IDCAMS (program ARCIMPRT does not modify these commands). Therefore, the DELETE command is used to delete the control data set before the data set is imported.

c. The IMPORT command must use the INFILE parameter; using the INDATASET parameter will result in an error return.

d. The following sample shows how to re-create a migration control data set. To re-create a backup control data set or an offline control data set, make changes to the "PARM=" in the EXEC statement and to the EXPORT DD statement *dsname* that reflects which control data set exported copy to use. Also make changes to the OUTDATASET name in the IDCAMS control statement that reflects which control data set is to be re-created.

```
//ARCIMPRT JOB
//* -------------------------------------------------------
//* RE-CREATE THE MCDS
//*   Note the parameter in the following execute statement
//*   specifying which CDS to recover.
//*   Also, any valid IDCAMS command may be specified in the
//*   SYSIN stream below.
//* -------------------------------------------------------
//STEP03  EXEC PGM=ARCIMPRT,PARM='MCDS'
//STEPLIB   DD DSNAME=where ARCIMPRT resides,
//             DISP=SHR
//SYSPRINT  DD SYSOUT=*
//AMSDUMP   DD SYSOUT=*
//JOURNAL   DD DSNAME=name of sorted backup journal from STEP 2,
//             DISP=SHR
//EXPORT    DD DSNAME=name of portable exported MCDS from STEP 3,
//             DISP=SHR
//SYSIN     DD *
  DELETE (MCDS cluster name) CLUSTER PURGE
  IMPORT INFILE(EXPORT) OUTDATASET(MCDS cluster name)
/*
```

5. Restart DFSMShsm from the processing unit from which you issued ARCIMPRT.
6. Verify that the control data sets are correct. See "Determining If Control Data Sets Are Accurate" on page 472.
7. Fix any errors in the control data sets. See "Repairing Errors in the Control Data Sets" on page 473.
8. If you are recovering the MCDS, issue a QUERY CDSVERSIONBACKUP to ensure that the latest final qualifier is correct. You can determine the final qualifier of the latest backup copy from your catalog using the LISTCAT LEVEL command, as shown in the following example:

   LISTCAT LEVEL(*uid*)

   If these qualifiers do not match, use the FIXCDS patch command to correct the field at offset X'B1' in the MHCR record to reflect the latest version in the catalog, as shown in the following example:

FIXCDS S MHCR PATCH(X'B1' 0000007)

9. Use the CONTROLDATASETS parameter of the BACKVOL command to get a current backup copy of the control data sets and journal data set.
10. Restart DFSMShsm in the other processing units.

If multiple journal backup copies are needed to recover the journal, and therefore recover a control data set, the following procedure should be used, because it will minimize the time DFSMShsm is down:

1. Use the ARCBJRNL utility (with multiple backup copies as input) and the ARCIMPRT utility to recover the control data set to a point just before what the active journal contains.
2. Bring down DFSMShsm.
3. Use the ARCBJRNL utility (with only the active journal as input) and the ARCIMPRT utility to finish recovering the control data set.

ARCBJRNL And ARCIMPRT Return Codes

The following tables describe the return codes that are returned by the ARCBJRNL and ARCIMPRT utilities. These codes are returned in general register 15 and can be referred to in your JCL.

RETCODE	MEANING
0	Normal return
4	No journal 'ARCCR' control record (MSGARC0461I)
8	Only journal 'ARCCR' control record (MSGARC0461I)
28	Get macro I/O error
32	Put macro I/O error
40	On-line journal OPEN error
44	Backup journal OPEN error
48	EOV or ABEND error
50	I/O error occurred on journal backup input data set
54	Open error occurred on journal backup input data set
58	ABEND error occurred processing journal backup input data set
99	Input parameter error (invalid value specified on PARM keyword)

RETCODE	MEANING
0	Normal return
4	IDCAMS message: minor problems but function completed
8	IDCAMS message: function completed but some processing bypassed
12	IDCAMS message: entire function could not be performed
16	IDCAMS message: severe error or problem encountered
18	Invalid backup journal used as input to ARCIMPRT (MSGARC0463I)
20	Invalid or no parameter passed to ARCIMPRT (MSGARC0462I)
22	Backup journal file open error
24	Backup journal file I/O error
26	Export file OPEN error
27	No CDS records found in exported CDS
28	Export file I/O error
30	Sysprint file OPEN error
32	Sysprint file I/O error

RETCODE	MEANING
34	Did not read a header record preceding the CDS backup copy (this can occur if the IMPORT command uses the INDATASET parameter instead of INFILE).

Recovering Multicluster Non-Key-Range CDSs

ARCIMPRT has been enhanced to recover multicluster non-key-range data sets. It uses the key boundaries to determine which journal records need to be applied to the cluster being recovered.

If you reorganize your CDSs, you should immediately back them up. If you do not, and you attempt to perform a CDS recovery using a CDS backup version created before the CDS was reorganized, the recovery fails with message ARC0466I. Then the following steps must be taken to restore your multicluster non-key-range CDS:

1. The cluster being recovered must be returned to the state in which it existed after the reorganization. For example, if the CDSs were split from two to three, then the job or jobs used to perform the split should be rerun. Only the cluster or clusters being recovered should be returned to the state they were in following the reorganization.
2. Reissue the ARCIMPRT job using the new FORCE parameter. The FORCE parameter causes ARCIMPRT to use the latest key boundaries in use, instead of those that existed at the time of the CDS version backup. To use the FORCE parameter on the MCDS, you would issue:

   ```
   PARM='MCDS,FORCE'
   ```

After the recovery, message ARC0467I is issued identifying the cluster that was recovered.

CDS Recovery Procedure using CDSVERSIONBACKUP with UPDATEC

For performance reasons, you should use the enhanced CDS recovery function instead of the UPDATEC command to recover your control data sets. However, if you want to use UPDATEC, use the following procedure:

1. Use the CDSVERSIONBACKUP parameter of the QUERY command to find out the initial characters and the final qualifier of the backup data set name. Save the QUERY information for use in steps 5 on page 470 and 11 on page 471. You will receive the following messages from the query:

   ARC0375I

   CDSVERSIONBACKUP,
   MCDSBACKUPDSN=*uid*.MCDS.BACKUP,
   BCDSBACKUPDSN=*uid*.BCDS.BACKUP,
   OCDSBACKUPDSN=*uid*.OCDS.BACKUP,
   JRNLBACKUPDSN=*uid*.JRNL.BACKUP

   ARC0376I

   BACKUPCOPIES=3,
   BACKUPDEVICECATEGORY=DASD,
   LATESTFINALQUALIFIER=V0000007,
   DATAMOVER=HSM

   Recommendation: If DFSMShsm is not active, you should issue the following IDCAMS LISTCAT command to determine the latest version: **LISTCAT LEVEL(*uid*)**

Also note whether DFSMShsm or DFSMSdss was the data mover (by looking at the prefix letter of the low-level qualifier).

2. Stop DFSMShsm in all processing units.
3. Use the Access Method Services DELETE command to delete the control data set that has the error. The following example shows the DELETE command using one of several parameters that are available.

   DELETE CLUSTER(*uid*.MCDS)

4. Re-create the control data set that was deleted in step 3. When you re-create the control data set, increase the space allocation for that control data set if more space is required. For more information, refer to the *z/OS DFSMShsm Implementation and Customization Guide*.
5. The next step depends on the data mover used to create the backup version.
   - If the data mover was DFSMSdss, restore the latest backup version into the re-created data set using the following JCL example:

```
//JOB1 JOB accounting information,REGION=nnnK
//STEP1     EXEC  PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE      DD    DSN=uid.MCDS.BACKUP.D0000106,DISP=(OLD,KEEP)
//DASD      DD    UNIT=3390,VOL=SER=222222,DISP=OLD
//SYSIN     DD    *
            RESTORE DATASET(-
            INCLUDE(uid.dsname))-
            INDDNAME(TAPE)-
            OUTDDNAME(DASD)-
            REPLACE
/*
```

   - If the data mover was DFSMShsm, use the Access Method Services IMPORT command to recover the latest backup version of the affected control data set by using the following JCL example:

```
//IMPORT JOB ,'USERID' MSGCLASS=A,REGION=768K
//STEP1     EXEC  PGM=IDCAMS
//DEST      DD    DSN=uid.MCDS,DISP=OLD
//SOURCE    DD    DSN=uid.MCDS,DISP=OLD,AMP=AMORG
//DASD      DD    DSN=uid.MCDS.BACKUP.V0000106,
//                DISP=OLD
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD    *
            IMPORT-
              INFILE(DASD)-
              OUTFILE(SOURCE)-
              INTOEMPTY-
              PURGE
/*
```

6. Restart DFSMShsm in emergency mode, EMERG=YES, in the processing unit from which you will issue the UPDATEC command.

   Recommendation: On the restart, ensure that your ARCCMD*xx* parmlib member is set correctly with either of the following specification:

   - SETSYS EMERG is specified.
   - Neither SETSYS EMERG nor SETSYS NOEMERG is specified.

   If you follow either of the above recommendations, then the EMERG=YES specification will not be overridden by your ARCCMDxx member.

7. Use the UPDATEC command to combine the entries from the journal data set with the entries from the recovered control data set. (See "Updating Recovered Control Data Sets using UPDATEC.") When you use this command, the backup version number of the journal data set is one greater than that of the backup version control data set used in step 5 on page 470. You must concatenate the journal backup data set name that you received in the ARC0375I message in step 1 on page 469 with the version number, and use that as the journal parameter of the UPDATEC command.

   If you need to use multiple backup versions of the journal data set to recover a control data set, be sure you apply the backup versions in the correct order (oldest to newest).
8. Stop and restart DFSMShsm in emergency mode, EMERG=YES, from the processing unit from which you issued the UPDATEC command.

   Recommendation: On the restart, ensure that your ARCCMD*xx* parmlib member is set correctly with either of the following specification:
   - SETSYS EMERG is specified.
   - Neither SETSYS EMERG nor SETSYS NOEMERG is specified.

   If you follow either of the above recommendations, then the EMERG=YES specification will not be overridden by your ARCCMDxx member.
9. Verify that the control data sets are correct. (See "Determining If Control Data Sets Are Accurate" on page 472.)
10. Fix any errors in the control data sets. (See "Repairing Errors in the Control Data Sets" on page 473.)
11. If you are recovering the MCDS, issue a QUERY CDSVERSIONBACKUP to ensure that the latest final qualifier is the same as that shown in step 1. If you do not have the output from the QUERY command in step 1, you can determine the final qualifier of the latest backup copy from your catalog using the LISTCAT LEVEL command, as shown in the following example:

    LISTCAT LEVEL(*uid*)

    If these qualifiers do not match, use the FIXCDS patch command to correct the field at offset X'B1' in the MHCR record to reflect the latest version in the catalog, as shown in the following example:

    FIXCDS S MHCR PATCH(X'B1' 0000007)
12. Use the CONTROLDATASETS parameter of the BACKVOL command to get a current backup copy of the control data sets and journal data set.
13. If the backup of the CDSs was successful, you can issue a SETSYS NOEMERG command, and release DFSMShsm's automatic functions with a RELEASE ALL command.
14. Restart DFSMShsm in the other processing units.

Updating Recovered Control Data Sets using UPDATEC

For performance reasons, you should use the enhanced CDS recovery function instead of the UPDATEC command to recover your control data sets. If you do use UPDATEC, the forms of the command are:

```
UPDATEC  BACKUPCONTROLDATASET

UPDATEC  BACKUPCONTROLDATASET  JOURNAL(dsname)

UPDATEC  MIGRATIONCONTROLDATASET
```

UPDATEC MIGRATIONCONTROLDATASET JOURNAL(*dsname*)

UPDATEC OFFLINECONTROLDATASET

UPDATEC OFFLINECONTROLDATASET JOURNAL(*dsname*)

UPDATEC ALL

UPDATEC ALL JOURNAL(*dsname*)

The forms with no journal specified use the current (active) journal data set to update the specified data set. The forms with a journal specified use a backed up journal data set requested by *dsname* to update the specified data set. For example, the command UPDATEC BACKUPCONTROLDATASET applies the transactions recorded in the active journal data set to the backup control data set. Thus, if the backup control data set were a copy of the last backup version of the backup control data set, the UPDATEC command would modify it to the state of the backup control data set before it was lost or damaged.

If the most-recent backup version of a control data set is also damaged, you can still recover the control data set by a three-step process:

1. Import the next-most-recent backup version of the data set.
2. Run UPDATEC with the journal that corresponds to the next-most-recent backup version. This journal is the most-recent *backed up* journal.
3. Run UPDATEC with the current journal.

Note: The UPDATEC command may be used to recover a damaged control data set. However, UPDATEC can take a long time to run. Because the process holds an exclusive enqueue and reserve on all the control data sets, no other DFSMShsm functions (such as RECALL) can be run when UPDATEC is running. Therefore, a complex will not be able to do any DFSMShsm work until UPDATEC finishes. In a JES3 environment, the whole complex may be unavailable until UPDATEC finishes processing.

VSAM Considerations for the Control Data Sets

The three control data sets are VSAM data sets and as such need to be periodically compressed and reorganized. To redistribute records or compress the control data sets, use the Access Method Services EXPORT and IMPORT commands. To move the control data sets to another volume or to increase their space allocations, use the Access Method Services REPRO, DELETE, and DEFINE commands.

Immediately after you reorganize your CDSs you should back them up.

Related reading: For additional information about VSAM control data sets, see "VSAM Considerations for Control Data Sets" in the *z/OS DFSMShsm Implementation and Customization Guide*.

Determining If Control Data Sets Are Accurate

During recovery processing, a control data set can lose synchronization with the actual conditions in the system. DFSMShsm provides a command allowing you to determine how closely the control data sets match the actual system conditions and to repair minor discrepancies.

The AUDIT command detects and reports discrepancies about data set information from:

- Computing system catalogs
- DFSMShsm control data set records
- The VTOC of the DASD storage volume on which the data set resides

An example of the AUDIT command is:

```
AUDIT  DATASETCONTROLS(MIGRATION) -
  NOFIX  OUTDATASET(outputname)  REPORT(ERRORS)
```

In processing this command, DFSMShsm reads each data set record in the MCDS. DFSMShsm compares the information about each data set to the information contained in the system catalogs and in the VTOCs of any DASD volumes that the catalogs or the MCDS indicates the data set resides on. DFSMShsm does not try (because of the NOFIX parameter) to reconcile any differences that it finds. Instead, DFSMShsm sends a report of any errors to the data set specified by the OUTDATASET parameter.

Recommendation: The OUTDATASET contains executable FIXCDS commands to create or correct CDS records, but those CDS changes are only a subset of the corrective actions taken by AUDIT. In addition, the CDS records may be read multiple times during the course of an audit. If the NOFIX option is specified, the changes to a record generated on its initial read are not reflected in that record when it is read a subsequent time, which may result in incorrect actions being taken. For these reasons, the later submission of output from an AUDIT NOFIX as a substitute for an AUDIT FIX is generally discouraged.

The AUDIT command helps determine how closely the control data sets match the actual system conditions. There are Access Method Services commands that verify the structure of the CDS. The Access Method Services DIAGNOSE and EXAMINE commands can be used to validate the VSAM volume data set data structures and analyze the structural consistency of the index and data component of a key-sequenced data set.

Related reading: For additional information about the AUDIT command, see the *z/OS DFSMShsm Storage Administration Reference*.

Repairing Errors in the Control Data Sets

For various reasons, the records in the control data set may occasionally need to be changed. You use the FIXCDS command to display or change control data set records. With the FIXCDS command, you can specify a specific record type, a specific data set, and a specific kind of change to make.

Related reading: For additional information about the FIXCDS command, see *z/OS DFSMShsm Diagnosis Guide and Reference*.

Recovering from a Journal I/O Error

A journal I/O error may have two causes. The first cause is a corrupted journal record header (MLOG). To recover from this type of error, do the following:

Action	Command
1. Put DFSMShsm in emergency mode.	SETSYS EMERGENCY

Action	Command
2. Issue the CONTROLDATASETS parameter of the BACKVOL command to back up the control and journal data sets. **Note:** This step may fail with another I/O error. If it does, see "Example 2: Replacing a Damaged Journal" on page 478 for additional information on how to replace a damaged journal.	BACKVOL CONTROLDATASETS
3. Stop DFSMShsm.	F DFSMSHSM,STOP
4. Use IEBGENER to copy some data into the journal data set to overwrite the existing control record.	
5. Start DFSMShsm.	S DFSMSHSM

The second cause of a journal I/O error is a hardware error. If overwriting the journal record header does not correct the journal I/O error, a hardware malfunction probably caused the error.

To correct a hardware-caused journal I/O error, you may have to either 1) allocate a new journal data set on the same volume in a different location or 2) allocate a new journal data set on a different volume.

To allocate a new journal data set on the same DASD volume, do the following:

Action	Command
1. Put DFSMShsm in emergency mode.	SETSYS EMERGENCY
2. Issue the CONTROLDATASETS parameter of the BACKVOL command to back up the control and journal data sets. **Note:** This step may fail with another I/O error. If it does, see "Example 2: Replacing a Damaged Journal" on page 478 for additional information on how to replace a damaged journal.	BACKVOL CONTROLDATASETS
3. Stop DFSMShsm.	F DFSMSHSM,STOP
4. Rename the old journal data set	
5. Allocate a new journal data set	
6. Delete the old journal data set	
7. Start DFSMShsm.	S DFSMSHSM

To allocate a new journal data set on a different DASD volume, follow the instructions in the next topic "Moving the Journal".

Moving the Journal

To move the journal data set from one DASD device to another, do the following:

Action	Command
1. Put DFSMShsm in emergency mode.	SETSYS EMERGENCY

Action	Command
2. Issue the CONTROLDATASETS parameter of the BACKVOL command to back up the control and journal data sets.	BACKVOL CONTROLDATASETS
3. Stop DFSMShsm.	F DFSMSHSM,STOP
4. Delete the old journal data set	
5. Allocate a new journal data set on a different DASD device.	
6. Start DFSMShsm.	S DFSMSHSM

Steps to Ensure That the Control Data Sets Are Accurate

After you recover the control data sets, you can do the following to ensure that they are accurate:

1. Issue the AUDIT commands to determine if any discrepancies exist between the data set records in the recovered control data sets and the data set records in the computing system catalog.
2. Issue the AUDIT commands with the FIX parameter to fix any errors that AUDIT can correct.
3. Issue the FIXCDS command to fix any minor problems encountered in the recovered control data sets.

CDS Backup and Recovery Examples

This section provides some examples of recovering your control data sets and journal data set.

Recovery Examples using the Enhanced CDS Recovery Function

The following examples help you recover your control data sets or journal data sets using the enhanced CDS recovery function.

Example 1: Recovering the MCDS from DASD

This example shows how you set up your environment, how to recover the MCDS, and the messages that are issued during the backup and recovery processing.

Set up your environment with the following SETSYS command:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(3) -
  BACKUPDEVICECATEGORY(DASD))
```

Issue the QUERY CDSVERSIONBACKUP command to find out what the last final qualifier was for the backup data sets. DFSMShsm issues the following messages:

```
ARC0375I    CDSVERSIONBACKUP,
            MCDSBACKUPDSN=uid.MCDS.BACKUP,
            BCDSBACKUPDSN=uid.BCDS.BACKUP,
            OCDSBACKUPDSN=uid.OCDS.BACKUP,
            JRNLBACKUPDSN=uid.JRNL.BACKUP

ARC0376I    BACKUPCOPIES=3,
            BACKUPDEVICECATEGORY=DASD,
            LATESTFINALQUALIFIER=V0000000,
```

DATAMOVER=HSM

Note: Since the LATESTFINALQUALIFIER begins with a V, the last data mover was DFSMShsm.

Because no backup data sets are preallocated, you preallocate the following backup data sets:

uid.MCDS.BACKUP.V0000001 through *uid*.MCDS.BACKUP.V0000003
uid.BCDS.BACKUP.V0000001 through *uid*.BCDS.BACKUP.V0000003
uid.OCDS.BACKUP.V0000001 through *uid*.OCDS.BACKUP.V0000003
uid.JRNL.BACKUP.V0000001 through *uid*.JRNL.BACKUP.V0000003

Now you let DFSMShsm execute to a point where DFSMShsm has successfully backed up the control data sets and the journal data set ten times. The backup data sets have the final qualifiers of V0000008, V0000009, and V0000010.

As DFSMShsm processes the MCDS, an I/O error occurs, so a part of the MCDS is no longer usable. To recover the MCDS, you would:

1. Issue QUERY CDSVERSIONBACKUP to find out what was the latest backup version. DFSMShsm issues the following messages:

   ARC0375I CDSVERSIONBACKUP,
   MCDSBACKUPDSN=*uid*.MCDS.BACKUP,
   BCDSBACKUPDSN=*uid*.BCDS.BACKUP,
   OCDSBACKUPDSN=*uid*.OCDS.BACKUP,
   JRNLBACKUPDSN=*uid*.JRNL.BACKUP

   ARC0376I BACKUPCOPIES=3,
   BACKUPDEVICECATEGORY=DASD,
   LATESTFINALQUALIFIER=V0000010,
   DATAMOVER=HSM

   Recommendation: If the QUERY CDSVERSIONBACKUP command is not successful or if DFSMShsm is not active, issue the following IDCAMS LISTCAT command to determine the latest version: **LISTCAT LEVEL(*uid*)**
2. Issue a IDCAMS LISTCAT(*uid*) to determine if the backup copy is good. This determination is made by checking the low-level qualifier for a V or an X.
3. Stop DFSMShsm in all processing units.
4. See "Requirements for Using the Enhanced CDS Recovery Function" on page 463 for information on using the enhanced CDS recovery function to recover the MCDS.
5. Restart DFSMShsm in the processing unit.
6. Verify that the control data sets are correct (see "Determining If Control Data Sets Are Accurate" on page 472).
7. Fix any errors in the control data sets (see "Repairing Errors in the Control Data Sets" on page 473).
8. Because you are recovering the MCDS, issue a QUERY CDSVERSIONBACKUP command to ensure that the latest final qualifier is the same as that shown in step 1. If you do not have the output from query command in step 1, you can determine the final qualifier of the latest backup copy from your catalog using the LISTCAT LEVEL command, as shown in the following example:

   LISTCAT LEVEL(*uid*)

If these qualifiers do not match, use the FIXCDS patch command to correct the field at offset X'B1' in the MHCR record to reflect the latest version in the catalog, as shown in the following example:

```
FIXCDS S MHCR PATCH(X'B1' 0000010)
```

9. Issue the BACKVOL CDS command to ensure that all copies of the control data sets and the journal data set are current. During backup of the control data sets and the journal data set, DFSMShsm issues the following messages:

   ARC0740I CDS BACKUP STARTING AT 04:15:35 ON 95/05/24 ON SYSTEM 381A TO DASD IN PARALLEL MODE, DATAMOVER=HSM

   ARC0742I BACKUP FOR MCDS STARTING AT 04:15:55 ON 95/05/24, BACKUP COPY TECHNIQUE IS UNDEFINED

   ARC0743I MCDS SUCCESSFULLY BACKED UP TO *uid*.MCDS.BACKUP.V0000008, ON VOLUMES VOL001,VOL002, TIME=04:20:03, DATE=95/05/24

   ARC0742I BACKUP FOR BCDS STARTING AT 04:25:55 ON 95/05/24, BACKUP COPY TECHNIQUE IS UNDEFINED

   ARC0743I BCDS SUCCESSFULLY BACKED UP TO *uid*.BCDS.BACKUP.V0000008, ON VOLUMES VOL003,VOL004, TIME=04:30:13, DATE=95/05/24

   ARC0742I BACKUP FOR OCDS STARTING AT 04:35:55 ON 95/05/24, BACKUP COPY TECHNIQUE IS UNDEFINED

   ARC0743I OCDS SUCCESSFULLY BACKED UP TO *uid*.OCDS.BACKUP.V0000008, ON VOLUME VOL005, TIME=04:40:44, DATE=95/05/24

   ARC0750I BACKUP FOR JRNL STARTING AT 04:40:55 ON 95/05/24

   ARC0743I JRNL SUCCESSFULLY BACKED UP TO *uid*.JRNL.BACKUP.V0000008, ON VOLUME VOL006, TIME=04:45:26, DATE=95/05/24

   ARC0748I LAST SUCCESSFUL CDS BACKUP-SET QUALIFIER IS V0000011

   ARC0741I CDS BACKUP ENDING AT 04:45:35 ON 95/05/24, STATUS=SUCCESSFUL

   Note: Because you have specified three backup copies and you are backing up to DASD, DFSMShsm replaces the V0000008 version of the backup data sets with V0000011.

   If you issue the QUERY CDSVERSIONBACKUP command again, DFSMShsm issues the following messages:

   ARC0375I CDSVERSIONBACKUP, MCDSBACKUPDSN=*uid*.MCDS.BACKUP, BCDSBACKUPDSN=*uid*.BCDS.BACKUP, OCDSBACKUPDSN=*uid*.OCDS.BACKUP, JRNLBACKUPDSN=*uid*.JRNL.BACKUP

ARC0376I BACKUPCOPIES=3,
BACKUPDEVICECATEGORY=DASD,
LATESTFINALQUALIFIER=V0000011,
DATAMOVER=HSM

10. Restart DFSMShsm in the other processing units.

Example 2: Replacing a Damaged Journal

This example shows how you set up your environment, how to recover the journal, and some of the messages that are issued during the backup and recovery processing.

Set up your environment with the following SETSYS command:

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

You preallocate the following backup data sets:

```
BHSM.MCDS.BACKUP.D0000001
BHSM.BCDS.BACKUP.D0000001
BHSM.OCDS.BACKUP.D0000001
BHSM.JRNL.BACKUP.D0000001

BHSM.MCDS.BACKUP.D0000002
BHSM.BCDS.BACKUP.D0000002
BHSM.OCDS.BACKUP.D0000002
BHSM.JRNL.BACKUP.D0000002

BHSM.MCDS.BACKUP.D0000003
BHSM.BCDS.BACKUP.D0000003
BHSM.OCDS.BACKUP.D0000003
BHSM.JRNL.BACKUP.D0000003

BHSM.MCDS.BACKUP.D0000004
BHSM.BCDS.BACKUP.D0000004
BHSM.OCDS.BACKUP.D0000004
BHSM.JRNL.BACKUP.D0000004
```

Assume DFSMShsm has successfully backed up the control data sets and the journal data set ten times. The backup data sets have the final qualifiers of D0000007, D0000008, D0000009, and D0000010.

As DFSMShsm processes the journal data set, an I/O error occurs. DFSMShsm disables the journal data set and issues the following message:

ARC0026A JOURNALING DISABLED DUE TO I/O ERROR.
MIGRATION, BACKUP, DUMP, AND RECYCLE HELD.

Issue a BACKVOL CDS command to back up the control data sets and the journal data set. DFSMShsm successfully backs up the control data sets and issues message

ARC0743I for each backup. The message indicates that DFSMShsm made the backup to the backup data set whose final qualifier is D0000007 (oldest version).

When DFSMShsm tries to back up the journal, another I/O error occurs. Because DFSMShsm cannot successfully complete the journal data set backup function, it renames the journal backup data set to X0000011. Then DFSMShsm issues the following message:

ARC0744E JRNL COULD NOT BE BACKED UP,
RC=*retcode*, REASON=*reascode*
MIGRATION, BACKUP, DUMP, AND RECYCLE HELD.

In addition, DFSMShsm issues the following message to indicate the latest final backup data set qualifier (D0000011). DFSMShsm waits for the completion of the CDS backups and then nulls (empties) the journal data set if no CDS backup failures occur.

ARC0748I LAST SUCCESSFUL CDS BACKUP-SET QUALIFIER IS
D0000011

At this point, assume the following conditions exist:

- The latest backup versions of the control data sets are backup data sets whose final qualifier is D0000011.
- The latest good backup version for the journal data set is a backup data set whose final qualifier is D0000010.
- The current control data sets are usable.

If you want to continue processing with the journal, you must stop DFSMShsm, delete and reallocate the journal data set to avoid getting another I/O error during processing of the data set. This process is described in "Reasons For Recovering the Control Data Sets" on page 462. A hardware problem could be causing the I/O error on the journal data set, so be sure you do not re-allocate the journal data set to where it was allocated before the I/O error occurred.

After you restart DFSMShsm, issue BACKVOL CDS to back up the control data sets and the new, empty journal data set. DFSMShsm backs up the control data sets and the journal data set to the backup data sets whose final qualifiers are D0000008. If DFSMShsm successfully backs up the control data sets and the journal data set, DFSMShsm renames the backup data sets with the final qualifier of D0000012 (or V0000012 if IDCAMS is the data mover).

Example 3: Recovering the MCDS from Tape When the Latest MCDS Backup Version Is Unusable

This example shows how to set up your environment, how to recover the MCDS when the most recent backup version is unusable, and the processing that occurs when recovering the MCDS.

Set up your environment with the following SETSYS command:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
  BACKUPCOPIES(6) -
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
```

JRNLBACKUPDSN(BHSM.JRNL.BACKUP) -
BACKUPDEVICECATEGORY(TAPE))

Assume DFSMShsm has successfully backed up the control data sets and the journal data set six times, and the latest final qualifier is D0000006. Therefore, backup data sets with final qualifiers of D0000001 through D0000006 exist on tape volumes.

If an error occurs so DFSMShsm cannot access the MCDS, you:

1. Issue QUERY CDSVERSIONBACKUP and observe that the latest final qualifier is D0000006.
2. Using DFSMSdss, restore the MCDS backup version of D0000006. Assume that, when you attempt this restore, the restore fails. You must now use MCDS backup version BHSM.MCDS.BACKUP.D0000005.
3. Using DFSMSdss, restore the MCDS backup version of D0000005.
4. Using IDCAMS, export the D0000005 backup version of the MCDS that was restored in step 3. The following example shows JCL that can be used to do the export:

```
//EXPORT JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTDD1   DD DISP=(,CATLG,DELETE),BUFNO=26,BLKSIZE=28332,
//            SPACE=(CYL,(200,20),RLSE),
//            UNIT=3390,VOL=SER=333333,
//            DSN=BHSM.MCDS.BACKUP.TEMP
//SYSIN    DD *
   EXPORT BHSM.MCDS.BACKUP.D0000005 -
   OUTFILE(OUTDD1) -
   TEMPORARY
/*
```

   Where:

   `BHSM.MCDS.BACKUP.TEMP` is the data set name of the exported copy being created.

   `BHSM.MCDS.BACKUP.D0000005` is the data set name of the control data set being exported.

   `BLKSIZE=28332` is the capacity of half a track on a 3390 device.
5. Stop DFSMShsm in all processing units.
6. Create the backup journal file using the online journal and the latest backup copy of the journal. Use `ARCBJRNL` and `PARM='BOTH'`, `JRNLO DD` for the online journal, and `BKUPJRNL DD` for the journal backup copy version D0000006 to create the backup journal file. See "Implementing Enhanced CDS Recovery using JCL" on page 464 for an example of the JCL needed to create the backup journal file.
7. Sort this newly created backup journal file. See "Implementing Enhanced CDS Recovery using JCL" on page 464 for an example of the JCL needed to sort the backup journal file.
8. Import and merge the exported MCDS from step 4 and the sorted backup journal file from step 7 using ARCIMPRT. Again, see "Implementing Enhanced CDS Recovery using JCL" on page 464 for an example of the JCL needed to do the import and merge.
9. Use the PATCH parameter of the FIXCDS command to update the MHCR to version 6, the latest control data set backup version.

10. Issue the BACKVOL CDS command to obtain a new set of backup copies of the control data sets and journal data set. When DFSMShsm successfully backs up these data sets, the latest backup copies are those data sets whose final qualifier is D0000007.

DFSMShsm uncatalogs the D0000001 backup version. Only versions D0000002 through D0000007 remain cataloged.

Chapter 13. Managing DFSMShsm Media

As time passes, conditions in your computer system change such that you will need to manage the DFSMShsm-owned storage media in your system. For example:

- When the data on backup or ML2 tapes expires, you then want to consolidate the valid data scattered over many tapes.
- You may expect many data sets to be migrated and want to make room on the migration volumes for the new data sets before it is done automatically.
- You may be replacing one DASD device type with another DASD or tape device type and want to move the data sets that are on level 0, level 1, and level 2 volumes to the new device types.
- You may want to remove volumes from DFSMShsm management or ownership.

Moving All Data Sets from One Level 0 Volume to Other Level 0 Volumes

Device conversion of level 0 volumes is performed expeditiously by DFSMSdss. However, the following capabilities exist and are useful under some conditions.

You can use DFSMShsm to move single data sets or all data sets from one level 0 volume to other level 0 volumes. The processing for non-SMS-managed data sets differs from that for SMS-managed data sets.

To move BDAM data sets between volumes of different track sizes, you might consider migrating and then recalling that data set using the DAOPTION.

The commands for moving SMS-managed data sets are:

MIGRATE DATASETNAME(*dsname*) CONVERT

MIGRATE DATASETNAME(*dsname*) CONVERT(*volser2*)

MIGRATE VOLUME(*volser1* MIGRATE(0)) CONVERT

MIGRATE VOLUME(*volser1* MIGRATE(0)) CONVERT(*volser2*)

The commands for moving all non-SMS-managed data sets from one level 0 volume to one or more other level 0 volumes are:

MIGRATE DATASETNAME(*dsname*) CONVERT

MIGRATE DATASETNAME(*dsname*) -
CONVERT(*volser2 unittype2*)

MIGRATE VOLUME(*volser1* MIGRATE(0)) CONVERT

MIGRATE VOLUME(*volser1* MIGRATE(0)) -
CONVERT(*volser2 unittype2*)

MIGRATE VOLUME(*volser1* MIGRATE(0)) CONVERT -

© Copyright IBM Corp. 1984, 2004

UNIT(*unittype1*)

MIGRATE VOLUME(*volser1* MIGRATE(0)) -
CONVERT(*volser2* *unittype2*) UNIT(*unittype1*)

Processing SMS-Managed Data Sets

The main difference in the processing of SMS-managed data sets is that you cannot direct the movement to a specific volume. Although you can specify a volume, SMS-processing determines the actual volume that the data set resides on after being moved. The process for moving data sets migrates the data sets to ML1 volumes and recalls them to level 0 volumes. At the beginning of processing for MIGRATE VOLUME commands, DFSMShsm obtains a list of the management classes active in the system. See "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138 for a discussion of the exception. DFSMShsm does not obtain a list of management classes for MIGRATE DATASETNAME commands. As it processes each data set, DFSMShsm checks the value specified for the data set by the COMMAND OR AUTOMIGRATE 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 and passes the information described in "Automatic Recall" on page 65.

If SMS is installed and active, the ACS routine may return to DFSMShsm a storage class and, optionally, a management class. If the ACS routine returns a storage class, DFSMShsm passes the data set name to DFSMSdfp, along with its storage class name and management class name (if present). DFSMSdfp selects a volume and places the data set on it. The volume selected is not necessarily the target volume specified by the MIGRATE command. The catalog entry is updated to reflect the current status.

To prevent SMS from selecting the source volume as the target volume for the recall, change the status attribute for the volume in the storage group. Suitable status attributes are DISABLENEW and QUIESCENEW.

If all the data sets on a volume are being processed (MIGRATE VOLUME is specified), any data set that should be expired is expired without being migrated.

Processing Non-SMS-Managed Data Sets

There are two differences in processing for non-SMS-managed data sets:

- No storage class, management class, and data class are received.
- Data sets are recalled to the volume specified.

The target unit type need not be specified if the volume has been added to DFSMShsm by an ADDVOL command.

For non-SMS-managed data sets, DFSMShsm migrates the data sets to an ML1 volume and then recalls them to the volume specified by *volser2*. If *volser2* is not specified, DFSMShsm selects the level 0 volumes to recall the data sets to in the manner specified in "Recall" on page 101. If *volser2* has not been identified to DFSMShsm, use *unittype2* to specify the type of unit that *volser2* can be mounted on. If *volser1* has not been identified to DFSMShsm, use the UNIT(*unittype1*)

parameter to specify the type of unit that *volser1* can be mounted on. To prevent selection of the source volume as the target volume for the recall, you must do two things:

- Specify the volume to DFSMShsm with NOAUTORECALL in the ADDVOL command
- Remove the volume from any pools of which it is a member

With BDAM data sets, when you specify a target volume on a different device type during DASD conversion, DFSMShsm recalls the data set you specify if the data on a track of the source volume fits on a track of the target volume. However, you are responsible for data addressability.

DFSMShsm maintains the same number of tracks for BDAM data set space during DASD conversion to a different device type. There is no change in the number of records per track, so space is wasted when you convert a BDAM data set to a DASD with a larger track size. DFSMShsm assumes that any user program accessing BDAM data depends on having the same relative track address.

DFSMShsm does not compact data sets or migrate them to SDSP data sets when it does DASD conversion.

If you specify the CONVERT parameter in a direct-to-tape environment, the command fails.

Examples of Moving Data Sets between Level 0 Volumes

Consider what happens if you use the MIGRATE CONVERT command in the system shown in Figure 69 on page 486.

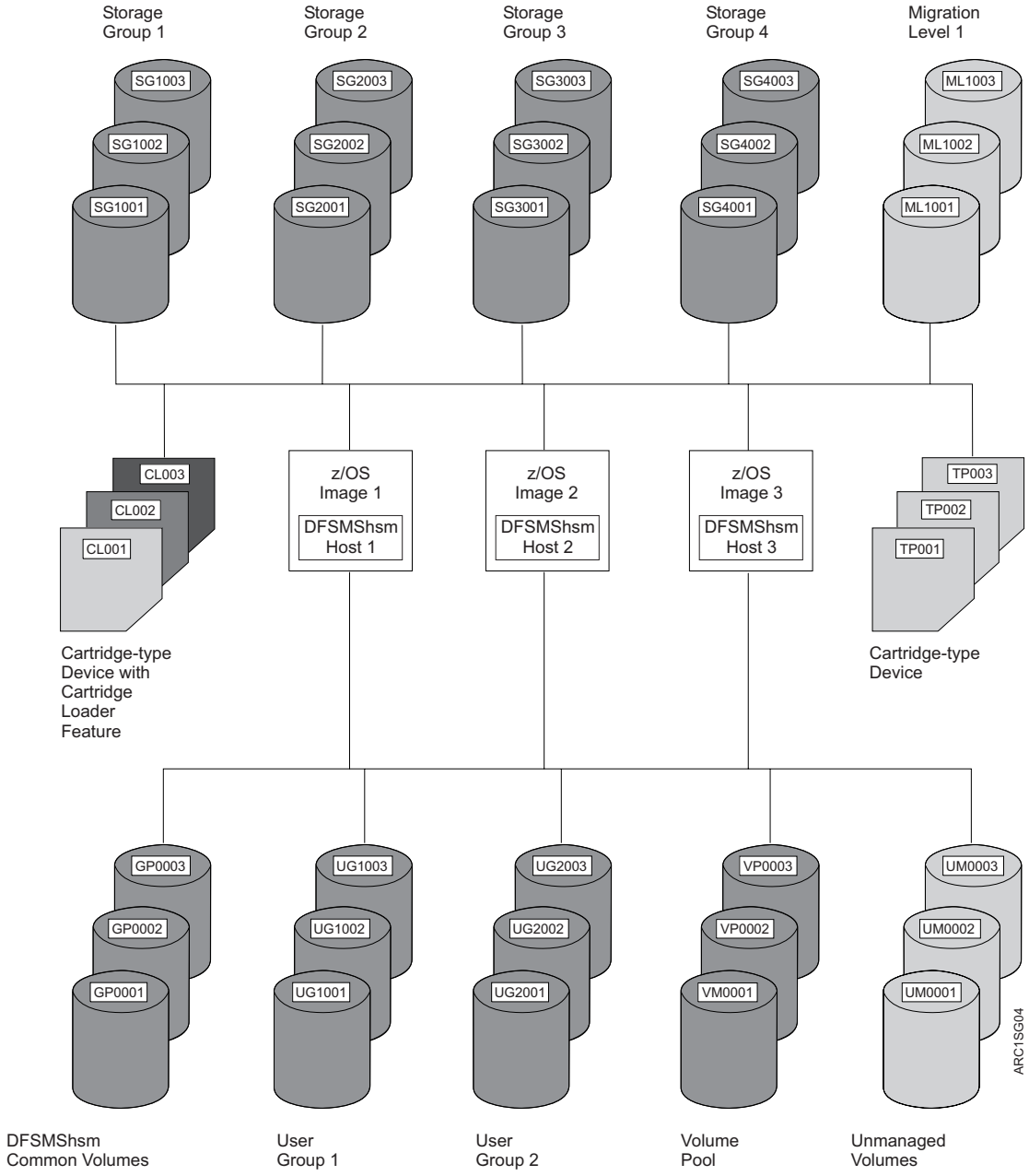


Figure 69. Example System for MIGRATE CONVERT

SMS-Managed Volume

Assume that you want to remove volume SG2002 from the system and replace it with a new volume. You must do three things:

- Prevent new allocations to volume SG2002
- Remove the data sets that DFSMShsm can process from the volume
- Remove the data sets that DFSMShsm cannot process from the volume

To prevent new allocations to the volume, you must change the status of the volume in the storage group. For example, you can establish the volume status in the storage group definition as shown in Figure 70.

```
STORAGE GROUP NAME: STGGP2
VOLUME SERIAL NUMBER: SG2002

   SYSTEM        SMS VOL
   NAME          STATUS
  --------       --------
  PROCUN1  ===>  DISNEW
  PROCUN2  ===>  DISNEW
  PROCUN3  ===>  DISNEW
```

Figure 70. SMS Volume Status

To remove the data sets that DFSMShsm can process, you can issue the following command:

MIGRATE VOLUME(SG2002 MIGRATE(0)) CONVERT

DFSMShsm migrates all of the data sets that it can process off of volume SG2002 and recalls them to other SMS-managed volumes. The DFSMSdfp automatic class selection routine selects the volume for data set recall.

To move any data sets that DFSMShsm cannot process, you can use DFSMSdss.

Non-SMS-Managed Volume

Assume that you want to remove all data sets from volume UG2003. You still must prevent allocations to the volume, remove data sets that DFSMShsm can process from the volume, and remove any data sets that DFSMShsm cannot process from the volume.

Preventing users other than DFSMShsm from allocating to the volume is a normal MVS process. You can prevent DFSMShsm from selecting the volume for recall of data sets by removing UG2003 from the user group with the following command in all z/OS images:

DEFINE POOL(GROUP2 VOLUMES(UG2001 UG2002))

Because you have volume UG2003 dedicated to GROUP2, you have specified the NOAUTORECALL parameter of the ADDVOL command. Thus, because the volume does not belong to any pool and has the attribute of NOAUTORECALL, DFSMShsm cannot recall any data sets to it.

To move the data sets that DFSMShsm can process, issue the following command:

MIGRATE VOLUME(UG2003 MIGRATE(0)) CONVERT

Notice that this is the same command used for the SMS-managed volume. DFSMShsm again migrates and recalls all of the data sets on the volume that it can process. However, because the volume is not SMS managed (because the ACS routines indicate the data sets should be non-SMS-managed), DFSMShsm selects the volumes to which the data sets are recalled. Because, presumably, all of the data sets on volume UG2003 belong to GROUP2, DFSMShsm recalls them to either UG2001 or UG2002.

To move any data sets that DFSMShsm cannot process, you can use DFSMSdss.

Moving Some Data Sets from Migration Level 1 Volumes

Two commands, FREEVOL and MIGRATE, are available to move data from ML1 volumes. You can move data sets from individual ML1 volumes with the FREEVOL command, or you can move or delete data sets from all ML1 volumes with the MIGRATE command. If you intend to make more space available on ML1 volumes, use the MIGRATE ML1 command.

Note: The "data" and "data sets" mentioned here and in the next few sections refer to data sets that have been created by DFSMShsm. If your installation stores non-SMS-managed user or system data on ML1 volumes (refer to *z/OS DFSMShsm Implementation and Customization Guide* for further restrictions), this data is not affected by FREEVOL and MIGRATE commands.

Use the FREEVOL AGE(0) command to empty DASD migration or DASD backup volumes in preparation for movement to new equipment or to replace partially damaged volumes. If you are attempting to empty an ML1 volume, there can be two types of data that need to be moved off the volume: backup versions and migrated data that must remain until it is backed up. Since backup of the migrated data only occurs when automatic backup is run on the primary DFSMShsm host, ensure that all data sets are removed from the ML1 volume by running automatic backup on your primary DFSMShsm host shortly before issuing the FREEVOL AGE(0) command.

Using the FREEVOL Command for Migration Level 1 Volumes

The FREEVOL command allows you to move data sets from a single ML1 volume to other ML1 volumes or to ML2 volumes. This command can be useful when converting to or from the use of SDSPs when the target is level 1. Selection of the target volumes can be manipulated by using the DRAIN parameter of the ADDVOL command. You can use the following commands to move data sets that exceed a certain age:

```
FREEVOL MIGRATIONVOLUME(volser) AGE(days)

FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL1)

FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL2)

FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD))

FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE))
```

If you use the data set backup functions BACKDS, HBACKDS, ARCINBAK, or ARCHBACK and run short of ML1 space, you can move BACKDS versions from all ML1 volumes to daily backup volumes by using the FREEVOL ML1BACKUPVERSIONS command.

In all forms of the command used in this section, the value for the AGE parameter must be greater than zero. Processing of the commands is different for SMS-managed and non-SMS-managed data sets.

Processing for SMS-Managed Data Sets

At the beginning of processing, DFSMShsm obtains a list of the active management classes in the system. See "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138 for a discussion of the exception. DFSMShsm then calculates the inactive age of each data set and compares it to the value of the LEVEL 1 DAYS NON-USAGE attribute in the management class with which the data set is associated. If the inactive age is greater than or equal to the value of the LEVEL 1 DAYS NON-USAGE attribute, the data set migrates to an ML2 volume, unless the value of the attribute is NOLIMIT. If the attribute value is NOLIMIT, the data set does not migrate.

The TARGETLEVEL parameter does not determine the level to which SMS-managed data sets migrate. However, if the MIGRATIONLEVEL2 subparameter is specified with a device category of TAPE or DASD, the data set migrates to an ML2 volume of the category specified. If the MIGRATIONLEVEL2 subparameter does not specify a device category, the data set migrates to a level 2 volume of the category specified by the TAPEMIGRATION parameter of the SETSYS command.

Processing for Non-SMS-Managed Data Sets

For non-SMS-managed data sets, DFSMShsm compares the time since the data set was last opened against the value specified with the AGE parameter. If the time since the data set was last opened is equal to or more than the value specified with the AGE parameter, DFSMShsm migrates the data set to a volume of the level specified by the TARGETLEVEL parameter. If the MIGRATIONLEVEL2 subparameter of the TARGETLEVEL parameter specifies a device category, the data set migrates to an ML2 volume of the category specified. If the MIGRATIONLEVEL2 subparameter does not specify a device category, the data set migrates to an ML2 volume of the category specified by the TAPEMIGRATION parameter of the SETSYS command.

Using the MIGRATE Command for Migration Level 1 Volumes

With the MIGRATE command you can either migrate data sets from all the ML1 volumes or delete migrated data sets from individual ML1 volumes. The commands to delete the migrated data sets do not affect SMS-managed data sets.

Migrating Data Sets

The commands to migrate data sets are:

MIGRATE MIGRATIONLEVEL1

MIGRATE MIGRATIONLEVEL1 DAYS(*days*)

For each form of the command, SMS-managed data sets and non-SMS-managed data sets are processed differently.

Processing for SMS-Managed Data Sets: For SMS-managed data sets, DFSMShsm obtains a list of the management classes that are active in the system, unless a volume function is processing or has just completed. See "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138 for a discussion of the exception. It then compares the time since the data set was last opened to the value of the LEVEL 1 DAYS NON-USAGE attribute in the appropriate management class for each data set. If the time since the data set was last opened is equal to or more than the value in the LEVEL 1 DAYS NON-USAGE parameter, the data set migrates to ML2. Otherwise, the data set remains on the

ML1 volume. If the value of the LEVEL 1 DAYS NON-USAGE attribute is NOLIMIT, the data set remains on the ML1 volume.

Processing for Non-SMS-Managed Data Sets: For non-SMS-managed data sets, the processing depends on whether the DAYS parameter is specified on the MIGRATE command.

If the DAYS parameter is *not* specified on the MIGRATE command, DFSMShsm checks the time since the data set was last opened against the value of the MIGRATIONLEVEL1DAYS parameter of the SETSYS command. If the data set has not been opened for a number of days more than or equal to the value specified by MIGRATIONLEVEL1DAYS, the data set migrates to ML2.

If the DAYS parameter is specified on the MIGRATE command, DFSMShsm checks the time since the data set was last opened against the value specified with the DAYS parameter. If the data set has not been opened for a number of days more than or equal to the value specified by the DAYS parameter, the data set migrates to ML2.

Deleting Non-SMS-Managed Data Sets from Migration Volumes

You can use the MIGRATE command to delete obsolete non-SMS-managed data sets from migration volumes. The forms of the command to delete data sets are:

MIGRATE VOLUME(*volser* DELETEBYAGE(*days*))

MIGRATE VOLUME(*volser* DELETEIFBACKEDUP(*days*))

If you specify DELETEBYAGE, DFSMShsm deletes any non-SMS-managed data set that has not been opened for more days than is specified by DELETEBYAGE(*days*) from the migration volume. It does not delete any data set that has not passed the expiration date indicated in the data set's VTOC entry.

If you specify DELETEIFBACKEDUP, DFSMShsm deletes any non-SMS-managed data set that has a valid backup version and that has not been opened for more days than is specified by DELETEIFBACKEDUP(*days*). DFSMShsm also marks the latest valid backup copy as a retired version. This command deletes the data set regardless of whether or not the data set has passed the expiration date indicated in the data set's VTOC entry.

Moving All the Data Sets from a Migration Level 1 Volume

You can move all the data sets from an ML1 volume either to other ML1 volumes or to ML2 volumes. Use the following commands to move all data sets:

FREEVOL MIGRATIONVOLUME(*volser*)

FREEVOL MIGRATIONVOLUME(*volser*) AGE(0)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL1)

FREEVOL MIGRATIONVOLUME(*volser*) AGE(0) -
TARGETLEVEL(MIGRATIONLEVEL1)

FREEVOL MIGRATIONVOLUME(*volser*) -

TARGETLEVEL(MIGRATIONLEVEL2)

FREEVOL MIGRATIONVOLUME(*volser*) AGE(0) -
TARGETLEVEL(MIGRATIONLEVEL2)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(DASD))

FREEVOL MIGRATIONVOLUME(*volser*) AGE(0) -
TARGETLEVEL(MIGRATIONLEVEL2(DASD))

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(TAPE))

FREEVOL MIGRATIONVOLUME(*volser*) AGE(0) -
TARGETLEVEL(MIGRATIONLEVEL2(TAPE))

For this operation, the AGE parameter must be omitted or must be zero. Processing is different for SMS-managed data sets and for non-SMS-managed data sets. For either type of data set, DFSMShsm moves all the migration copies on the volume. Small data sets that move to an ML1 volume can go to SDSP data sets.

Processing for SMS-Managed Data Sets

At the beginning of processing, DFSMShsm obtains a list of the management classes that are active in the system, unless a volume function is processing or has just completed. See "DFSMShsm Retrieval of Management Classes and Storage Groups" on page 138 for a discussion of the exception. DFSMShsm then compares the time since each data set was last opened to the value specified with the LEVEL 1 DAYS NON-USAGE attribute in the management class with which the data set is associated. If the time since the data set was last opened is equal to or more than the value specified with the LEVEL 1 DAYS NON-USAGE attribute, the data set migrates to an ML2 volume. If either the time since the data set was last opened is less than the value specified with the LEVEL 1 DAYS NON-USAGE attribute or the attribute value is NOLIMIT, the data set migrates to another ML1 volume.

DFSMShsm does not use the TARGETLEVEL parameter to determine the migration level to which to migrate an SMS-managed data set. However, if the TARGETLEVEL parameter specifies MIGRATIONLEVEL2 with a device category, any SMS-managed data set that migrates to ML2 migrates to a device of the category specified by the MIGRATIONLEVEL2 subparameter. If the MIGRATIONLEVEL2 subparameter does not specify a device category, any SMS-managed data set that migrates to ML2 migrates to a device of the category specified by the TAPEMIGRATION parameter of the SETSYS command.

Processing for Non-SMS-Managed Data Sets

For non-SMS-managed data sets, DFSMShsm migrates all data sets from the ML1 volume to either:

- ML2 volumes if TARGETLEVEL is not specified
- The level of volumes specified by the TARGETLEVEL parameter

If a device category is specified by the MIGRATIONLEVEL2 subparameter of the TARGETLEVEL parameter, DFSMShsm migrates the data sets to the category of device specified by the MIGRATIONLEVEL2 subparameter. If the

MIGRATIONLEVEL2 subparameter does not specify a device category, DFSMShsm migrates the data sets to the category of device specified by the TAPEMIGRATION parameter of the SETSYS command.

Moving Data Sets from a DASD Migration Level 2 Volume to Other Migration Level 2 Volumes

You can use DFSMShsm to move data sets from one DASD ML2 volume to other ML2 volumes. The FREEVOL command is used to move the data sets.

Use the following commands to move data sets between ML2 volumes:

FREEVOL MIGRATIONVOLUME(*volser*)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(DASD))

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(TAPE))

FREEVOL MIGRATIONVOLUME(*volser*) AGE(0)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2) AGE(0)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(DASD)) AGE(0)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(TAPE)) AGE(0)

FREEVOL MIGRATIONVOLUME(*volser*) AGE(*days*)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2) AGE(*days*)
FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(DASD)) AGE(*days*)

FREEVOL MIGRATIONVOLUME(*volser*) -
TARGETLEVEL(MIGRATIONLEVEL2(TAPE)) AGE(*days*)

For the forms of the command with no AGE specified or with AGE(0) specified, DFSMShsm migrates all SMS-managed and non-SMS-managed data sets to other ML2 volumes. For the forms of the command with an age greater than 0 specified, DFSMShsm moves only the data sets that are equal to or exceed the specified inactive age. The category of volume is specified by the TARGETLEVEL(MIGRATIONLEVEL2) parameter. If a device category is specified with the MIGRATIONLEVEL2 subparameter, the data sets migrate to volumes of that device category. If no device category is specified with the MIGRATIONLEVEL2 subparameter, the data sets migrate to the device category specified by the TAPEMIGRATION parameter of the SETSYS command.

Removing Residual Dump and Backup VTOC Copy Data Sets from Migration Level 1 Volumes

During dump or backup (volume) processing, DFSMShsm creates a data set (the VTOC copy data set), that contains selected information from the VTOC of the volume DFSMShsm is dumping or backing up. These data sets are created on ML1 volumes. The VTOC copy data set drives the recovery of backed up data sets. Occasionally, VTOC copy data sets that are no longer needed remain as residual data sets on ML1 volumes. This happens when the primary volume that was dumped or backed up is removed from the system, and there is no longer a need to recover that volume.

To delete the VTOC copy data sets, use the DATA SET option from the ISMF primary option menu. This option can give you a list of the data sets associated with the primary volume that has been removed from the system. The DELETE command can then be used to remove the VTOC copy data sets.

The following search argument assumes that the SETSYS BACKUPPREFIX is (DFHSM) and that the volume serial number identifies the volume of interest. A data set name search argument of

'DFHSM.VTOC.*.%volser.*'

on the data set selection entry panel lists all the backup VTOC copy data sets. To find the dump VTOC copy data sets, change the second qualifier from VTOC to DUMPVTOC. Enter DELETE (on the data set list panel) in the line operator column next to the VTOC copy data sets you want to remove.

Preventing Further Migration to a DASD Migration Volume

Although the FREEVOL and MIGRATE commands can be used to empty migration volumes, the command does not prevent further use of the volumes. To prevent a migration volume from accepting any further migration copies, issue the following command:

ADDVOL *volser* **UNIT(***unittype***) MIGRATION(DRAIN)**

The DRAIN parameter prevents an ML2 volume from being associated with a key range, thus preventing the volume from being used for migration output. However, if an ML2 volume that has already been associated with a key range is given the DRAIN parameter in an ADDVOL command, that volume remains eligible to be used for migration output.

If you want to prevent such a volume from being used, you must first issue a DELVOL command with the UNASSIGN parameter; then issue an ADDVOL command with the DRAIN parameter. To determine which DASD ML2 volumes are associated with key ranges, use the QUERY MIGRATIONLEVEL2 command.

Moving Data Sets from DASD Backup Volumes

You can move all backup versions of SMS and non-SMS-managed data sets that meet a specified age criterion from DFSMShsm daily backup volumes. Backup versions are moved from a specified daily backup volume to spill DASD or tape volumes. A parameter can be specified to prevent the movement of the most recent backup version for each data set.

Backup versions can be moved from a specific backup volume to spill tape volumes by using the following command:

FREEVOL BACKUPVOL(*volser*) TARGETLEVEL(SPILL(TAPE))

You can selectively move backup versions from a specific backup volume to the type of volumes set up by the SETSYS SPILL parameter where the AGE parameter is greater than zero. The command to move the data set is:

FREEVOL BACKUPVOL(*volser*) AGE(*days*) RETAINNEWEST

Removing Volumes from DFSMShsm Management or Control

You can use the DELVOL command to remove volumes that have been made known to DFSMShsm by the ADDVOL command, or have automatically been made known to DFSMShsm. Volumes are automatically ADDVOLed to DFSMShsm in several ways; for example, an assignment of a scratch tape volume from DFSMShsm management or control, or the use of an SMS-managed primary volume.

The recycle function and the DELVOL PURGE command are the only two ways to reuse a tape volume that has been written by DFSMShsm.

When SMS storage groups are changed, use the DELVOL command to remove, from DFSMShsm control, volumes that were removed from the SMS storage groups before daily automatic functions started. DELVOL these volumes on all DFSMShsm systems that know about them. Do this so that the MVT entries will not be present.

Related reading: For additional information about the DELVOL command, see the *z/OS DFSMShsm Storage Administration Reference*.

Removing Dump Copies and Dump VTOC Copy Data Sets from DFSMShsm

To discard the dump copies associated with a volume that has been removed from DFSMShsm's control, perform the following procedure:

1. Obtain a list of the dump information associated with the DFSMShsm-managed volume by issuing the following command. For *volser*, use the level 0 or ML1 volume that was dumped.

   ```
   LIST VOLUME(volser) BCDS ALLDUMPS
   ```

2. Discard each dump copy by issuing the following command. For *volser*, use one of the dump volumes that is part of the dump copy. The PURGE, REASSIGN, UNASSIGN, and LASTCOPY subparameters of the DUMP command may be specified.

   ```
   DELVOL volser DUMP
   ```

   During the DELVOL process, when the last dump copy that requires the dump VTOC copy data set is deleted from the dump generation, the dump VTOC copy data set is deleted.

3. After you have removed all dump copies associated with a DFSMShsm-managed volume, you can delete the MCP record. This BCDS record is deleted by issuing the following command. For *volser*, use the level 0 or ML1 volume that was dumped.

```
FIXCDS P volser DELETE
```

   The MCP record also contains information that is related to the backup VTOC copy data set. This data set drives the recovery of backed-up data sets. Therefore, do not delete the MCP record if you still want to recover the data that was backed up from the level 0 volume.

Related reading:

- For additional information about the subparameters of the DUMP command, see the *z/OS DFSMShsm Storage Administration Reference*.
- For information about how to delete VTOC copy data sets, see "Removing Residual Dump and Backup VTOC Copy Data Sets from Migration Level 1 Volumes" on page 493.

JES3 Considerations

These considerations apply only to non-SMS-managed volumes.

A DELVOL command for a volume is accepted at any time after initialization. However, use this command only when absolutely necessary. When you must delete a DFSMShsm primary volume that JES3 setup might have reserved for use, leave the volume online if possible. This allows jobs to run that have already been set up to use the deleted volume as one of a pool. If the required data set is actually on another primary volume or if the data set is still migrated, the job can complete successfully. The volume is not selected for subsequent automatic recall of a data set because the volume has been deleted from DFSMShsm.

Avoid the use of the DELVOL command unless you are certain that JES3 has not and will not associate or set up any migrated data sets for recall to the primary volume to be deleted. You cannot readily use the ADDVOL command to return the volume to DFSMShsm's control because the ADDVOL command is processed only during initialization of DFSMShsm in a JES3 environment.

An inadvertently used DELVOL command could prevent a job from running. If a migrated data set is associated with five or fewer pool volumes and one of the volumes is deleted from DFSMShsm's control, the recall is directed to one of the remaining four volumes.

Consolidating Valid Data on Tape Volumes

As time goes by, data on your backup and ML2 tape volumes becomes invalid. Migrated data sets expire or are marked for deletion by DELETE and HDELETE commands. Backup versions roll off or are marked for deletion by BDELETE, HBDELETE, and EXPIREBV commands. To consolidate the valid data on fewer tapes, you can use the DFSMShsm recycle process. You must perform two actions to use the recycle process:

1. You must define the parameters of the recycle process to DFSMShsm. You perform the definition only once.
2. You must direct DFSMShsm to begin the recycle process. You do this each time that you want recycle to run.

Defining the Parameters of the Recycle Process

To define the parameters of the recycle process, you must specify the following information:

- The maximum percent of data remaining valid on a full (backup or ML2) or partial (nonassociated ML2) tape volume and its connected set before the volume is recycled
- The desired number of ML2 partial tape volumes not associated with any migration or recycle output task that remain after the recycle process
- The kind of unit used for output in recycling tape volumes
- Whether the recycle process waits for a tape allocation
- How many recycle tasks can run concurrently
- Whether to duplex recycle output. For ML2 tapes, specify SETSYS DUPLEX(MIGRATION(Y | N)). For backup tapes, specify SETSYS DUPLEX(BACKUP(Y | N)).

The SETSYS command is used for defining the default parameters of the recycle process. The command that is added to the ARCCMDxx member in each DFSMShsm host for the example system is:

```
SETSYS ML2RECYCLEPERCENT(25) RECYCLEPERCENT(25) -
  RECYCLEOUTPUT(BACKUP(ACL),MIGRATION(ACL)) -
  MAXRECYCLETASKS(3) ML2PARTIALSNOTASSOCIATEDGOAL(8)
```

The following discussions of the SETSYS parameters related to recycle processing use the term *connected set*. A connected set is a single tape or a group of 2 to 255 tapes that has the following properties:

- The first tape has no valid data set spanning to a previous tape. (A spanning data set is a data set that is contained on more than one tape, and can span from 2 to 40 tape volumes.)
- The last tape has no valid data set spanning to a successive tape.
- All tapes have either a valid data set spanning to their previous tape, to their successive tape, or to both.
- All tapes are of one function type (for instance, all migration tapes).
- All tapes can be read on the same tape drive (for instance, a 3480X 18-track drive with IDRC compaction).
- A valid data set is contained entirely in only one connected tape set.
- Although a connected set can be up to 255 volumes long, DFSMShsm recycles only those that are 40 or fewer volumes long. For an ML2 partial tape to be considered, it must be the only tape in its connected set.

Input tape drives that are allocated for connected sets remain allocated until all input tapes have been processed.

ML2RECYCLEPERCENT and **RECYCLEPERCENT** control the maximum amount of data that is *valid* on ML2 volumes and backup volumes, respectively, when DFSMShsm automatically selects the volume for processing during recycling. In the example system, when DFSMShsm detects that it has invalidated enough data to bring the amount of valid data on a volume and its connected set to 25% or less, it writes an informational message in the command activity log. When the RECYCLE command is issued for the appropriate category of volume, DFSMShsm automatically selects each connected set of that category that has 25% or less valid data and recycles the complete connected set.

Note: For reel tapes, the percentage is valid blocks divided by total blocks written to this tape. For cartridge tapes, the percentage is valid blocks divided by estimated reuse capacity. This reuse capacity is estimated as the average total blocks written for each cartridge type.

The percentages for tapes can be displayed using either RECYCLE DISPLAY or LIST TTOC commands. (If you use the RECYCLE DISPLAY command, volumes which have 0 percent valid will not be listed. When such a volume is recycled, it will not be mounted, thus the volume does not have to be retrieved to be recycled.)

ML2PARTIALSNOTASSOCIATEDGOAL controls the desired number of partial ML2 tape volumes that are available. The recycle process considers all the ML2 partial tape volumes in the HSMplex that are not associated as output for any migration or recycle task and are not connected to a predecessor tape. If the number of those volumes is greater than eight, the excess volumes over eight are recycled. The tapes with the least amount of valid data (and that meet the other recycling criteria specified for the installation) are selected for recycling.

RECYCLEOUTPUT restricts the category of output device used by DFSMShsm for writing the tape volume that consolidates the data from the tape volumes being recycled. Because you want the output tapes mounted on the 3490 devices with the automatic cartridge loader feature, you use the esoteric name that was specified with the USERUNITTABLE parameter of the SETSYS command.

MAXRECYCLETASKS specifies the maximum number of tape processing subtasks that DFSMShsm will run. The default value is 2, and this value can be dynamically changed even while RECYCLE is running by using the SETSYS MAXRECYCLETASKS(*nn*) command (the minimum number of tasks is 1, the maximum number is 15). In the example system, MAXRECYCLETASKS is set to 3 subtasks (6 tape drives involved).

Notes:

1. Each subtask requires two tape drives. If duplexing, each task requires three tape drives. One drive for the input tape, one for the original tape, and one for the alternate tape.
2. Each subtask, when dealing with single-file-format tape cartridges, keeps the input tape drive allocated for as long as possible (certain error conditions will cause the drive to be deallocated).

RECYCLETAPEALLOCATION specifies whether DFSMShsm is to wait for tape allocation to complete or is to proceed to process other requests. You have already specified NOWAIT in the initial overall SETSYS command.

Related reading: For additional information about how to periodically deallocate an input unit during recycle processing and the effect of the SETSYS parameter RECYCLEINPUTDEALLOCFREQUENCY, see the *z/OS DFSMShsm Storage Administration Reference*. .

Initiating Recycle Processing

You can direct DFSMShsm to recycle a single connected set or to recycle entire categories of volumes. Some commands that recycle tape volumes are:

```
RECYCLE ALL EXECUTE
```

RECYCLE ALL EXECUTE LIMIT(*nnnn*)

RECYCLE ALL EXECUTE PERCENTVALID(*pct*)

**RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(*include_range_list*))
–EXCLUDE(RANGE(*exclude_range_list*)))**

RECYCLE ML2 EXECUTE

RECYCLE ML2 EXECUTE PERCENTVALID(*pct*)

RECYCLE ML2 VERIFY TAPELIST(FULLDSNAME(*tapelist_dsn*))

RECYCLE DAILY EXECUTE

RECYCLE DAILY EXECUTE PERCENTVALID(*pct*)

RECYCLE DAILY(*day*) EXECUTE

RECYCLE DAILY(*day*) EXECUTE PERCENTVALID(*pct*)

RECYCLE BACKUP EXECUTE

RECYCLE BACKUP DISPLAY TAPELIST (PREFIX(*tapelist_prefix*))

RECYCLE SPILL EXECUTE

RECYCLE SPILL EXECUTE PERCENTVALID(*pct*)

RECYCLE VOLUME(*volser*) EXECUTE

RECYCLE ALL OUTDATASET(*dsname*) EXECUTE

The BACKUP parameter allows both daily and spill backup tapes to be selected for input.

The EXECUTE parameter causes DFSMShsm to perform the recycle function of moving valid data sets onto other tape volumes.

The LIMIT parameter quiesces the recycle process after the specified number of tapes are returned to scratch. DFSMShsm calculates the total number of tapes returned by subtracting the number of output tapes used from the number of input tapes freed.

The SELECT parameter adds increased granularity to the RECYCLE eligibility tests. The *include_range_list* and *exclude_range_list* values help determine which tape volumes are candidates to be recycled. These values may be useful whenever you are addressing tape media conversion issues.

When a large number of tapes need to be manually pulled from shelves, the optional TAPELIST parameter produces a list of those tapes. The list is composed of pull groups, which are subsets of tapes, with each group in volume-serial sequence. The pull groups are listed in the sequence in which recycle processing will most likely request them. This projected sequence may differ from the actual processing sequence due to other factors affecting recycle processing, including,

but not limited to: running more than one recycle task, unrecoverable errors, a mixture of device types or connected sets containing more than five volumes.

An alternative VERIFY parameter specifies the same lists as EXECUTE TAPELIST, but does not move any data sets. An alternative DISPLAY parameter specifies a single list in volume-serial sequence, but does not move any data sets.

The OUTDATASET parameter allows you to send RECYCLE messages and, if specified, the DISPLAY output to an output DASD data set rather than to SYSOUT.

The PREFIX and FULLDSNAME parameters allow you to send the tape list output to a DASD data set, rather than to SYSOUT.

For the RECYCLE VOLUME form of the command, DFSMShsm does not check the volume to see whether the amount of valid data is equal to or less than the percent of valid data or any other selection criteria specified in either the RECYCLE command or the SETSYS command; DFSMShsm recycles the volume's connected set, no matter how much of the data on the volume is invalid.

For the forms of the command using recycle categories of volumes (or all DFSMShsm-owned tape volumes), DFSMShsm automatically checks each connected set to determine whether it is eligible to be recycled. A volume is eligible for recycling only if the following conditions are met:

- The tape volume is an ML2 tape or a backup tape, but not an unassigned backup tape.
- A backup tape volume is marked full, or an ML2 volume is either marked full or is a partial tape not associated with any migration or recycle task and not connected to a predecessor tape.
- The percentage of valid data on the volume's connected set is equal to or less than the applicable value. This percentage is specified with the PERCENTVALID parameter, the SETSYS ML2RECYCLEPERCENT parameter, or the SETSYS RECYCLEPERCENT parameter. When these parameters are not specified, the DFSMShsm default is 20%.
- The SELECT criteria, if specified, must be met. That is, all tapes of a connected set must be within the INCLUDE RANGE and none in the EXCLUDE RANGE.
- The volume has not previously failed recycle processing, or if it has failed, now contains no valid data.
- The TTOC for the volume is complete.
- The volume does not have a disaster alternate volume that has been flagged as a disaster alternate volume.

Each volume in an eligible connected set containing data is mounted, and all valid data is transferred to other tape volumes. When the first volume has had all of its valid data transferred, the next volume processed from that category has all of its valid data transferred to the same target volume until the target volume becomes full.

DFSMShsm recycles ML2 volumes to ML2 volumes. It recycles both daily backup volumes and spill backup volumes to spill backup volumes.

DFSMShsm selects and maintains different target tape volumes for recycle, data set migration, volume backup, and volume migration processing. This allows you to run migration, backup, and recycle at the same time.

If you specify PERCENTVALID(0), DFSMShsm selects only those tapes that are already empty but have not yet been returned to scratch. You may want to use PERCENTVALID(0) at those times when tape operators are not available.

Note: Tape drives will not be allocated if you specify PERCENTVALID(0).

PERCENTVALID Calculation

The PERCENTVALID calculation determines the valid data that exists within an entire connected set; it doesn't determine valid data for an individual volume within that set (unless the connected set consists of a single volume). In addition, the first tape within the connected set must meet the current PERCENTVALID specification for the calculation to proceed.

Examples:

```
Volser   % Valid   Predecessor  Successor
------   -------   -----------  ---------
TAPEA       35        -NONE-      TAPEB
TAPEB        5        TAPEA       -NONE-
```

If RECYCLE ALL PERCENT(30) EX is issued, the TAPEA–TAPEB connected set will not be selected, since the first tape in the set does not meet the PERCENTVALID criterion.

```
Volser   % Valid   Predecessor  Successor
------   -------   -----------  ---------
TAPEC       25        -NONE-      TAPED
TAPED       75        TAPEC       -NONE-
```

If RECYCLE ALL PERCENT(30) EX is issued, the TAPEC–TAPED connected set will not be selected. Even though the first tape in the set meets the PERCENTVALID criterion, the entire connected set does not meet the criterion.

```
Volser   % Valid   Predecessor  Successor
------   -------   -----------  ---------
TAPEE        5        -NONE-      TAPEF
TAPEF       45        TAPEE       -NONE-
```

If RECYCLE ALL PERCENT(30) EX is issued, the TAPEE–TAPEF connected set will be selected: the first tape volume in the set meets the PERCENTVALID criterion, and the entire connected set meets the criterion.

Selecting Tape Volumes during Recycling

When you recycle tape volumes, DFSMShsm selects the target tape volume to receive the valid data sets. No other task can be using this tape volume.

Making Cartridge-Type, Single-File-Format Tapes from Reel-Type or Multiple-File-Format Tapes

Many of DFSMShsm's advanced tape operations can be performed only on cartridge-type, single-file-format tape volumes. If you have reel-type or cartridge-type, multiple-file-format, DFSMShsm-owned volumes in your system, you can transfer the data to cartridge-type, single-file format tapes by using the recycle function. Ensure that your RECYCLEOUTPUT parameter specifies a unit type that uses the magnetic tape cartridge. Then you can issue RECYCLE commands either to recycle individual volumes or to recycle categories of volumes. If you recycle categories of volumes, you will want to use the PERCENTVALID parameter to specify a high percentage so that all of the volumes in the category are recycled.

Limiting Data that is Written to a Cartridge-Type Device

When you want to direct data to a cartridge-type tape volume within a single-file-format tape environment, you can specify the percent full from 1 to 9999 on the SETSYS TAPEUTILIZATION(UNITTYPE(unit) PERCENTFULL(pct)) command. When the percent full is reached, DFSMShsm forces end-of-volume function on that volume. This command limits the amount of data that is written on a cartridge-type tape volume, which ensures that the tape can be copied with the TAPECOPY command. This also ensures that the original and the alternate tapes are identical when they are duplexed.

Notes:

1. When you specify SETSYS DUPLEX for either backup or migration, or both, and you specify the NOLIMIT parameter for your tapes, PERCENTFULL is automatically set to 97%. If you designate a specific percentage, such as 100, DFSMShsm uses the percentage that you specify.
2. Other effects of specifying the NOLIMIT parameter with the TAPEUTILIZATION command are that only one buffer is used for recycle processing rather than three, and no action is taken to reduce data set tape volume spanning.
3. You can use the following table to determine the PERCENTFULL value on the SETSYS TAPEUTILIZATION(UNITTYPE(unit)PERCENTFULL(pct)) command.

Output Device	IBM Standard Length Cartridge Percentages	IBM Enhanced Length Cartridge Percentages	Other Cartridge Length Percentages
3490E native attached drive	1–100	1–100	1–9999, when compared to the IBM 3490E cartridge
3590B native attached drive	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3590B emulating 3490E	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3590E emulating 3490E or 3590B	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3590 Model H emulating a 3590 Model B or 3490E	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3592 Model J emulating 3490E or 3590B	1–100	N/A	N/A

Output Device	IBM Standard Length Cartridge Percentages	IBM Enhanced Length Cartridge Percentages	Other Cartridge Length Percentages
Note: If your environment consists of different technology 3590 tape devices configured as stand-alone (non-SMS tape), you must associate an esoteric name with each type of device. You also must request that DFSMShsm use the esoteric names rather than the generic names. This also applies to a mixed 3490 environment.			

4. If backup or migration is using a system-managed tape library, specify the LIBRARYBACKUP or LIBRARYMIGRATION parameters with the SETSYS TAPEUTILIZATION command.
5. If you use virtual tapes in an IBM VTS system for your backup volumes, and if the backups (and any duplexing) in IBM libraries are done exclusively to a VTS by this instance of DFSMShsm, it is recommended that you specify the PERCENTFULL(*pct*) option for the LIBRARYBACKUP parameter with a value of 118%. Specifying PERCENTFULL(118) allows DFSMShsm to fill a virtual 3490 tape to its logical capacity, rather than assuming the presence of inter-record gaps and other tape formatting blocks not actually written to a virtual tape. This also applies to migration to an IBM VTS system. For a detailed description of the SETSYS TAPEUTILIZATION command, refer to the *z/OS DFSMShsm Storage Administration Reference*.

Related reading: For additional information about the LIBRARYMIGRATION parameters, see the *z/OS DFSMShsm Storage Administration Reference*.

Specifying Capacity Utilization

On a CAPACITYMODE-switchable 3590 or 3592 device that is emulating a 3490 device, you have an option when you specify tape capacity. You can specify either to fill tapes to their full capacity or to stop at a point compatible with non-CAPACITYMODE-switchable devices. You specify whether to fill tapes to their full capacity using the SETSYS TAPEUTILIZATION (UNITTYPE(unittype)...CAPACITYMODE(COMPATIBILITY | EXTENDED)) command.

Note: CDS backup, ABARS output, and DUMP tapes do not support this function because they do not have the same block limitation as the supported functions.

DFSMShsm writes to 97% of the capacity of MEDIA5, MEDIA6, MEDIA7, and MEDIA8 tapes unless you specify otherwise through the SETSYS TAPEUTILIZATION command. DFSMShsm uses the reported cartridge type on the physical device to determine the tape's capacity.

Related reading: For more information about commands related to capacity utilization, see *z/OS DFSMShsm Storage Administration Reference*.

Considerations for Using the 3592 Model J Tape Drive

Here are some tips you might find helpful when using the 3592 Model J tape drive.

- 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-Hxx device using enhanced media, you specify a TAPESPANSIZE of 1200 MB. To allow 2% unused tape for a MEDIA5 tape on a 3592 Model J device (no performance scaling), you specify a TAPESPANSIZE of 6000 MB. All size calculations for scaled tapes are based upon the scaled size and not the unscaled size.

- If the speed of data access on MEDIA5 tape is more important than full use of capacity, consider using performance scaling. Performance scaling allows 60 GB of physical (and more logical) data 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 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 MEDIA6, MEDIA7, or MEDIA8 tape cartridge media.

- If your installation is using MEDIA5 tapes with performance scaling, consider using MEDIA7 tapes for high performance functions. The available MEDIA5 tapes could then be used to their full capacity. DFSMShsm recycle processing of 3592 tapes can take significantly longer than the recycle of smaller tapes because the amount of data moved at the same RECYCLEPERCENT can be much larger. Besides moving more data, the likelihood of a tape takeaway for recall processing increases with the number of data sets still remaining 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 recycled require the least processing time.

  Another option to consider is decreasing the ML2RECYCLEPERCENT parameter, the RECYCLEPERCENT parameter, or both. For example, if your installation uses performance scaling (60 GB) for ML2, and you would like to move no more than 6 GB of data when an ML2 tape is recycled, set ML2RECYCLEPERCENT(10). If your installation uses full capacity (300 GB) for backup tapes and you would like to move no more than 6 GB of data when a backup tape is recycled, set RECYCLEPERCENT(2). These examples assume that the ML2 and Backup tapes in the installation are filled to capacity because the calculations are based upon 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 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.

  While lowering the recycle threshold reduces recycle processing time and decreases the number of times each tape must be recycled, it may also increase the overall number of tapes needed in your installation. Also, if you have a mix of ML2 or Backup tape capacities in need of recycle processing, you may want to recycle tapes with the RANGE parameter and use the appropriate recycle threshold for the tape capacities in the range.

  In an SMS tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select WORM tapes for ABACKUP processing. The output data set prefix 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 is to allow tape pooling to determine if ABARS data sets go to WORM or R/W media. Optionally, if the DEVSUP parameter ENFORCE_DC_MEDIA=ALLMEDIATY or ENFORCE_DC_MEDIA=MEDIA5PLUS is used, then the data class must request the appropriate media type for it to be successfully mounted.

- If your installation is running OS/390 V2R10, consider using the Fast Subsequent Migration function to reduce the need to RECYCLE these high-capacity tapes.
- If your installation is running z/OS V1R3 or higher in a Sysplex environment, consider using the Common Recall Queue to optimize mounts of migration tapes.
- AUDIT MEDIACONTROLS for a FAILEDCREATE situation usually only needs to look at the last few files on a tape. If it is available for your system level, make sure Audit APAR OA04419 is applied.
- To fully utilize MEDIA5 tapes, unless you have a specific need to run in 22-bit FBID mode, operate 3592 Model J tape drives emulating device type 3490 in 32-bit FBID mode. MEDIA6, MEDIA7, and MEDIA8 tapes cannot be used on a 3592 Model J emulating device type 3490. The 3592 Model J tape drive always operates in 32-bit FBID mode when it is emulating device type 3590.
- If you decide to change the performance scaling attribute for a particular function and partially full MEDIA5 tapes already exist, mark full the partial MEDIA5 tapes for that function. This is necessary because OPEN processing carries over the performance scaling attribute from partial tapes to the next mounted empty tape as long as the tape is mounted under the same allocation. When the function runs the next time, it will need to mount an empty tape and the performance scaling change can take effect.

Limiting the Number of ML2 and Backup Data Sets that Span Tape Volumes

The TAPESPANSIZE optional parameter of the SETSYS command can be used to reduce the occurrences of data sets spanning tape volumes. This is of benefit when running certain functions such as recall, recover, and recycle. When a data set spans tape volumes, a recall or recovery of that data set requires two tape mounts rather than one. And if you are recycling a connected set of tape volumes, recycle determines if the first tape and the average of the remaining tapes meet the selection criteria. If not, no tapes in the connected set are recycled, potentially leaving some tapes with very little data on them.

With the SETSYS TAPESPANSIZE command, you can specify a value between 0 and 4000. This value represents the maximum number of megabytes of tape that DFSMShsm can leave unused while trying to eliminate spanning data sets. To state this differently, this value is the minimum size of a data set that is allowed to span tape volumes. Data sets whose size is less than the value do not normally span volumes. Only those data sets whose size is greater than or equal to the specified value are allowed to span volumes. The data written to tape results from compression by the tape subsystem, so predictions of that resultant size are not always correct. Therefore, spanning can occur for some data sets that do not compress as expected.

When data is being written to tape, DFSMShsm estimates whether the space remaining on a tape can contain the entire data set, based on the following criteria:

- The limit that is specified with the SETSYS TAPEUTILIZATION command
- Whether the data set is already compressed on DASD

- The compression ratio that is assumed for your data sets
- Whether (for a logical 3490 device) the new data set would exceed the architected block count

If TAPEUTILIZATION specifies the NOLIMIT parameter, no action is taken to reduce the spanning of the data set tape volume.

Note: The use of the NOLIMIT parameter is not recommended.

Related reading:

- For additional information about tuning, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the spanning of the data set tape volume, see the *z/OS DFSMShsm Storage Administration Reference*.

How to Handle Inconsistencies between Tape Media Contents and OCDS TTOC Records

When DFSMShsm detects an inconsistency between the TTOC record and the content of the tape media, an ARC0378I message is issued during the DFSMShsm startup to notify the system programmer that the missing TTOC record data set entries need to be resolved.

To get a list of all the volumes whose TTOC records are missing data set entries:

1. Issue a LIST TAPETABLEOFCONTENTS command with the SELECT(FAILEDCREATE) parameter. Inconsistent TTOC records for a failing DFSMShsm host will not be listed until DFSMShsm is restarted in that z/OS image because the records appear to still be in use.
2. Use the extended AUDIT MEDIACONTROLS function to reconstruct the missing TTOC data set entries for the indicated tape volume.

   If data checks occur or invalid file block identifiers are encountered during AUDIT MEDIACONTROLS processing, a RECYCLE FORCE command can be issued to recycle the known valid data set entries on a tape volume. A warning message (ARC0378I) is issued, and the recycled tape volume is not deleted.

   In general, you should *not* use the RECYCLE FORCE command to recycle a tape volume before using the AUDIT MEDIACONTROLS command to resolve the missing TTOC data set entries. If you do use it, you must *immediately* perform the following steps to avoid losing the valid data sets on the just-recycled tape volume:

   a. Issue a LIST DSN MCDS SELECT(VOLUME(volume serial number of recycled migration tape)) command, or a LIST DSN BCDS SELECT(VOLUME(volume serial number of recycled backup tape)) command to list the data set names of the data sets that DFSMShsm indicates still reside on the tape volume.
   b. Recall or recover the indicated data sets that do not have TTOC data set entries associated with valid data sets residing on the tape volume.
   c. Issue a DELVOL command for the recycled tape volume.

When an ML2 tape volume cannot be deleted because an ARC0260I message indicates that valid data may exist on a volume, it is because the OCDS TTOC record indicates that the volume may contain valid migration copies. This condition occurs when an inconsistency exists between the TTOC record and the content of the tape media. Inconsistencies are created whenever TTOC creation or

TTOC invalidation information is lost because of a DFSMShsm cancel, a system failure due to a power outage, or an I/O error on the OCDS.

A TTOC creation inconsistency is present when the ARC0260I message is preceded by an ARC0378I message that relates to the same volume. TTOC creation inconsistencies are resolved by running the extended AUDIT MEDIACONTROLS function to reconstruct missing data set entries on the OCDS TTOC record that relates to the tape volume. After the extended AUDIT MEDIACONTROLS function is completed for the inconsistent tape volume, the normal generic RECYCLE function established by the installation will DELVOL the tape volume after all valid data is moved to another ML2 tape volume. If an immediate DELVOL of the tape volume is necessary, the single volume RECYCLE function will DELVOL the tape volume after all valid data is moved to another ML2 tape volume.

A TTOC invalidation inconsistency is present when the ARC0260I message is not preceded by an ARC0378I message that relates to the same volume. TTOC invalidation inconsistencies are resolved by the normal generic RECYCLE function established by the installation. If an immediate DELVOL of the tape volume is necessary, the single volume RECYCLE function can be used. Either way, the RECYCLE function will DELVOL the tape volume after all valid data is moved to another ML2 tape volume.

Recovering a Damaged DFSMShsm Single-file Migration Level 2 Volume

For information on recovering the undamaged remains of a single-file tape after it has been partially overwritten, see *z/OS DFSMShsm Storage Administration Guide*.

Making Migration Level 2 Key Range Volumes Available

When the ML2 volume for a specific key range becomes full during migration, DFSMShsm tries to assign a new volume to the key range. If no ML2 volumes are available, you can do one of the following to make volumes available:

- Use the ADDVOL command to add more ML2 volumes to DFSMShsm.
- Use the UNASSIGN parameter of the DELVOL command for each ML2 volume you want to be made available for reuse. When you use this parameter, specify DASD ML2 volumes that have been used before and that you want to reuse now. Ensure that space is available on these volumes because of subsequent recalls and deletions.
- Use a DEFINE command to assign a volume to a key range. To associate a specific volume with a key range, use the DEFINE command. Issue a QUERY ML2 command to determine the current association of volumes to key ranges. Then enter a DEFINE command specifying the same set of volumes but with the new volume replacing the old volume associated with the key range.

These three commands (DELVOL, ADDVOL, and DEFINE) allow you to reuse the current level 2 volumes.

Recovering a Migrated Data Set

A restriction was noted in the discussion on recovering backup versions of data sets: DFSMShsm cannot recover a data set to a migration volume.

Note: DFSMShsm can recover a non-VSAM data set that is currently marked as migrated in the computing system catalog if a recover data set name is issued with NEWNAME specified, and the NEWNAME data set is not a migrated data set.

To recover a backup version of a migrated data set, use one of the following procedures:

- Recall the migrated data set and recover a backup version of that data set. You can use the REPLACE parameter if necessary.
- Delete the migrated data set and recover a backup version of the data set you just deleted.
- Issue a recover data set name with NEWNAME(*newdsname*) specified. If the NEWNAME(*newdsname*) is not a migrated data set, it will successfully recover.

Recovering a Migration Volume

You can use the RECOVER command to recover a data set to a level 0 volume. The RECOVER command recovers a data set in its original format. Because the migrated copy of a data set is stored in a different format from the original user's data set, you cannot use the RECOVER command to recover a data set to a migration volume. Depending on whether you use DFSMShsm commands to periodically dump your DASD migration volumes, you can use one of the following procedures to recover an entire migration volume.

Note: These procedures assume that you have been using DFSMShsm to back up data sets because the procedures recover the latest backup version of each data set that was on the migration volume.

Be aware that the latest backup version of each data set might not be identical to the data set that DFSMShsm migrated. For example, the data set might have needed backup when DFSMShsm migrated it, but DFSMShsm did not have a chance to back up the migrated data set before the migration volume became inaccessible.

If you use DFSMShsm's BACKVOL command to periodically dump your DASD ML2 volumes and use the automatic dump facility to periodically dump your ML1 volumes, perform "Procedure 1" to recover the data, mostly to a DASD migration volume.

If you do not dump your DASD migration volumes or need to recover the data sets that were on a tape migration volume and have no alternate tape, perform "Procedure 2" on page 509 to recover all the data to primary volumes. For a discussion of alternate volumes, see Chapter 9, "Making Disaster Backup Copies of DFSMShsm-Owned Tape Volumes," on page 401.

Procedure 1

1. To ensure that DFSMShsm does not select the volume being recovered as a target volume for migration, issue the DFSMShsm ADDVOL command with the DRAIN parameter. Issue the command in all DFSMShsm hosts.
2. Restore the latest dump copy of the volume. This restores the DASD migration volume to its level when the dump copy was created. Issue the following DFSMShsm command to request a full volume restore of the volume.

```
RECOVER * TOVOLUME(volid) UNIT(unittype) FROMDUMP
```

3. If the migration volume contains SDSP data sets and you have your own procedures to back up these data sets more frequently than you dump the volume, recover the most recent backup of the SDSP data set to the migration volume being recovered.
4. After the volume has been successfully restored, issue the FREEVOL command. This leaves data on the migration volume that is no longer valid, migrated data. It also enables you to easily obtain a list of the data sets that still need to be recovered (those that the MCDS indicates are still on the migration volume). The FREEVOL command process is driven off of the volume table of contents (VTOC) and includes all data sets in an SDSP data set on the volume that are currently migrated. Use the AGE(0) parameter to indicate that all data sets must be migrated. You can specify either ML1 or ML2 as the target migration level when you want to process a ML1 volume. Use the following command as an example:

   ```
   FREEVOL MIGRATIONVOLUME(volser) TARGETLEVEL(MIGRATIONLEVEL1) AGE(0)
   ```

5. Request from DFSMShsm a list of all the migrated data sets that are indicated as being on the volume being recovered. These data sets represent the data sets that were migrated to the volume after the most recent dump copy was created. The migration copies of the data sets have been lost, so the data sets must be recovered from backup versions. Use the following LIST command to request this list of data sets:

   ```
   LIST DATASETNAME MIGRATIONCONTROLDATASET -
   SELECT(VOLUME(volser)) OUTDATASET(dsname)
   ```

   The OUTDATASET parameter is highly recommended so that the output from the LIST command is written to the data set whose name is given by *dsname*. This allows you to use a text editor to modify the output to easily perform the remaining steps:
6. Make one more copy of the output data set, *dsname*.
7. For each data set in the output from 5, perform steps 8 and 9. You cannot perform these steps concurrently for the same data set. By using two copies of the list of data sets from step 5, you can modify each of the two lists to perform steps 8 and 9 for all of the data sets at one time. Use a text editor and its global change command to add and remove the necessary text around each data set name.
8. Delete the migrated data set. This uncatalogs the data set and deletes the MCDS data set record and any associated MCL, MCA, and MCO records. Do this with the following DFSMShsm command:

   ```
   DELETE dsname
   ```

9. Recover the latest backup version for each data set. Do this with the following DFSMShsm command:

   ```
   RECOVER dsname REPLACE
   ```

   Note: The backup copy that is recovered may not be the most current version of a data set if the data set was migrated to the ML1 volume before it was backed up.

10. If the recovered volume is an ML1 volume, there might be backup versions that were restored to the migration volume. Some of these may represent valid backup versions. These can exist if data set backup requests were processed between the time automatic backup last ran on the primary DFSMShsm host before the volume was dumped and the time when the volume was dumped. You can use the following procedure to assist you in recovering the backup versions, see "Recovering Backup Versions from a Damaged Migration Level 1 Volume" on page 510.
11. After all valid, DFSMShsm-owned data sets have been removed from the volume, you can reinitialize the volume. DFSMShsm no longer requires access to it. You can also remove the volume from DFSMShsm's control with the DELVOL command.

The data sets will probably migrate the next time automatic primary space management occurs on the DFSMShsm-managed volumes. This happens because when DFSMShsm recovers a data set, the data set VTOC entry is restored exactly as it was when the backup version being recovered was created. DFSMShsm does not reset the date-last-referenced field to the current date when it recovers the data set unless you rename the data set. Therefore, the recovered data sets will probably have a high inactive age.

Procedure 2

Rule: Do not start any step until the previous steps have completed.

1. Determine which data sets were on the migration volume. To do this, use the following command:

```
LIST DSN MCDS SELECT(VOLUME(volser)) -
OUTDATASET(dsname)
```

   For *volser*, specify the volume serial number of the migration volume. DFSMShsm writes the list in the data set whose name is *dsname*. You can issue this command from a console, terminal, or batch job.
2. Do a LISTCAT command if you want to save information about the data sets that you are going to delete in the next step. You may want to know, for example, whether the status of a generation data set (GDS) is DEFERRED or ACTIVE before it is deleted, because after being deleted it is always recovered with a status of DEFERRED.
3. Delete the MCDS data set record. This uncatalogs the data set and deletes the MCDS data set record and any associated MCL, MCA, and MCO records. To do this, use the following command for each data set on the migration volume:

```
DELETE dsname
```

   When you use a text editor such as ISPF, you can edit the data set produced by the LIST command (step 1) so it contains one line for each data set. Each line contains the data set name. Replicate this data set once because you will need a copy for the next step. Then, use a global change command to precede each data set name with the characters HSENDCMD DELETE. A DFSMShsm-authorized user can then run the data set as a command procedure (CLIST). However, if you have many data sets, include the necessary JCL in the data set and submit the data set as a batch job.
4. Recover the latest backup version for each data set that migrated to the migration volume. To do this, issue the following DFSMShsm command:

```
RECOVER dsname
```

When you use a text editor such as ISPF, you can edit the copy of the data set containing the name of each data set on a separate line. Then, use a global change command to precede each data set name with the characters HSENDCMD RECOVER. A DFSMShsm-authorized user can then run the data set as a command procedure (CLIST). If you have many data sets, include the necessary JCL in the data set and submit the data set as a batch job.

5. Issue the IDCAMS ALTER ROLLIN command if you want to change the status of a recovered GDS from DEFERRED to ACTIVE.

   The catalog status of a recovered GDS is dependent upon its catalog status before being recovered. If the GDS is not cataloged before recovery, its catalog status after recovery is DEFERRED. If the GDS is cataloged before recovery, its catalog status after recovery remains the same, with the following exception: when DFSMShsm is used as a data mover to recover a rolled-off GDS, the cataloged status of the GDS, after it is recovered, is DEFERRED.
6. Follow the procedure titled "Recovering Backup Versions from a Damaged Migration Level 1 Volume" if the volume being recovered is an ML1 volume and there are backup versions that must be recovered or replaced.

The data sets will probably migrate the next time automatic space management occurs on the DFSMShsm-managed volumes. This happens because when DFSMShsm recovers a data set, the data set VTOC entry is restored exactly as it was when the backup version being recovered was created. DFSMShsm does not reset the date-last-referenced field to the current date when it recovers the data set. Therefore, the recovered data sets will probably have a high inactive age.

Related reading: For additional information about the LISTCAT command, see *z/OS DFSMS Access Method Services for Catalogs*.

Recovering Backup Versions from a Damaged Migration Level 1 Volume

Use the following steps to recover backup versions from a damaged migration level 1 volume.

1. Issue the following Access Method Services command to print a list of MCM records. Use this list to determine if any of the backup versions on a damaged or restored ML1 volume are still valid.

   ```
   PRINT IDS(bcds-dsname) FROMKEY(X'26') -
   TOKEY(X'26') OUTFILE(ddname)
   ```

   For *bcds-dsname*, substitute the fully qualified data set name of the DFSMShsm backup control data set. For *ddname*, substitute the DDname of a preallocated data set that contains the output from the PRINT command.
2. Examine each MCM record that was printed to see if it represents a backup version that resides on the migration volume being recovered. To do this, look at offset 152 (X'98') in the record for the ML1 volume serial number. For all MCM records that have the volume serial number of the volume being recovered, you should perform the following. For *dsname* in the following commands, use the original data set name found at offset 64 (X'40') of the MCM record.

a. Use the following DFSMShsm command to find out the version number of the backup version indicated as being on the volume being recovered:

```
LIST DATASETNAME(dsname) BACKUPCONTROLDATASET
```

b. Using the version number found in the previous step, issue the following DFSMShsm command to delete that backup version:

```
BDELETE dsname VERSIONS(bvn)
```

c. Issue the following DFSMShsm command to create a replacement backup version of the data set:

```
BACKDS dsname
```

An optional first step might be appropriate if the number of backup versions on the migration volume is large or if the original data sets no longer exist from which to create new backup versions. The movement of the backup versions phase of automatic backup, run in the primary DFSMShsm host, moves all of the valid backup versions to backup volumes. Allowing automatic backup to run first moves all backup versions off all ML1 volumes so that only the MCM records for the damaged volume must be examined. The remainder of the procedure must still be followed for lost backup versions. If the migration volume was dumped, backup versions might have been lost if placed on the migration volume after the dump copy was created. If the migration volume was not restored from a dump copy, backup versions could still be lost if they resided on the parts of the volume that were damaged.

Returning Dump Tapes to Use

Dump tapes in dump classes with the NOAUTOREUSE parameter specified are not automatically returned to usability when their generation is rolled off or expires. To return these tapes to use:

1. Physically return the tape volumes to the computer site.
2. Issue the following command for each tape volume:

```
DELVOL volser DUMP(REASSIGN)
```

3. Determine if the volumes are ready for reuse. If you have specified the TAPEDELETION(HSMTAPE) parameter of the SETSYS command, the volumes are ready for reuse. If you have specified the TAPEDELETION(SCRATCH) parameter, you migh be required to reinitialize the volumes before they are returned to the scratch pool.

Managing SDSP Data Sets

Three procedures must be considered in managing SDSP data sets:

- Reorganizing the data sets
- Backing up and recovering the data sets
- Removing the data sets

Reorganizing SDSPs

Migration of user data sets into an SDSP causes VSAM CI and CA splits. Erasing the VSAM records of recalled and deleted data sets in an SDSP makes space

reusable. However, these dynamics cause a redistribution of the free space in the SDSP. The SDSP will eventually appear to be full even though some free space may exist within the SDSP. When the high-used relative byte address equals the high-allocated relative byte address, the SDSP can appear to be full to a migration request if the SDSP cannot be extended. If DFSMShsm fails in migrating to an SDSP, it issues a message to the migration activity log, and stops directing migrations to that SDSP.

For discussions of selecting SDSPs and the causes of out-of-space conditions, see "SDSP Selection" on page 60 and "Out-of-Space Condition during Migration" on page 60.

In this case, and after migration cleanup has run, you should consider reorganizing the SDSP. The frequency of reorganization depends on the size and number of SDSPs and on the number of data sets that regularly migrate to SDSPs in your environment. You should have enough SDSPs so that when one becomes full, the migrations occur to other SDSPs while you schedule and perform the SDSP reorganization. Use the TSO LISTC command once per week to determine the status of SDSPs as reported by the high-used and high-allocated relative byte addresses. After observing the SDSP status for several weeks, you can establish an appropriate frequency for SDSP reorganization in your environment. You may also choose to expand the SDSP, should you encounter an out-of-space condition. Alternatively, a reorganization of the SDSP can be automatically triggered when DFSMShsm issues message ARC0542I to the console.

The format of the SDSP data set name is *uid*.SMALLDS.V*volser*. Where

- *uid* is a high-level qualifier that is the DFSMShsm-authorized user ID from the startup procedure.
- SMALLDS and V are constants.
- *volser* is the volume serial number of the volume where the SDSP resides.

Note: An SDSP data set name is essentially a fixed name relating to a specific volume serial number in your environment and, once established, should not be changed.

You can reorganize the SDSP data set by using the Access Method Services EXPORT and IMPORT commands. Then, use the DFSMShsm ADDVOL command to add the ML1 volume to DFSMShsm again and specify the SMALLDATASETPACKING parameter to make the volume eligible for SDSP data set selection again.

Related reading:

- For additional information about the SDSP data set name is generated, see the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the EXPORT and IMPORT commands and how to use them, see *z/OS DFSMS Access Method Services for Catalogs*.

SDSP Backup and Recovery

You should back up your SDSP data sets daily, such as at the end of automatic space management. The frequent backup is desirable because of the large number of user data sets whose data is stored in an SDSP data set. An SDSP data set is an active data set and is dynamically changing. The more frequently you back it up, the more current it will be if it must be recovered.

Example: If there are 50 000 data sets in an SDSP data set and it is lost, DFSMShsm has to do 50 000 data set recovers. If you have backed up the SDSP, you can recover the SDSP from your latest backup and then recover only a few user data sets that have migrated to it since your latest backup of the SDSP.

DFSMShsm does not automatically back up an SDSP data set as a single entity, but it can back up the individual data sets that the SDSP data set contains. However, you can use DFSMShsm commands to back up and recover an SDSP data set. To back up an SDSP data set, issue the BACKDS command and include only the data set name of the SDSP.

To recover an SDSP that has been backed up by DFSMShsm, do the following:

1. Issue a DELVOL MIGRATION command for the ML1 volume on which the SDSP resides. DFSMShsm's recovery function does not recover backup versions to migration volumes, so the ML1 volume must be temporarily removed from DFSMShsm's inventory of migration volumes.
2. Issue a RECOVER command for the SDSP data set.
3. Issue an ADDVOL command for the ML1 volume after the RECOVER command has been processed successfully. This adds the volume back to DFSMShsm's inventory of migration volumes.

Removing SDSPs

There are two different methods of removing SDSPs: removing an entire ML1 volume that contains the SDSP or removing the SDSP from the ML1 volume that it resides on.

Removing a Migration Level 1 Volume That Contains an SDSP

The following steps remove all data sets from an ML1 volume that includes an SDSP:

1. Remove all migrated data sets from the ML1 volume using the FREEVOL command. The data sets will select a target volume using the same selection process DFSMShsm uses during migration. Small data sets that move to another ML1 volume may go into an SDSP data set. The FREEVOL process cleans up data sets in the SDSP in need of scratching.
2. Verify that all valid data sets are out of the SDSP by using the following command:

   ```
   LIST dsname SEL(VOL(ml1volser) SDSP) TERM
   ```

   If there are data sets in the SDSP that do not move and are no longer wanted, try to delete them. It is important that you do not leave the control records for a data set in DFSMShsm's system when you delete the SDSP.
3. Delete the SDSP data set using IDCAMS.
4. Check the ML1 volume to verify the SDSP and all user data sets have been moved off the volume. Use ISPF 3.4 or the LIST VTOC command to display the data set list for the ML1 volume. If the ML1 volume contains BACKDS data sets, automatic backup should be run on the primary DFSMShsm host to move them to a backup tape.
5. Issue the following DELVOL command for the the ML1 volume. This disassociates the ML1 volume from each DFSMShsm host and removes the ADDVOL for the ML1 volume from each startup parmlib.

   ```
   HSEND DELVOL ml1volser MIGRATION
   ```

Removing an SDSP from a Migration Level 1 Volume

The steps for removing only the SDSP from an ML1 volume are:

1. Code the ARCMMEXT user exit to select only the data sets in the SDSP that will be migrated during the FREEVOL. Use the FREEVOL command to move all the valid data sets in the SDSP to another volume. The data sets in the SDSP will select a target volume using the same selection process DFSMShsm uses during migration. Small data sets that move to another ML1 volume may go into an SDSP data set.
2. Verify that all valid data sets are out of the SDSP by using the following command:

   ```
   LIST dsname SEL(VOL(ml1volser) SDSP) TERM
   ```

   If there are data sets in the SDSP that do not move and are no longer wanted, try to delete them. It is important that you do not leave the control records for a data set in DFSMShsm's system when you delete the SDSP.
3. Delete the SDSP data set using IDCAMS.
4. Use ISPF 3.4 or the LIST VTOC command to display the data set list for the ML1 volume. Check this list to verify the SDSP has been moved off the volume.
5. Issue the following ADDVOL NOSDSP command for the source ML1 volume. Update all startup parmlibs with an ADDVOL command for the ML1 volume to reflect the NOSDSP status.

   ```
   ADDVOL ml1volser UNIT(unittype) MIGRATION(ML1 NOSDSP)
   ```

6. Update the SETSYS SDSP parameter to reflect a NOSDSP status if you are removing all SDSPs from the DFSMShsm system.

Protecting DFSMShsm-Owned Tape Volumes

You can protect DFSMShsm-owned tape volumes with password protection, RACF protection, expiration date protection, or a combination of the three.

Related reading: For additional information about protecting DFSMShsm-owned tape volumes, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Converting to a New Tape Technology

If you are converting from an old tape technology to a new tape technology, identify whether the old tapes have the same logical device type as the new ones. Then mark the old tapes full so that they are no longer available for output consideration. With DFSMShsm support of the 3590E device, you do not have to perform this step, because DFSMShsm can distinguish 3592, 3590E, 3590B, 3590H, and VTS tapes from other type cartridges. However, DFSMShsm does not differentiate between other hardware that has a common logical device type.

Use the following command to identify the old backup tapes:

```
LIST TTOC SELECT (BACKUP NOTFULL)
```

Use the following command to mark the old tapes as full:

```
DELVOL volser BACKUP(MARKFULL)
```

Requesting DFSMShsm Service from Your Applications

DFSMShsm has macros available that allow you to request DFSMShsm service from your application programs. You provide information to the macro and execute it from your application program. The macro builds the required DFSMShsm control information and issues the request for DFSMShsm service.

Related reading: For additional information about the macros, see *z/OS DFSMShsm Managing Your Own Data*.

Chapter 14. Obtaining Information from DFSMShsm

In using DFSMShsm, you may have occasion to require information about what is happening in the processing. DFSMShsm provides extensive facilities for determining what the conditions are in the product. The DFSMShsm log and the activity logs have already been mentioned. In addition to the logs, you can obtain the following from DFSMShsm:

- Information from the control data sets
- Information about the present operating conditions in the DFSMShsm address space
- Information about availability of output tape volumes
- Statistical information about DFSMShsm operations
- A trace of DFSMShsm's processing
- Information about the conditions in the system if a specific error occurs
- Determining the results of volume space management data set selection without actually moving data sets

Related reading: For additional information about using DCOLLECT for reporting, see the *z/OS DFSMSdfp Storage Administration Reference*.

Obtaining Information from the Control Data Sets

Use the LIST command to obtain information from the control data sets. Information can be about:

- Backup volumes
- Copy pools
- Data sets
- Dump classes
- Dump volumes
- DFSMShsm host serialization
- DFSMShsm-managed volumes
- DFSMShsm-owned volumes
- Tape volumes
- User authorization

In obtaining the information, you can receive it in any of three ways: as a printed listing, as a DASD data set, or as terminal output. For small amounts of data, such as DFSMShsm host serialization information or user-authorization information, terminal output is satisfactory. However, the LIST command is capable of producing very large amounts of information, for example, when you request information about all migration volumes. For such requests, you want to be sure to direct the output to either a printed listing or a DASD data set that you can edit or print later.

The parameters that control the destination for the LIST command output are the same for any request. In showing the forms of the command for each type of output, all other parameters (such as request) are being identified.

The form of the command for obtaining terminal output is:

LIST (request) TERMINAL

The form of the command for obtaining a printed listing is:

© Copyright IBM Corp. 1984, 2004

LIST (request) SYSOUT

LIST (request) SYSOUT(*class*)

The first form of the command directs the printed output to the SYSOUT class specified with the SETSYS command. The second form of the command specifies a SYSOUT class to which the output is directed.

The form of the command to direct the output to a DASD data set is:

LIST (request) OUTDATASET(*dsname*)

dsname is the fully qualified name of the data set to which you want the LIST command output sent. If the data set exists, DFSMShsm places the report at the end of the existing data set. If the data set does not exist, DFSMShsm dynamically allocates and catalogs an output data set.

Related reading: For additional information about the LIST command, see the *z/OS DFSMShsm Storage Administration Reference*.

Obtaining Backup Volume Information

The forms of the requests for obtaining backup volume information are:

LIST BACKUPVOLUME

LIST BACKUPVOLUME SELECT(EMPTY)

LIST BACKUPVOLUME(*volser*)

The first request lists the information about all the backup volumes in the system. The second request lists the information about all the empty backup volumes in the system. The second request can be used to determine which DFSMShsm tapes can be placed in the cartridge loader. The third request lists the information about a specific backup volume.

Obtaining Data Set Information

You can obtain information about a single data set or about every data set for which DFSMShsm has any record. You can also select groups of data sets. Optional parameters allow you to select:

- Information from the backup control data set
- Information from the migration control data set
- Information from both the backup and the migration control data sets
- Information about data sets that have been recalled
- Information about data sets with a certain minimum age, maximum age, or between a minimum and a maximum age
- Information about data sets on ML1 volumes
- Information about data sets on ML2 volumes
- Information about the designated data sets on a specific migration volume
- Information about the data sets in SDSP data sets
- Information about the data sets not in SDSP data sets
- Information about VSAM data sets
- A summary of the information about the data sets

Many permutations exist for requesting information about data sets and levels. Some of them are shown here.

Sample requests for information about data sets are:

LIST DATASETNAME BACKUPCONTROLDATASET

This request obtains information about all data sets that are recorded in the backup control data set.

LIST DATASETNAME(*dsname*) MIGRATIONCONTROLDATASET

This request obtains information about the data set identified by *dsname* from the migration control data

LIST DATASETNAME BOTH INCLUDEPRIMARY

This request obtains information about all data sets recorded in both the backup control data set and the migration control data set. In addition, it obtains information about any data sets that have migrated and been recalled.

LIST LEVEL(*levelid*) MIGRATIONCONTROLDATASET -
SELECT(MIGRATIONLEVEL1)

This request obtains information about data sets with the qualifier *levelid* recorded in the migration control data set, but only for the data sets on ML1 volumes.

LIST DATASETNAME MIGRATIONCONTROLDATASET -
SELECT(SMALLDATASETPACKING)

This request obtains information about all data sets recorded in the migration control data set as being migrated to SDSP data sets.

Related reading: For additional information about the LIST command, see the *z/OS DFSMShsm Storage Administration Reference*.

Obtaining Copy Pool Information

You can obtain information about all copy pool fast replication backup versions by using the following command:

```
LIST COPYPOOL
```

This request provides version information for each copy pool that has at least one attempted backup. The list of source and target volume pairs is not included as part of this list request. The list output consists of the following information:

- Copy pool name
- Version number
- Valid or invalid
- VTOCENQ
- Token
- Backup date and time

You can obtain information about a specific copy pool by using the following command:

```
LIST COPYPOOL(cpname)
```

This request provides detailed version information for each valid or invalid backup version for the specified copy pool. The date and time fields are zero when the PREPARE subparameter is used with the FRBACKUP command. The list output consists of the following information:

- Copy pool name
- Version number
- Valid or invalid
- VTOCENQ
- Token
- Backup date and time (for Prepare versions these fields display a value of zero)
- Source and target volume pairs

You can obtain information about the volumes contained in a specific copy pool backup storage group by using the following command:

```
LIST COPYPOOLBACKUPSTORAGEGROUP(cpbsgname)
```

This request provides a list of each volume in the copy pool backup storage group that DFSMShsm has selected as a target volume. Volumes that are defined to the storage group that are not listed have not been selected as target volumes. In addition, the name of the source volume that earch target volume is a target for will be listed.

You can obtain information about which copy pool a volume is associated with by using the following command:

```
LIST PRIMARYVOLUME BCDS
```

This request provides information about the copy pools that a primary volume is associated with. Specify either the BCDS or BOTH parameter for this request to process correctly. Use the LIST COPYPOOL(cpname) command to determine which target volumes are associated with a particular source volume.

Obtaining Dump Class Information

You can obtain information about either all dump classes or an individual dump class. The forms of the request for dump class information are:

LIST DUMPCLASS

LIST DUMPCLASS(*class*)

Obtaining Dump Volume Information

You can request information about all dump volumes or specific dump volumes. In addition, you can select the volumes with specific characteristics. The forms of the requests for information about dump volumes are:

```
LIST DUMPVOLUME

LIST DUMPVOLUME -
  SELECT(AVAILABLE)

LIST DUMPVOLUME -
  SELECT(UNAVAILABLE)

LIST DUMPVOLUME -
  SELECT(EXPIRED)

LIST DUMPVOLUME -
  SELECT(UNEXPIRED)

LIST DUMPVOLUME -
  SELECT(NORETENTIONLIMIT)

LIST DUMPVOLUME -
  SELECT(DUMPCLASS(class))

LIST DUMPVOLUME(volser)
```

The first form of the request obtains information about all the dump volumes in the system. The next six forms of the request obtain information about all dump volumes in the system that meet certain conditions. The eighth form obtains information about all dump volumes in a group of volumes made from the same DASD volume from one dump operation. The variable *volser* can be any of the dump volumes in the group.

You can determine the data sets that were on a particular volume when the volume was dumped. The commands to do this are:

```
LIST ML1(volser) BCDS ALLDUMPS

LIST PRIMARY(volser) BCDS ALLDUMPS

LIST DUMPVOLUME(volser) -
  CONTENTS
```

The first two commands give lists of the volumes to which the source volume has been dumped and the date of the dump. The third command lists the contents of the source volume when it has been dumped. You can specify any one of the dump volumes in the set of dump volumes for *volser* for a particular date.

If a volume is processed by both full-volume dump and incremental backup, DFSMShsm maintains VTOC copy data sets for both operations. To determine the most up-to-date copy, issue the following command:

```
LIST PRIMARYVOLUME(volser) BCDS
```

If the full-volume dump is more recent than the backup, use the three preceding dump-related commands to obtain the latest VTOC copy. If the backup was more recent than the dump, use the LIST command for BACKUPCONTENTS as indicated in "Obtaining DFSMShsm-Managed Volume Information" on page 523.

Obtaining DFSMShsm Host Serialization Information

As DFSMShsm processes volumes and data sets, it obtains control of certain control data set records. While DFSMShsm in one z/OS image has control of a control data set record, other hosts in the same or different z/OS image cannot use that control data set record. To determine which control data set records that a particular DFSMShsm host has control of, issue the following command:

LIST HOST(*hostid*)

If one DFSMShsm host in a multiple DFSMShsm-host environment fails and you want it to relinquish control of all CDS records, issue the following command:

LIST HOST(*hostid*) RESET

Obtaining Lists of Backup and Dump Parameters and Backup Volsers

Note: In the following cases, full backup tapes are not listed after a FIXCDS REFRESH function has been performed.

To obtain a list of all backup and dump parameters and all nonfull tape backup volumes and DASD backup volumes, issue the following command:

QUERY BACKUP(ALL)

To obtain a list of all backup parameters and all nonfull daily tape backup volumes and DASD daily backup volumes, issue the following command:

QUERY BACKUP(DAILY)

To obtain a list of all backup parameters, all nonfull daily tape backup volumes, and DASD daily backup volumes assigned to a particular day in the backup cycle, issue the following command:

QUERY BACKUP(DAILY(*day*))

To obtain a list of all backup parameters and all nonfull spill tape backup volumes and DASD spill backup volumes, issue the following command:

QUERY BACKUP(SPILL)

To obtain a list of all backup parameters and all nonfull tape unassigned backup volumes and DASD unassigned backup volumes, issue the following command:

QUERY BACKUP(UNASSIGNED)

Obtaining Information about Migration and DFSMShsm-Managed Volumes

To obtain information about migration and DFSMShsm-managed volumes, you can select the following types of information:

- All migration volumes
- ML1 volumes
- All ML2 volumes
- ML2 DASD volumes

- ML2 tape volumes
- All migration and DFSMShsm-managed volumes
- A particular migration or DFSMShsm-managed volume

You also can select whether the information is to be taken from the backup control data set, the migration control data set, or both, and whether to select only empty volumes.

Again, because of the many possible forms for this request, only a sample is presented.

To request information about all migration volumes from both the migration control data set and the backup control data set, issue the following command:

```
LIST MIGRATIONVOLUME BOTH
```

To request information about all ML1 volumes from the backup control data set, issue the following command:

```
LIST MIGRATIONLEVEL1 BACKUPCONTROLDATASET
```

To request information about empty ML2 volumes from the migration control data set, issue the following command. You can use this command to obtain a list of tapes that can be inserted in the cartridge loader:

```
LIST MIGRATIONLEVEL2(TAPE) -
  MIGRATIONCONTROLDATASET SELECT(EMPTY)
```

To request information about ML2 DASD volumes from both control data sets, issue the following command:

```
LIST MIGRATIONLEVEL2(DASD) BOTH
```

To obtain information about ML2 tape volumes from the migration control data set, issue the following command:

```
LIST MIGRATIONLEVEL2(TAPE) MIGRATIONCONTROLDATASET
```

To obtain information about all DFSMShsm-managed and migration volumes from both control data sets, issue the following command:

```
LIST VOLUME BOTH
```

To obtain information about a particular volume from the backup control data set, issue the following command:

```
LIST VOLUME(volser) BACKUPCONTROLDATASET
```

Obtaining DFSMShsm-Managed Volume Information

You can also obtain information about the contents of a particular level 0 volume when the latest or next-to-latest incremental backup was performed. The information can be obtained from the backup control data set, the migration control data set, or both. For information about the incremental backup, you can select information about multiple-volume or VSAM data sets.

To obtain information about the contents of a level 0 volume when the latest incremental backup was performed, issue the following command:

LIST PRIMARYVOLUME(*volser*) BCDS BACKUPCONTENTS(0)

If a volume is processed by both full-volume dump and incremental backup, DFSMShsm maintains VTOC copy data sets for both operations. To determine the most up-to-date copy, issue the following command:

LIST PRIMARYVOLUME(*volser*) BCDS

If the backup was more recent than the full-volume dump, use the preceding backup-related command to obtain the latest VTOC copy. If the dump was more recent than the backup, use the LIST command for DUMPCONTENTS as indicated on page 521.

To obtain information about a DFSMShsm-managed volume that is contained in the migration control data set, issue the following command:

LIST PRIMARYVOLUME(*volser*) MIGRATIONCONTROLDATASET

To obtain information about the non-VSAM multiple-volume data sets that were on a level 0 volume at the next-to-latest incremental backup, issue the following command:

LIST PRIMARYVOLUME(*volser*) BCDS BACKUPCONTENTS(1) -
SELECT(MULTIPLEVOLUME)

To obtain information about the VSAM data sets that were on a level 0 volume at the latest incremental backup, issue the following command:

LIST PRIMARYVOLUME(*volser*) BCDS BACKUPCONTENTS(0) -
SELECT(VSAM)

Obtaining Tape Volume Information

You can obtain information about all tape volumes or specific tape volumes. In addition, you can select the volumes with specific characteristics.

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and where you *do not* want a list of data set information for the volume, issue the following command:

LIST TAPETABLEOFCONTENTS

To obtain a list of tape volume information from an OCDS, where you specify the volume serial number and you *do not* want a list of data set information for the volume, issue the following command:

LIST TAPETABLEOFCONTENTS(*volser*) NODATASETINFORMATION

To obtain a list of tape volume information from an OCDS, where you specify the volume serial number and you *do* want a list of data set information for the volume, issue the following command:

LIST TAPETABLEOFCONTENTS(*volser*) DATASETINFORMATION

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of volume entries for all ML1 volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(MIGRATIONLEVEL1)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of volume entries for all ML2 volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(MIGRATIONLEVEL2)
```

To obtain a list of ML2 partial tape volumes that are currently associated for output by specific migration or recycle tasks, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(ASSOCIATED)
```

To obtain a list of ML2 partial tape volumes that are available for an initial allocation, and are not currently associated for output by any migration or recycle task, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(NOTASSOCIATED)
```

To obtain a list of all ML2 volumes taken away from another function because RECALL needed a data set on the volume, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(RECALLTAKEAWAY)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of backup volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(BACKUP)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of both migration and backup volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(BOTH)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of tape volumes marked full that do not have alternate tape volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(NOALTERNATEVOLUME)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of tape volumes that have failed recycle, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(FAILEDRECYCLE)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of tape volumes that have a mismatch between the OCDS TTOC record and the actual tape media, issue the following command:

LIST TAPETABLEOFCONTENTS SELECT(FAILEDCREATE)

To obtain a list of tape volume information showing volumes that are marked as disaster alternate volumes, issue the following command:

LIST TAPETABLEOFCONTENTS SELECT(DISASTERALTERNATEVOLUMES)

Obtaining User Authorization Information

To determine the level of DFSMShsm authorization of a particular user ID, issue the following command:

LIST USER(*userid*)

Determining Which Tape Dump Volumes Are Available for Output during Dump Processing

To determine which tape volumes are available for volume dump processing, issue the following command:

LIST DUMPVOLUME SELECT(AVAILABLE)

This command lists each dump volume that is available for use for dump output.

To find out what tape volumes are available for a particular dump class for volume dump processing, issue the following command:

LIST DUMPVOLUME SELECT(AVAILABLE DUMPCLASS(*class*))

Obtaining Information about the DFSMShsm Address Space

You can use the DFSMShsm QUERY command to display the status of DFSMShsm parameters, statistics, and pending requests. The output of the QUERY command is messages returned to your terminal and written in the DFSMShsm log.

To obtain a list of parameters that apply to aggregate backup and recovery, issue the following command:

QUERY ABARS

To obtain a list about what functions are running in DFSMShsm, issue the following command:

QUERY ACTIVE

This list includes the following types of information:

- The status of the data set backup function
- The status of variables that affect the placement and selection of requests on the common recall queue
- The processing status of FRBACKUP and FRRECOV COPYPOOL commands

To obtain a list of volumes that are defined for aggregate recovery, issue the following command:

QUERY ARPOOL

To display the progress of DFSMShsm automatic functions, issue the following command:

QUERY AUTOPROGRESS

To obtain a list of backup and dump parameters, including new data set backup parameters, issue the following command:

QUERY BACKUP

To obtain a list of all backup and dump parameters and all backup volumes, issue the following command:

QUERY BACKUP(ALL)

To obtain a list of all backup parameters and all daily backup volumes, issue the following command:

QUERY BACKUP(DAILY)

To obtain a list of all backup parameters and daily backup volumes assigned to a particular day in the backup cycle, issue the following command:

QUERY BACKUP(DAILY(*day*))

To obtain a list of all spill parameters and all spill backup volumes, issue the following command:

QUERY BACKUP(SPILL)

To obtain a list of backup parameters and all unassigned backup volumes, issue the following command:

QUERY BACKUP(UNASSIGNED)

To obtain a list of parameters from the SETSYS command that are governing the backup of the control data sets, issue the following command:

QUERY CDSVERSIONBACKUP

To determine how full the DFSMShsm coupling facility structures are, issue the following command:

QUERY COMMONQUEUE

To determine all outstanding recall requests in the CRQ, issue the following command:

QUERY COMMONQUEUE(RECALL)

To display space-allocated-and-used statistics for the control data sets, issue the following command:

QUERY CONTROLDATASETS

To determine if any of the volumes defined to a copy pool are in an active FlashCopy releationship, issue the following command:

QUERY COPYPOOL(*cpname***)**

For FlashCopy version 2, the source volume may be in more than one relationship. This command can also be used to determine if another backup version can be created when using FlashCopy version 1, or if a recover can be performed. This command works only on FlashCopy relationships.

To display the common service area limits that were specified with the SETSYS command, issue the following command:

QUERY CSALIMITS

To display a list of requests associated with a specific data set, issue the following command:

QUERY DATASETNAME(*dsname***)**

Note: This command returns information only for those requests that were initiated by the host on which the QUERY DATASETNAME command is issued.

To display information for active DFSMShsm address spaces in a single z/OS image, issue the following command:

QUERY IMAGE

To display the key ranges and the associated ML2 volumes, issue the following command:

QUERY MIGRATIONLEVEL2

Note: This command also displays the volume serial numbers of the tape ML2 volumes that are currently selected as the target volumes for data set and volume migration and secondary space management.

To display a list of the recall pools, issue the following command:

QUERY POOL

To display a list of requests associated with a specific request number, issue the following command:

QUERY REQUEST(*reqnum*)

Note: This command returns information only for those requests that were initiated by the host on which the QUERY REQUEST command is issued.

To display a list of data set levels that are prevented from migrating by the SETMIG command, issue the following command:

QUERY RETAIN

To display the parameters that you set with the SETSYS command, issue the following command.

QUERY SETSYS

To display the space statistics on specified primary and ML1 volumes, issue the following command:

QUERY SPACE(*volser1,volser2, . . .,volsern*)

Note: The QUERY SPACE command is not valid for SMS-managed volumes.

To display the parameters you specified in the PROC statement of the DFSMShsm startup procedure, issue the following command:

QUERY STARTUP

To display selected DFSMShsm statistics that have accumulated during the day, issue the following command:

QUERY STATISTICS

To display the TRAP requests you have established, issue the following command:

QUERY TRAPS

To display a list of requests associated with a specific user ID, issue the following command:

QUERY USER(*userid*)

Note: This command returns information only for those requests that were initiated by the host on which the QUERY USER command is issued.

To display the current volume pools, issue the following command:

QUERY VOLUMEPOOL

To display requests for DFSMShsm services that are waiting to be processed, including information on the number of requests waiting in the common recall queue, issue the following command:

QUERY WAITING

Note: This command returns information only for those requests that were initiated by the host on which the QUERY WAITING command is issued.

Determining Which Tape Volumes Are Available for Output

The QUERY command is also used to determine tape volumes that are to be requested for backup and migration if volumes are available for selection.

Determining Which Tape Backup Volumes Are Available for Output

To find out which tape backup volumes are available for volume backup processing, follow this procedure:

1. Issue a QUERY BACKUP command to determine the current day in the backup cycle.
2. Issue a QUERY BACKUP(DAILY(*x*)) to find out what tapes are assigned to day *x* in the backup cycle. Replace *x* with the current day in the backup cycle.
3. Look at the volume serial numbers of the daily backup volumes assigned to the current day in the backup cycle. The first tape daily backup volume shown with an 'A' for available in the ARC0164I message is the first volume that DFSMShsm considers to allocate when it selects a tape backup volume.
4. To determine which tape backup volumes are empty and available for subsequent volume selection (if you specified SELECTVOLUME(SPECIFIC) on the SETSYS command), issue the following command:

   LIST BACKUPVOLUME SELECT(EMPTY)

You should be aware that additional tapes can be made available by performing the RECYCLE, EXPIREBV, or DELVOL commands, and by creating new backup tapes, which causes existing backup tapes to be invalidated.

Determining Which Tape Migration Level 2 Volumes Are Available for Output

To find out which tape ML2 volumes are available for migration processing, follow this procedure:

1. Issue a QUERY MIGRATIONLEVEL2 command.
2. Look at the volume serial numbers of the ML2 volumes. The ARC0227I message shows the volsers of the tapes currently associated with the respective data set, volume, and secondary space management tasks. These are the tapes DFSMShsm attempts to allocate when the tasks start. If you want to know what other ML2 tapes exist and might be used in place of these, do a LIST TTOC SELECT(NOTASSOCIATED).
3. Issue one of the following commands to determine which tape ML2 volumes are empty and available for subsequent volume selection (if you specified SELECTVOLUME(SPECIFIC) on the SETSYS command):

   LIST MIGRATIONLEVEL2(TAPE) SELECT(EMPTY)

   LIST TTOC SELECT(ML2 EMPTY)

Obtaining Statistical Information about DFSMShsm Operations

You can use the REPORT command to obtain statistics about DFSMShsm's operations. You can report the statistics on either a daily basis for all volumes or a volume basis for specific volumes. You can restrict the scope and format of the statistics reported to the backup, deletion of data sets, migration, recall, recovery, recycle, and spill functions. You can restrict the beginning date of the statistics or the dates between which the statistics are reported. You can also restrict the report to a summary of the statistics.

To report daily statistics about data sets that are backed up and to send the report to a DASD data set, issue the following command.

```
REPORT DAILY FUNCTION(BACKUP) OUTDATASET(dsname)
```

To obtain daily statistics about all functions, issue the following command:

```
REPORT DAILY FUNCTION
```

The report is sent to the SYSOUT class specified with the SETSYS command.

To obtain volume statistics for all volumes and to send the report to SYSOUT class A, issue the following command. This report also includes the total number of times that a fast replication backup was successfully created for a volume and the number of failed attempts for the volume. Recover statistics that are associated with the fast replication operation are also reported.

```
REPORT VOLUME FUNCTION SYSOUT(A)
```

To obtain volume statistics for one volume about migration between two dates and delete the statistics records after the report is printed, issue the following command:

```
REPORT VOLUME(volser FUNCTION(MIGRATION) -
  FROMDATE(date) TODATE(date) DELETE
```

The complete syntax of the report command is given in *z/OS DFSMShsm Storage Administration Reference*.

After you have formatted a report, you can free the space allocated for the statistics that made up the report. To clean up unneeded report statistics, do one of the following:

- Issue the DELETE parameter of the REPORT command. DFSMShsm deletes the statistics records (used as input for the report) after the REPORT command finishes its processing. For example, if you allocate one additional cylinder for the accumulation of statistics and control records for one week, issue the DELETE parameter of the REPORT command once a week.
- Specify how often DFSMShsm deletes MCDS data set records and statistics records with the MIGRATIONCLEANUPDAYS(*recalldays statdays*) parameter of the SETSYS command. The cleanup of these records occurs as a part of secondary space management.

Using the Problem Determination Aid to Trace DFSMShsm Processing

To help you gather information about DFSMShsm processing, DFSMShsm provides the problem determination aid (PDA). This aid can be used for both primary and secondary address spaces. It gathers diagnostic information at specific internal module points, records the data in a wraparound table in storage, and writes the data to data sets on DASD. You use the diagnostic information to provide data to your service representative should a problem develop with DFSMShsm. The purpose of the PDA is to gather sufficient information during DFSMShsm execution to enable subsequent analysis to pinpoint module flow and resource usage related to any given problem.

Note: If the DFSMShsm PDOX data set is on a non-SMS volume that will be backed up, use the ALTERDS *datasetname* VERSIONS(0) command to prevent DFSMShsm from backing it up.

If the PDOX data set is on an SMS volume that will be backed up, assign it to an SMS management class that does not allow backup copies.

Specifying to DFSMShsm to Use the PDA

In order for the PDA to trace DFSMShsm processing, it must not be disabled in the ARCCMD*xx* member referred to in the startup member. If you disable it at that time, no space is set aside for the circular file in storage, and the DASD data sets are not opened. The PDA is controlled by the following command:

```
SETSYS PDA
```

The default (if you do not specify the SETSYS command with the PDA parameter in the ARCCMD*xx* member) is PDA(ON).

If you do not want to use the PDA, add the following command to the ARCCMD*xx* member:

```
SETSYS PDA(NONE)
```

If you specify PDA(NONE), no space is set aside for the circular file in storage, and the DASD data sets are not opened.

After the PDA is started, you can turn it off and on during DFSMShsm operation. To turn the PDA off, issue the following command:

```
SETSYS PDA(OFF)
```

The PDA stops recording the diagnostic data, but the circular file space remains in storage, and the DASD data sets remain open.

To resume use of the PDA, issue the following command:

```
SETSYS PDA(ON)
```

Using the TRAP Command to Obtain Information about Conditions in the System When an Error Occurs

The TRAP command allows you to determine conditions in the system if an error occurs. You can test for all error codes in all modules, for a specific error code in all modules, for all error codes in a specific module, or for a specific error code in a specific module. When an error occurs, you can have DFSMShsm abnormally end, log the error, or take a SNAP dump. You can control whether the abnormal end or SNAP dump occurs always, never, or once.

Note: TRAP is not supported in the secondary address space.

To trap all error codes in all modules and cause an abnormal end for every error, issue the following command:

```
TRAP ALL ABEND(ALWAYS)
```

To trap all error codes in all modules and cause an abnormal end once, issue the following command:

TRAP ALL ABEND(ONCE)

To trap a specific error code in a specific module and log the error, issue the following command:

TRAP *module errorcode* LOG

When the error indicated by *errorcode* occurs in the module indicated by *module*, an entry is written in the DFSMShsm log.

To cause a SNAP dump every time a specific error code occurs in a specific module, issue the following command:

TRAP *module errorcode* SNAP(ALWAYS)

Related reading: For additional information about the TRAP command, see the *z/OS DFSMShsm Diagnosis Guide and Reference*.

Using DEBUG to Determine the Results of DFSMShsm Processing without Moving Data Sets

When you first set up DFSMShsm or when you make major changes to the data set selection parameters you have established for it, you may want to know which data sets are selected during volume space management, backup and backup expiration before you allow it to move any data sets. To allow this freedom to test, DFSMShsm provides a DEBUG mode of operation. To run in DEBUG mode, put the following command in the ARCCMDxx member for the DFSMShsm host on which you want to perform the testing:

SETSYS DEBUG

The DEBUG parameter of the SETSYS command permits DFSMShsm to exercise all of its volume backup, dump, and migration functions; but instead of moving or deleting data sets, it merely writes messages to the activity logs to indicate what data set movement or deletion would have occurred. For example, if you are already using DFSMShsm and want to test the effects of the EXPIREBV command, you could issue the SETSYS DEBUG command followed by the EXPIREBV command. DFSMShsm would then write messages to the activity logs, indicating which of your backup versions would have been deleted by the EXPIREBV command.

Issuing the ABACKUP and ARECOVER commands while SETSYS DEBUG is in effect causes the commands to be processed as if the DISPLAY parameter were in effect.

If DFSMShsm is stopped when it is running in DEBUG mode, any command requests that have not been processed are lost and are not processed when DFSMShsm is restarted. See “Stopping DFSMShsm” on page 542 for a discussion of restarting processing when DFSMShsm is restarted.

The FREQUENCY parameter of the DEFINE DUMPCLASS command is calculated from the last date that the level 0 volume was dumped while DFSMShsm was not

in DEBUG mode. This date is not updated when DFSMShsm is running in DEBUG mode because it would prevent volume dumps from processing on schedule when NODEBUG is specified. In DEBUG mode, if the date last dumped is zeros, the FREQUENCY value is not calculated, the level 0 volume is eligible to be dumped, and the date last dumped is not updated. To cancel the effect of the DEBUG parameter of the SETSYS command, you can issue the following command:

```
SETSYS NODEBUG
```

However, if the DEBUG parameter is specified in the ARCCMDxx member, you may find it more convenient just to remove the SETSYS DEBUG command from the ARCCMDxx member.

Chapter 15. DFSMShsm Tape Libraries

A system-managed tape library is a named collection of storage groups, and tape subsystems and their associated tape cartridges.

Automated Tape Library

An automated tape library (ATL) allows you to use system-managed storage for mountable tape cartridges. Mount requests for these cartridges are handled by a robotic tape cartridge handler. An ATL provides management facilities for tape volumes, but does not manage the data sets contained on those tape volumes.

Related reading: For additional information about the automated tape library, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Manual Tape Library

A manual tape library (MTL) also allows you to use system-managed storage for mountable tape cartridges. Like an ATL, it provides management facilities for tape volumes, but does not manage the data sets contained on those tape volumes. Unlike an ATL, mount requests are directed to a human operator.

Related reading: For additional information about a manual tape library, see the *z/OS DFSMShsm Implementation and Customization Guide*.

© Copyright IBM Corp. 1984, 2004

Chapter 16. Using DFSMShsm with DFSMSrmm

The DFSMS/MVS component 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.

DFSMSrmm supports any of the naming conventions DFSMShsm uses for its tape data sets, and all DFSMShsm options—with the exception of password security on tape volumes—are supported.

DFSMSrmm lets you use a system scratch tape pool or a DFSMShsm-managed tape pool. DFSMSrmm manages the movement of disaster recovery and vital record data for both pool types. It also provides a way to retain DFSMShsm tapes until the contents are invalid.

Recommendation: Use the system scratch tape pool.

You must let DFSMShsm decide whether a tape volume contains valid data and whether it should return to the scratch pool or DFSMShsm-managed tape pool.

You can manage the retention of valid tape volumes in two ways:

- Use DFSMSrmm vital record specifications to retain tape volumes until DFSMShsm finishes with them.
- 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.

You should use vital record specifications because you can avoid expiration dates, operator message handling, and the need to reinitialize tape volumes before you can reuse them. However, if you are using a DFSMShsm-managed tape pool, 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.

Related reading: For additional information about the DFSMSrmm EDGTVEXT interface, see the *z/OS DFSMSrmm Implementation and Customization Guide*.

Retaining DFSMShsm Tape Volumes by Expiration Date

If you use the system scratch tape pool for DFSMShsm tape volumes, you need a way to manage tape volumes protected with expiration dates. In this situation, DFSMSrmm lets you automate the responses to expiration date protection messages for system scratch pool tape volumes.

© Copyright IBM Corp. 1984, 2004

Related reading: For additional information about using the system scratch tape pool for DFSMShsm tape volumes, see the *z/OS DFSMSrmm Implementation and Customization Guide*.

Retaining DFSMShsm Tape Volumes by DFSMSrmm Vital Record Processing

DFSMSrmm vital record processing uses either data set names or volume serial numbers to determine how long to keep tapes. This retention period overrides any expiration date or retention period previously specified for the volume. In addition, you can use vital record specifications to identify volumes that should be moved out of the library for safe keeping.

Related reading: For additional information about DFSMSrmm vital record processing, see the *z/OS DFSMSrmm Implementation and Customization Guide*.

Tape Security When Using DFSMShsm and DFSMSrmm

The way in which you use the DFSMShsm option DFSMShsm TAPESECURITY may conflict with the way DFSMSrmm protects volumes.

Related reading: For additional information about protecting volumes, see the *z/OS DFSMSrmm Implementation and Customization Guide*.

Recommendations

You should perform the following steps when running DFSMShsm with DFSMSrmm:

- Run DFSMShsm with a system scratch pool that DFSMSrmm manages.
- Use RACF tape security.
- Use retention period and AUTOREUSE for automatic dumps.
- Set the following DFSMShsm parameters. For the system options, specify:

```
SETSYS  SELECTVOLUME(SCRATCH) -
                TAPEDELETION(SCRATCH) -
                TAPESECURITY(RACF) -
                PARTIALTAPE(REUSE)
```

For automatic dump definitions, specify:

```
DEFINE  DUMPCLASS(class -
        AUTOREUSE -
        RETENTIONPERIOD(days))
```

Chapter 17. Operating Procedures for DFSMShsm

The computing system operator is expected to perform certain operations for DFSMShsm. This chapter describes those operations. You might want to copy this chapter to provide your system operators with directions for performing the operations.

The following operations are described here:
- Partitioning DFSMShsm functions
- Starting DFSMShsm
- Stopping DFSMShsm
- Preventing DFSMShsm functions from running
- Allowing previously prevented DFSMShsm functions to run
- Canceling DFSMShsm queued requests
- Printing DFSMShsm log information
- Restarting DFSMShsm after an abnormal end
- Running automatic functions concurrently

Partitioning DFSMShsm Functions

This section describes the capabilities and functions of DFSMShsm and the uses for DFSMShsm control data sets. When you map these functions against the storage management needs of your installation, you might want to allocate certain subsets of functions to specific DFSMShsm hosts. In this context, a host is a DFSMShsm address space that performs automatic functions, such as migration. Secondary address spaces that perform aggregate backups and recovery (ABARS) belong to the main host and are not themselves considered hosts.

One z/OS image can have more than one DFSMShsm host. In a sysplex, multiple z/OS images can each have one or more hosts, all sharing the same set of control data sets (an HSMplex).

There are two possible modes of host activity that you can specify at startup time for any host: HOSTMODE=MAIN (the default) and HOSTMODE=AUX.

HOSTMODE=MAIN This (main) DFSMShsm host
- Processes implicit requests, such as recalls, or deletes migrated data sets, from user address spaces
- Processes explicit commands from TSO, such as HSENDCMD and HBACKDS, as well as requests from batch jobs, such as ARCINBAK
- Manages ABARS secondary address spaces
- Allows MODIFY commands from a console or from an HSM Monitor/Tuner (HMT)
- Can run automatic backup, dump, and space management

HOSTMODE=AUX This (auxiliary) DFSMShsm host
- Allows MODIFY commands from a console or from an HSM Monitor/Tuner (HMT)

© Copyright IBM Corp. 1984, 2004

- Can run automatic backup, dump, or space management

In addition, another keyword PRIMARY identifies the primary host in an HSMplex.

Rule: Within one z/OS image, all started hosts must be running DFSMShsm V2R10, or a later release.

Notes:

1. Up to 39 DFSMShsm hosts can be started concurrently in an HSMplex.
2. Within one z/OS image, any started host running on V2R10 or subsequent releases can share data with any host running on prior releases in the same HSMplex, but from a different z/OS image.
3. An HSMplex can consist of both single and multiple DFSMShsm-host environments.

There are several advantages to starting more than one DFSMShsm host in an z/OS image:

- Multiple DFSMShsm address spaces mean less work per address space and less contention between functions, because each SYSZTIOT resource serializes only functions in its address space.
- Multiple DFSMShsm address spaces mean that each address space that is doing some part of DFSMShsm's work can have an appropriate MVS dispatching priority for that type of work.
- Multiple DFSMShsm address spaces provide a larger number of tasks that perform any given DFSMShsm function; for example, migration.
- DFSMShsm functions that operate in more than one address space allow more MIPs that are allocated to DFSMShsm functions.

Within a single z/OS image, you can have one of the following combinations, all at OS/390 V2R10 or later, and all sharing the same set of control data sets:

- One main host
- One main host and one or more auxiliary hosts

Note: Because you can start up or shut down hosts in any order, another permitted (but normally temporary) combination is one or more auxiliary hosts without a main host.

Related reading: For additional information about the HOSTMODE and PRIMARY keywords, see the discussion of DFSMShsm multiple host environments in the *z/OS DFSMShsm Implementation and Customization Guide*.

Starting DFSMShsm

Before you can start DFSMShsm, you must have a DFSMShsm start procedure.

DFSMSHSM is used as the start procedure name. You can replace DFSMSHSM with any identification you designate for your computing system.

You can start DFSMShsm in either of two ways:

- The first time you start DFSMShsm after it is installed, you can load your system control program at initial program load. In response to the request:

IEA101A SPECIFY SYSTEM PARAMETERS

you reply with:

R 0,CLPA,CMD=nn

where CMD=*nn* identifies the startup member in the PARMLIB data set for COMMND*nn*, which contains a START DFSMShsm command. You add the START command for DFSMShsm to your parameter library member, COMMND*nn*, in the following manner:

COM='S DFSMSHSM,*parameters*'

- You can issue the START DFSMShsm command from the system console after the system control program is loaded at initial program load. You do not have to load the system control program at initial program load to restart DFSMShsm. The command to start DFSMShsm is:

  S DFSMSHSM,*parameters*

DFSMShsm only runs as a started task; you cannot start DFSMShsm as a batch job. If you plan to run more than one DFSMShsm host in an z/OS image, you must issue the START command once for each host. At that time, you must specify a unique host identifier in the startup procedure, on the START command itself, or in an ARCSTR*xx* PARMLIB member that is referred to by the startup procedure or by the command.

Rule: To enable a multiple address space environment for DFSMShsm, you must specify a new keyword HOSTMODE in the DFSMShsm start procedure.

If you start more than one DFSMShsm host in an image, the following conditions must exist:

- All the hosts must share the same set of control data sets; that is, they must be in the same HSMplex.
- All the hosts, therefore, share the same journal.
- Each host has its own log, activity logs, and PDA data sets.

To assist in problem analysis, you may want to incorporate the host-identifier characters into the names of the procedures that are used to start those hosts.

When DFSMShsm gains control after a START command has been issued, a message (ARC0001I) is issued to notify the operator that DFSMShsm is starting up. After the control data sets are opened and all major subtasks are attached, another message (ARC0008I) is issued to notify the operator that initialization is successful and that DFSMShsm is ready to accept commands.

Messages appear at the system console for each DFSMShsm command processed from the parameter library member if the monitoring of startup messages is in effect when the command is processed. These messages are only informational but could be useful if any problems occur. You start monitoring startup messages by specifying one of the following:

- COM='S DFSMSHSM,STARTUP=YES'
- SETSYS MONITOR(STARTUP)

You stop monitoring startup messages by specifying one of the following:

- COM='S DFSMSHSM,STARTUP=NO'
- SETSYS MONITOR(NOSTARTUP)

You can include the SETSYS command in the parameter library member as many times as you want. This allows you to control which commands issue messages to the system console when DFSMShsm is starting.

You can install DFSMShsm on multiple z/OS images of a sysplex or in multiple hosts in one z/OS image (or both). However, be sure that the HOST parameter of the startup procedure or the HOST parameter that is specified with the START command is unique for each instance of DFSMShsm that shares the same set of control data sets.

Note: You cannot start the DFSMShsm ABARS secondary address space from the operator's console by using an MVS START command. If you attempt this, DFSMShsm fails the command.

Related reading: For additional information about creating a start procedure, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Stopping DFSMShsm

To stop DFSMShsm, issue the following MODIFY command:

F DFSMSHSM,STOP

F DFSMSHSM,STOP without additional keywords causes no promotion of a secondary host.

Note: DFSMSHSM is the started procedure name. You can replace it with any identification you designate for your computing system.

You can also issue the MVS STOP DFSMSHSM command. When DFSMShsm is stopped, the normal job-log information is spooled to the printer along with the DFSMShsm log and activity logs that DFSMShsm maintains. Either one of the STOP commands causes an orderly shutdown of the DFSMShsm address space and an orderly closing of the DFSMShsm control data sets.

F DFSMSHSM,STOP PROMOTE makes the host eligible to have its unique functions taken over by a promotable host. You can issue a STOP PROMOTE command on a primary host, or a host that runs secondary space management, in sysplex environment with secondary host promotion enabled. This causes an orderly shutdown, and allows another host to take over the unique functions of the stopped primary host, or the secondary space management functions of the stopped host running secondary space management.

F DFSMSHSM,STOP DUMP issues a dump and makes the host eligible to have its unique functions taken over by a promotable host.

In a sysplex, a manual shutdown of the primary or secondary space management host, without using the PROMOTE or DUMP parameters, does not automatically cause promotion of another host, because it is assumed that there is a reason for the shutdown, such as for routine maintenance or because all systems in the sysplex are being shut down.

An orderly shutdown means that each attached subtask ends and all DFSMShsm secondary address spaces must complete processing. Because many DFSMShsm commands and automatic functions are potentially long-running, DFSMShsm periodically checks whether shutdown has been requested, to prevent a delay in the shutdown process. If DEFERMOUNT is used, or the correct tape was not mounted during end volume processing, and the timer was set by the MOUNTWAITTIME parameter of the SETSYS command, the timer expires and checks for shutdown. Volume functions check for shutdown before processing the next data set. After all subtasks have been detached and all secondary address spaces have completed processing, DFSMShsm issues a message (ARC0002I) to notify the operator that shutdown is complete and returns control to the system.

When DFSMShsm is shut down and restarted but MVS has not been reloaded at initial program load, user and HSENDCMD commands that did not complete are restarted as part of the DFSMShsm initialization process. Commands entered from the system console that have not been processed are lost, and any request that exceeds the extended common service area (ECSA) limit per address space is also lost.

Related reading: For additional information about the secondary host promotion function, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Shutdown Processing on the Common Recall Queue

When a host shuts down in a common recall queue (CRQ) environment, WAIT recall requests on the CRQ fail with RC=65, but new NOWAIT requests replace the failed requests. The new requests have the same priority as the original and can be processed by any eligible host that is connected to the CRQ. NOWAIT requests remain available in the CRQ.

Preventing DFSMShsm Functions from Running

You can prevent all or part of DFSMShsm's functions from running without stopping DFSMShsm. You can also interrupt all or part of the functions after they have begun running. For those functions that process on a volume basis (backup, dump, migration, and recover), you can choose whether to interrupt the function at the end of the data set currently being processed or at the end of the volume currently being processed. You use the HOLD command to prevent or interrupt DFSMShsm processing.

To prevent or interrupt all DFSMShsm processing, issue the following command:

```
HOLD ALL
```

Note: HOLD ALL does not control the hold level of the common recall queue (CRQ) placement and selection functions. To place a hold on the CRQ, specify HOLD COMMONQUEUE, with its optional subparameters.

To queue all aggregate backup processing and prevent any new commands from being processed, issue the following command:

```
HOLD ABACKUP
```

To queue all aggregate recovery processing and prevent any new commands from being processed, issue the following command:

```
HOLD ARECOVER
```

To interrupt AUDIT command processing after it finishes the current data set, issue the following command:

```
HOLD AUDIT
```

To interrupt or prevent all backup processing, issue the following command:

```
HOLD BACKUP
```

Backup is held by this command at the end of the data set currently being processed.To hold automatic backup processing but allow command backup processing, issue the following command:

```
HOLD BACKUP(AUTO)
```

Automatic backup is stopped by this command at the end of the data set currently being processed, but command backup processing continues and is allowed to start new data sets or volumes.

To interrupt or prevent backup processing at the end of the volume currently being processed, issue the following command:

```
HOLD BACKUP ENDOFVOLUME
```

To specify that a host should neither place requests on nor select requests from the common recall queue, issue the following command. After you issue HOLD COMMONQUEUE, any requests already in the CRQ remain available for processing by hosts that are connected to the CRQ. If no CRQ hosts are available, and the request is a WAIT, it fails. The originating host processes all recall requests that are received after you issue HOLD COMMONQUEUE.

```
HOLD COMMONQUEUE or HOLD COMMONQUEUE(RECALL)
```

To specify that the host should not place any additional requests onto the CRQ, issue the following command. After you issue this command, the host places all new recall requests in the local queue and continues to select requests to process from the CRQ. The host selects the highest priority request available in the local and common recall queues. When the highest priority request in the local queue and the highest priority request in the CRQ match, the host selects the request from the local queue. If you specify this command on all hosts, then all hosts place new recall requests in the local queue, but continue to select requests to process from the CRQ.

```
HOLD COMMONQUEUE(RECALL(PLACEMENT))
```

To specify that a host should continue to place but not select recall requests in the CRQ, issue the following command. After you issue this command, if a host places a request in the local queue because the CRQ is unavailable, the host processes

that request. If you specify this command on all hosts, all WAIT requests in the CRQ fail because no host is available to process them.

```
HOLD COMMONQUEUE(RECALL(SELECTION))
```

To prevent or interrupt dump processing, issue the following command. Dump processing stops at the end of the volume currently being processed.

```
HOLD DUMP
```

To prevent or interrupt automatic dump processing, issue the following command. Automatic dump processing stops at the end of the volume currently being processed. Processing initiated by DUMP commands is allowed to start.

```
HOLD DUMP(AUTO)
```

To specify no data movement, issue the following command.

Note: The SETSYS EMERGENCY command does not affect the placement of recall requests onto the common recall queue by the host with the command in effect, as other active hosts can process those requests.

```
SETSYS EMERGENCY
```

To interrupt dump processing at the end of the current unit of work being processed (usually a number of tracks or a cylinder), issue the following command:

```
HOLD DUMP ENDOFDATASET
```

To interrupt expiration of backup data sets by the EXPIREBV command, issue the following command. Expiration of backup versions ends when the data set that is being processed expires.

```
HOLD EXPIREBV
```

To prevent fast replication from processing any new FRBACKUP commands, issue the following command. No new fast replication backup commands are selected for processing.

```
HOLD FRBACKUP
```

To prevent fast replication from processing any new FRRECOV commands, issue the following command. No new fast replication recover commands are selected for processing.

```
HOLD FRRECOV
```

To interrupt or prevent processing by the LIST command, issue the following command. LIST command processing stops immediately. This command can be

useful if a list command is returning a large amount of data to your terminal and you want to interrupt the process. In such a case, the operator may be required to issue the HOLD command.

```
HOLD LIST
```

You can prevent DFSMShsm from making entries in the DFSMShsm log. Even though the log entries are prevented, DFSMShsm continues to make entries in the activity logs. To prevent entries into the DFSMShsm log, issue the following command:

```
HOLD LOG
```

To interrupt or prevent all migration processing, issue the following command. All migration stops at the end of the data set currently being processed.

```
HOLD MIGRATION
```

To interrupt or prevent only automatic volume space management and automatic secondary space management, issue the following command. Automatic volume space management and automatic secondary space management stops at the end of the data set currently being processed, but migration caused by commands is allowed to start.

```
HOLD MIGRATION(AUTO)
```

To interrupt migration at the end of the volume currently being processed, issue the following command:

```
HOLD MIGRATION ENDOFVOLUME
```

To prevent all recall processing, including no more processing from an already mounted migration tape, issue the following command.

Note: In a common recall queue (CRQ) environment, with the HOLD RECALL command in effect, the host is not eligible to select requests from the CRQ.

```
HOLD RECALL
```

To prevent all recall processing from tape volumes (except those from already mounted migration tapes) but allow recall processing from DASD volumes, issue the following command.

Note: In a CRQ environment, with the HOLD RECALL(TAPE) command in effect, the host is not eligible to select requests requiring tape from the CRQ.

```
HOLD RECALL(TAPE)
```

To prevent recall processing from tape volumes when the RECALL command is issued by a TSO user but to allow all other recall processing, issue the following command.

Note: In a CRQ environment, with the HOLD RECALL (TAPE(TSO)) command in effect, the host is not eligible to select requests requiring tape and TSO from the CRQ.

```
HOLD RECALL(TAPE(TSO))
```

To prevent all recover processing, including no more processing from an already mounted backup tape, issue the following command:

```
HOLD RECOVER
```

To prevent recover processing for data sets on tape backup (except those from already mounted backup tapes) and dump volumes but to allow recover processing for data sets on DASD backup volumes, issue the following command:

```
HOLD RECOVER(TAPEDATASET)
```

To interrupt or prevent recycle processing, issue the following command. Recycle processing stops at the end of the data set being processed.

```
HOLD RECYCLE
```

To interrupt or prevent REPORT command processing, issue the following command:

```
HOLD REPORT
```

Allowing Previously Prevented DFSMShsm Functions to Run

You can use the RELEASE command to allow DFSMShsm to run functions that you previously prevented from running, and to allow activity logs to be closed and produced as output.

To allow all held DFSMShsm functions except logging to run, issue the following command. This command does not restart writing in the DFSMShsm log and does not cause the activity logs to be closed and processed.

Note: RELEASE ALL does not control the releasing of holds on the common recall queue. You must instead specify RELEASE COMMONQUEUE, with its optional subparameters.

```
RELEASE ALL
```

To allow the ABACKUP command to be processed, issue the following command. Any queued ABACKUP commands are allowed to process.

```
RELEASE ABACKUP
```

To allow the ARECOVER command to be processed, issue the following command. Any queued ARECOVER commands are allowed to process.

```
RELEASE ARECOVER
```

To allow the AUDIT command to be processed, issue the following command. Any queued AUDIT commands are allowed to process. Any AUDIT command that was interrupted by a HOLD command does not restart.

```
RELEASE AUDIT
```

To release all held backup functions, issue the following command. All backup processing is allowed to start. If the time is still within the backup start window, automatic backup starts processing any functions that have not already been processed. Any volumes that were not already fully processed are candidates for processing if less than 24 hours has passed since automatic backup was last started from the beginning and if the time is still within the backup window.

```
RELEASE BACKUP
```

To allow automatic backup processing that was prevented by a HOLD BACKUP(AUTO) command, issue the following command. This command does not release automatic backup processing if automatic backup is prevented by a HOLD BACKUP command that prevents any backup processing. If the time is still within the backup start window, automatic backup starts processing any functions that had not been processed when backup was held or interrupted. Any volumes that were not already fully processed are candidates for processing if less than 24 hours has passed since automatic backup was last started from the beginning, and if the time is still within the backup window.

Note: If automatic backup is not scheduled to run on a particular day, you can use the BACKUP parameter of the RELEASE command to move backup versions (and backup migrated data sets). Automatic backup is not scheduled to run if you have specified SETSYS AUTOBACKUPSTART(0) or if you have specified N for this day in the backup cycle.

```
RELEASE BACKUP(AUTO)
```

To allow processing of data set backups to both tape and ML1 DASD that were prevented by a HOLD BACKUP(DSCOMMAND) command, issue the following command.

Note: Use of the BACKUP(DSCOMMAND) parameter allows data set backup by command processing. This command will not affect automatic backup processing.

```
RELEASE BACKUP(DSCOMMAND)
```

To allow processing of data set backups that are directed to ML1 DASD that were prevented by a HOLD BACKUP(DSCOMMAND(DASD)) command, issue the following command:

```
RELEASE BACKUP(DSCOMMAND(DASD))
```

To allow processing of data set backups that were directed to tape that were prevented by a HOLD BACKUP(DSCOMMAND(TAPE)) command, issue the following command:

```
RELEASE BACKUP(DSCOMMAND(TAPE))
```

To allow processing of data set backups that were directed to DASD that were prevented by a HOLD BACKUP(AUTO DSCOMMAND(DASD)) command, issue the following command:

```
RELEASE BACKUP(AUTO DSCOMMAND(DASD))
```

To allow processing of data set backups that were directed to tape if they were prevented by a HOLD BACKUP(AUTO DSCOMMAND(TAPE)) command, issue the following command.

Note: If you release a particular category, you will release the hold on that category's subcategory, as well. However, you cannot release a subcategory if the initial HOLD function occurred at a higher level. In contrast, you can release the higher level and then hold the complementary subcategory.

```
RELEASE BACKUP(AUTO DSCOMMAND(TAPE))
```

To release a common recall queue hold at the COMMONQUEUE level, holds at the RECALL level, and holds at the PLACEMENT and SELECTION level, issue the following command.

Note: When RELEASE COMMONQUEUE is issued, all unselected requests on the local queue move to the CRQ.

```
RELEASE COMMONQUEUE
```

If there is not a CRQ hold at the COMMONQUEUE level, to release a hold at the COMMONQUEUE(RECALL) level and a hold at the PLACEMENT and SELECTION level, issue the following command.

Note: When RELEASE COMMONQUEUE(RECALL) is issued, all unselected requests on the local queue move to the CRQ.

```
RELEASE COMMONQUEUE(RECALL)
```

To release a CRQ hold at the PLACEMENT level only if a hold is not at a higher level also, issue the following command.

Note: When RELEASE COMMONQUEUE(RECALL(PLACEMENT)) is issued, all unselected requests on the local queue move to the CRQ.

```
RELEASE COMMONQUEUE(RECALL(PLACEMENT))
```

To release a CRQ hold at the SELECTION level only if a hold is not at a higher level also, issue the following command:

```
RELEASE COMMONQUEUE(RECALL(SELECTION))
```

To allow dump processing to run, issue the following command. Any queued dump commands are allowed to process. If the time is still within the dump window, automatic dump starts processing any functions that had not been completed when the dump function was held.

```
RELEASE DUMP
```

To allow automatic dump processing to run if automatic dump was held by a HOLD DUMP(AUTO) command, issue the following command. If the time is still within the dump start window, automatic dump starts processing any functions that had not been completed when the dump function was held.

```
RELEASE DUMP(AUTO)
```

To allow expiration of backup versions if EXPIREBV was held by a HOLD EXPIREBV command, issue the following command. New EXPIREBV commands are processed. Any EXPIREBV processing that was interrupted by the HOLD EXPIREBV command is not restarted.

```
RELEASE EXPIREBV
```

To release the fast replication process by the FRBACKUP command, issue the following command.

```
RELEASE FRBACKUP
```

To release the fast replication process by the FRRECOV command, issue the following command.

```
RELEASE FRRECOV
```

To close the activity log data sets and process them, issue the following command. This command causes one of two actions.

- If the activity logs are directed to a SYSOUT data set, the activity logs are closed and printed.
- If the activity logs are directed to DASD data sets, the current DASD data sets are closed and new activity log data sets are allocated.

```
RELEASE HARDCOPY
```

To allow LIST command processing, issue the following command. Any queued LIST commands are allowed to process. Any list operation that was in process when list was held is not restarted.

```
RELEASE LIST
```

To allow entries to the DFSMShsm log after the entries have been prevented, issue the following command:

```
RELEASE LOG
```

To allow all migration processing, issue the following command. Any queued MIGRATION commands are allowed to begin processing. If the time is still within the automatic primary space management or automatic secondary space management processing windows, any automatic primary space management or automatic secondary space management functions that were not completed when migration was held begin processing.

```
RELEASE MIGRATION
```

To allow automatic primary space management or automatic secondary space management processing if they were prevented by a HOLD MIGRATION(AUTO) command, issue the following command. If the time is still within the automatic primary space management or automatic secondary space management processing windows, any automatic primary space management or automatic secondary space management functions that were not completed when automatic primary space management or automatic secondary space management was held are started.

```
RELEASE MIGRATION(AUTO)
```

When DFSMShsm is migrating data sets to both DASD and tape devices, an error condition can end migration to either of the device types without ending migration to the other. After the error is corrected, you can restart migration to the device that had the error by issuing one of the following commands:

```
RELEASE MIGRATION(AUTO)
RELEASE MIGRATION
```

To allow all recall processing, issue the following command. Any queued RECALL commands are allowed to process. In a common recall queue (CRQ) environment, RELEASE RECALL makes the host eligible to select requests from the CRQ.

```
RELEASE RECALL
```

To allow recall from tape data sets if recall was held by a HOLD RECALL(TAPE) command, issue the following command:

```
RELEASE RECALL(TAPE)
```

To allow recall from tape data sets that were requested by TSO users (if the recall was held by a HOLD RECALL(TAPE(TSO)) command, issue the following command:

```
RELEASE RECALL(TAPE(TSO))
```

To allow all recover processing, issue the following command. Any queued RECOVER commands are allowed to process.

```
RELEASE RECOVER
```

To allow data sets to be recovered from tape volumes if recover processing was held by a HOLD RECOVER(TAPEDATASET) command, issue the following command:

```
RELEASE RECOVER(TAPEDATASET)
```

To allow recycle processing, issue the following command. Any queued recycle requests begin processing. Any volumes that were being recycled when recycle was held are not restarted.

```
RELEASE RECYCLE
```

To allow REPORT commands to be processed, issue the following command. Any queued report requests begin processing. Any report that was in process when reporting was held is not restarted.

Note: If DFSMShsm functions are being held because of a journal-inhibited condition and that condition cannot be resolved at that time, the following DFSMShsm commands can be issued to bypass journaling.

- SETSYS NOJOURNAL
- RELEASE XXX (for whatever function you want to release)
- SETSYS JOURNAL (after a successful CONTROLDATASET BACKUP and the journal reset is completed)

Attention: If journaling is bypassed, you cannot perform a CONTROLDATASET RECOVERY for that period of time the journal was inhibited. The journal updates are lost for that period of time.

```
RELEASE REPORT
```

Canceling DFSMShsm Queued Requests

DFSMShsm lets you cancel queued requests for:

- An individual command
- A user ID
- A data set name

Rule: If you are using the common recall queue, you must submit the cancel request on the same host that originated the recall request that you want to cancel.

Use the CANCEL command to identify the commands for cancelation. The only commands that can be canceled are those commands that have not started. Identification of the commands to cancel can be specified by the DFSMShsm request number, user ID, or data set name. Use the QUERY command to find the request numbers, user IDs, or data set names. Use the QUERY ACTIVE command to determine which DFSMShsm requests are currently being processed.

Note: The HSM Monitor/Tuner also has the capability to cancel certain active DFSMShsm tasks.

Exceptions:

1. Depending on whether you choose to back up data sets that are in use (see "Specifying How DFSMShsm Should Respond to a Data Set in Use" on page 175

175), DFSMShsm during a BACKVOL can schedule multiple requests to retry backing up (BACKDS) data sets found to be open for update when backup was initially attempted. All requests scheduled during one BACKVOL are assigned the *same* request number originally assigned to the BACKVOL request itself. Canceling the request number cancels *all* requests. To cancel only one request, use the DATASETNAME parameter.

2. Data set recoveries scheduled during a volume recovery (or during APPLYINCREMENTAL processing following a volume restore) are assigned the *same* request number originally assigned to the volume recovery or volume restore request. Canceling that request number cancels *all* those requests. To cancel only one request, use the DATASETNAME parameter.

A CANCEL command with the USERID or DATASETNAME parameters cancels all existing nonexecuting requests having the specified user ID or data set name.

The following command cancels a single request:

CANCEL REQUEST(*num*)

The following command cancels the requests of a particular user:

CANCEL USERID(*userid*)

The following command cancels the requests for a particular data set:

CANCEL DATASETNAME(*dsn*)

Note: Depending on whether you choose to back up data sets that are in use (see "Specifying How DFSMShsm Should Respond to a Data Set in Use" on page 175), DFSMShsm during autobackup can schedule multiple requests to retry backing up (BACKDS) data sets found to be open for update when backup was initially attempted. Each request scheduled during autobackup has a request number of *zero*, which cannot be specified with the REQUEST parameter. You should use the DATASETNAME parameter to cancel one of these requests.

The CANCEL and QUERY commands cannot be canceled.

Printing DFSMShsm Log Information

The information in the DFSMShsm log can be used to determine the status of the data sets and volumes in DFSMShsm. Figure 71 on page 554 shows the DFSMShsm log printing environment.

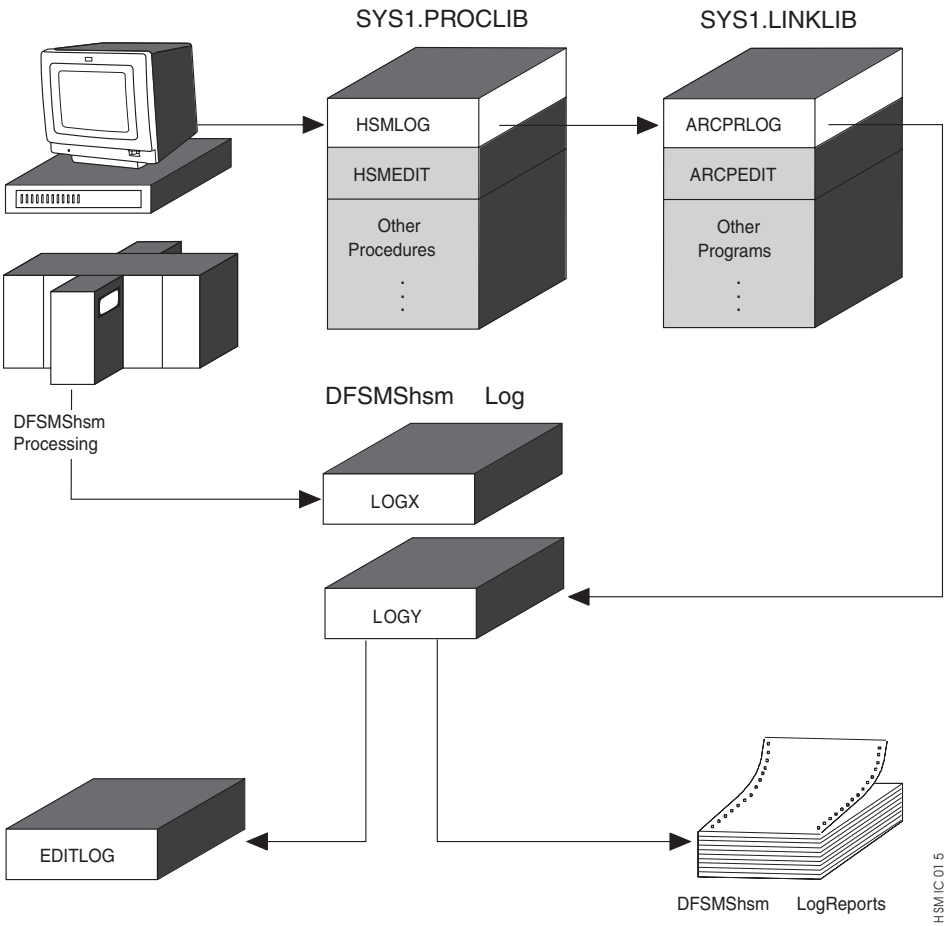


Figure 71. DFSMShsm Log and Edit Log Overview

The DFSMShsm log is actually two data sets, called LOGX and LOGY. LOGX is used to write the log information into, while LOGY is being processed for the information it contains. When LOGX becomes full, the data set receives a D37-04 ABEND indicating that it ran out of space. DFSMShsm automatically switches the LOGX and LOGY data sets: LOGX becomes LOGY, and LOGY becomes LOGX. This switch allows access to the data recently recorded in the log.

The following messages are displayed on the operator's console:

```
IEC031I D37-04,IFG0554T,DFHSM,DFHSM,ARCLOGX,
        HSMLIB,DFHSM,HSMLOGX
ARC0020I ******************************
ARC0024I DFSMShsm LOGGING INHIBITED DUE TO EOV ERROR
ARC0024I (CONT.) ON LOG
ARC0020I ******************************
ARC0020I ******************************
ARC0027I  DFSMShsm LOG SWITCHED
ARC0020I ******************************
```

You can also force the logs to be switched by entering the SWAPLOG command. The following message are displayed:

```
ARC0020I ******************************
ARC0027I  DFSMShsm LOG SWITCHED
ARC0020I ******************************
```

In either case, it is now possible to use the ARCPRLOG program and the ARCPEDIT programs to process LOGY. You use the ARCPRLOG program to do an initial formatting of the information in the log and to print the log.

If the LOGY data set or the edit-log data set is RACF-protected, you must take special steps before calling the HSMLOG procedure. The HSMLOG procedure processes the ARCPRLOG program to read, format, and print information in the LOGY data set and to insert selected log records into the edit-log data set. The HSMLOG procedure normally resides in SYS1.SAMPLIB in member ARCSTRST, and the ARCPRLOG program normally resides in the LINKLIB data set. After processing the ARCPRLOG program, the HSMLOG procedure empties the LOGY data set. If the HSMLOG procedure is called as a started task, the RACF started-procedures table (ICHRIN03) must be updated to include the procedure name (HSMLOG) and the RACF user ID to be associated with the procedure when it is started. If the HSMLOG procedure is called as part of a job, you must code USER=*userid* on the JCL JOB statement. In either case, the *userid* specified in the RACF started-procedures table or the *userid* specified on the JOB statement must have sufficient RACF access authority to read and update the LOGY data set.

If the edit-log data set is RACF-protected, you must take special steps before calling the HSMEDIT procedure. The HSMEDIT procedure processes the ARCPEDIT program to format and print the information in the edit-log data set. The HSMEDIT procedure normally resides in SYS1.SAMPLIB in member ARCSTRST, and the ARCPEDIT program normally resides in the LINKLIB data set. If the HSMEDIT procedure is called as a started task, the RACF started-procedures table must be updated to include the procedure name (HSMEDIT) and the RACF user ID to be associated with the procedure when it is started. If the HSMEDIT procedure is called as part of a job, you must code USER=*userid* on the OS/VS2 JCL JOB statement. In either case, the *userid* specified in the RACF started-procedures table or the *userid* specified on the JOB statement must have sufficient RACF access authority to read the edit-log data set.

The ARCPRLOG program selects the management work elements (MWEs) from all DFSMShsm commands and the statistics records for each completed DFSMShsm function for output to the ARCPEDIT program. You use the ARCPEDIT program to analyze, edit, and print the selected log records produced by the ARCPRLOG program.

The following example of JCL uses the ARCPRLOG and ARCPEDIT programs with the PROCLIB data set. Job HSMLOG initially analyzes, edits, and prints the information from the log and the LOGY data set.

```
//HSMLOG    JOB
//PRINTLOG  EXEC PGM=ARCPRLOG
//ARCPRINT  DD   SYSOUT=A
//ARCLOG    DD   DSN=DFHSM.HSMLOGY1,DISP=OLD
//ARCEDIT   DD   DSN=DFHSM.EDITLOG,DISP=OLD
//*
//EMPTYLOG  EXEC PGM=IEBGENER
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   DUMMY
//SYSUT2    DD   DSN=DFHSM.HSMLOGY1,DISP=OLD
//SYSUT1    DD   DUMMY,DCB=(DFHSM.HSMLOGY1)
//*
```

Notes:

1. If the HSMLOGY1 data set is empty, do not use the ARCPRLOG program. If you do, unpredictable results can occur.
2. DSN=DFHSM is only an example of the data set name. Specify whatever name you choose to name the data set.
3. Please note that the log data set used as input to the ARCPRLOG program should not be compressed. This is because the log data set is created with RECFM=F but is opened by ARCPRLOG for update with RECFM=U. This is not allowed for compressed data sets.

Job EDITLOG formats and prints selected log records produced by the HSMLOG job.

```
//EDITLOG   JOB
//EDITLOG   PROC UID=HSM
//EDITLOG   EXEC PGM=ARCPEDIT
//ARCPRINT  DD SYSOUT=A
//ARCLOG    DD DSN=&UID..EDITLOG,DISP=SHR
//          PEND
//*
```

Figure 72 on page 560, Figure 73 on page 561, and Figure 74 on page 562 show examples of the printed log output. The following is an alphabetical list of the information contained in the printed log output:

Field Name	Meaning
ABCC	System abnormal end completion code
ALLOC	Time of day allocated
APPLYINC	Indicates APPLYINCREMENTAL or not
ASCB	Pointer to the address space control block of the requester
AVGTIME	Average time for migration, per data set
BLKR	Number of 16K blocks read from a tape backup volume
BLKW	Number of 16K blocks written to a tape backup volume
BUFL	Length of buffer for the “control data set read” management work element

Field Name	Meaning
BUFU	Amount of buffer for the “control data set read” management work element
BYTR	Number of bytes read
BYTW	Number of bytes written
CCNT	Number of management work elements processed in the current request
CMDL	Length of the command
CMDO	Offset for TSO scan and parse
CMDT	Command text
COPY DSN	Data set name for the backup version
CPUSEC	Total job step time in milliseconds for the day
DARC	Dynamic allocation error return code
DATE	Date of the log record
DCOP FAIL	Number of requested dump copies that failed
DCOP REQ	Number of dump copies requested
DEVT	Device type as in the UCB
DFLGS	Dump restore flags
DLM	Date last migrated
DLU	Date last used
DSN	Data set name
DSORG	Data set organization
DUMP COPY REQUESTS	Number of dump copies requested
DUMPCLASS(ES)	DUMPCLASSes listed or none
DUMPVOL	Dump volume serial
END	Time of day ended
FAILURES	Number of dump copy failures
FFLGS	Subsequent migration flags
FLGS	Flag bytes
FLG2	More DFSMShsm flags
FLG3	More DFSMShsm flags
FRVOL	Volume from which the backup version was copied
FRVSN	Volume from which the data set migrated
FUNC	Function description
GEN	Backup version generation number requested
GROUP	Group name for RACF
GROUP	Group name for RACF
HOST	Host identifier of an DFSMShsm address space
ID	Log record code identification
JOBNM	Job name of the requesting job
KEY	DFSMShsm record key
LEN	Length of the management work element

Field Name	Meaning
MCNT	Number of management work elements in the request
MIG	Migration
MSGID	Message identification
MWE FUNC	Function code of the management work element
NABEND	Number of DFSMShsm subtask ABENDs for the day
NAME	Contents of the management work element
NDCOPFAIL	Number of dump copies that failed
NDCOPSSYS	Number of dump copies requested by the system
NDCOPSUSE	Number of dump copies requested by the user
NDSMIGFAIL	Number of failing migrate requests
NDSMIGL1	Number of migrates to ML1
NDSMIGL2	Number of migrates to ML2
NDSPW	New data set password for the recovered data set
NDSRECAFAIL	Recall failures
NDSRECAL1	Recalls from L1
NDSRECAL2	Recalls from L2
NDSRESTFAIL	Number of data set restores that failed
NDSRESTSREQ	Number of data set restores requested
NDUMPFAIL	Number of volume dumps that failed
NDUMPSSYS	Number of volume dumps requested by the system
NDUMPSUSE	Number of volume dumps requested by the user
NEWDSN	New data set name for the recovered data set
NGBRECA	Gigabytes read during recall
NGBYTMIG	Gigabytes read during migration
NI	Number of inserts in the message
NIPL	Number of DFSMShsm startups for day
NKBRECA	Kilobytes read during recall
NKBYTMIG	Kilobytes read during migration
NMBRECA	Megabytes read during recall
NMBYTMIG	Megabytes read during migration
NMWE	Number of DFSMShsm MWEs received
NSHUT	Number of DFSMShsm shutdowns for day
NTBRECA	Terabytes read during recall
NTBYTMIG	Terabytes read during migration
NVOLRESTFAIL	Number of volume restores that failed
NVOLRESTSREQ	Number of volume restores requested
PW	Data set password
RACFLG	RACF flags
RC	Return code from the request
REAS	Reason code for the errors
RECFM	Record format

Field Name	Meaning
RETPD(S)	Retention period list
RQN	Request number
SEQN	Sequence number of this management work element in a multiple request
START	Time of day started
TIMES: REQUEST RECEIVED, STARTED, ALLOCATED, ENDED	Time request was received. Time request was started. Time allocations were completed. Time request completed
TO	Message directed to
TOD	Time of day
TOVOL	Volume to which the backup version was copied
TOVSN	Volume to which the data set migrated
TRKR	Number of tracks read
TRKW	Number of tracks written
TSCR	Create time stamp (always blank)
TSLU	Last update time stamp (always blank)
TYPE	DFSMShsm record type
UID	User identification
VER	Backup version number requested
VSN	Volume serial number

Note: The TRKR and TRKW fields are approximate values for the number of tracks read and written and can vary, based on the blocking factors used.

```
LEN=0163  TOD=202630  DATE=95/06/07  ID=13  NAME=MWE COMPLETED
 MWE FUNC=09  TOD=00:00:00  FLGS=000044  UID=B342236   RQN=0015  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=011  CMDO=007  CMDT='SWAPLOG'
LEN=0182  TOD=202636  DATE=95/06/07  ID=04  NAME=ARCCTL RCVE MWE
 MWE FUNC=09  TOD=00:00:00  FLGS=200004  UID=B342236   RQN=0016  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=030  CMDO=000  CMDT='DEFINE DUMPCLASS(DCLASS01)'
LEN=0182  TOD=202636  DATE=95/06/07  ID=15  NAME=CATALOG CREATE OR UPDATE
 CAT LEN=0170  TYPE=?  TSLU=          TSCR=          KEY=|DCLASS01
     00000000 00000000 00160000 F3F4F8F0 40404040 40404040 40404040 40404040
     40404040 40404040 00000000 00000000 00000000 00000000 00000000 00000000
     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     00000000 00000000 00000054
LEN=0084  TOD=202639  DATE=95/06/07  ID=03  NAME=OUTPUT MSG TO OPER/USER
     MSGID=0216  TO=B342236   NI=3   ARC0216I DUMPCLASS DEFINITION MODIFIED, CLASS=DCLASS01 RC=0
LEN=0182  TOD=202639  DATE=95/06/07  ID=13  NAME=MWE COMPLETED
 MWE FUNC=09  TOD=00:00:00  FLGS=000044  UID=B342236   RQN=0016  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=030  CMDO=026  CMDT='DEFINE DUMPCLASS(DCLASS01)'
LEN=0182  TOD=202644  DATE=95/06/07  ID=04  NAME=ARCCTL RCVE MWE
 MWE FUNC=09  TOD=00:00:00  FLGS=200004  UID=B342236   RQN=0017  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=030  CMDO=000  CMDT='DEFINE DUMPCLASS(DCLASS02)'
LEN=0182  TOD=202644  DATE=95/06/07  ID=15  NAME=CATALOG CREATE OR UPDATE
 CAT LEN=0170  TYPE=?  TSLU=          TSCR=          KEY=|DCLASS02
     18000000 00000006 00000003 40404040 40404040 E3C8D9D6 E640C1E6 C1E84040
     40404040 40404040 000A0000 00000000 00000000 00000000 00000000 00000000
     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     00000000 00000000 00000054
LEN=0084  TOD=202646  DATE=95/06/07  ID=03  NAME=OUTPUT MSG TO OPER/USER
     MSGID=0216  TO=B342236   NI=3   ARC0216I DUMPCLASS DEFINITION MODIFIED, CLASS=DCLASS02 RC=0
LEN=0182  TOD=202646  DATE=95/06/07  ID=13  NAME=MWE COMPLETED
 MWE FUNC=09  TOD=00:00:00  FLGS=000044  UID=B342236   RQN=0017  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=030  CMDO=026  CMDT='DEFINE DUMPCLASS(DCLASS02)'
LEN=0273  TOD=202650  DATE=95/06/07  ID=04  NAME=ARCCTL RCVE MWE
 MWE FUNC=09  TOD=00:00:00  FLGS=200004  UID=B342236   RQN=0018  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=121  CMDO=000  CMDT='BACKVOL VOL(MIG215)            DUMP(DCLASS(BADDCL01,BADDCL02) RETPD(*,30)'
LEN=0273  TOD=202651  DATE=95/06/07  ID=13  NAME=MWE COMPLETED
 MWE FUNC=09  TOD=00:00:00  FLGS=000044  UID=B342236   RQN=0018  ASCB=F90100  RC=0000  REAS=0000
 ABCC=00000000
     CMDL=121  CMDO=117  CMDT='BACKVOL VOL(MIG215)            DUMP(DCLASS(BADDCL01,BADDCL02) RETPD(*,30)'
LEN=0110  TOD=202651  DATE=95/06/07  ID=03  NAME=OUTPUT MSG TO OPER/USER
     MSGID=0184  TO=**OPER**  NI=4   ARC0184I ERROR WHEN READING THE DFSMShsm CONTROL DATA SET W RECORD
FOR BADDCL01, RC=0004
LEN=0091  TOD=202652  DATE=95/06/07  ID=03  NAME=OUTPUT MSG TO OPER/USER
     MSGID=0643  TO=**OPER**  NI=2   ARC0643I BACKVOL COMMAND REJECTED, DUMP CLASS BADDCL01 NOT DEFINED
```

Figure 72. Example of a List from the ARCPRLOG Program

```
********************  TOD=202630  DATE=00095219  NAME=MWE COMPLETED   ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='SWAPLOG'

********************  TOD=202639  DATE=00095219  NAME=MWE COMPLETED   ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='DEFINE DUMPCLASS(DCLASS01)'

********************  TOD=202646  DATE=00095219  NAME=MWE COMPLETED   ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='DEFINE DUMPCLASS(DCLASS02)'

********************  TOD=202651  DATE=00095219  NAME=MWE COMPLETED   ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='BACKVOL VOL(MIG215)                DUMP(DCLASS(BADDCL01,BADDCL02) RETPD(*,30)'

********************  TOD=202654  DATE=00095219  NAME=VOLUME DUMP ENDED ********************
    FUNC=VOLUME DUMP              DSN=
    TOVOL=        FRVOL=MIG215  DEVT=00000000  DSORG=     RECFM=      START=202654  END=202654
    BATCH REQUEST    WAIT SPECIFIED
    USER REQUEST

********************  TOD=202658  DATE=00095219  NAME=MWE COMPLETED   ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='BACKVOL VOL(MIG215)                DUMP(DCLASS(DCLASS01,DCLASS02) RETPD(N,30)'

********************  TOD=202946  DATE=00095219  NAME=VOLUME DUMP ENDED ********************
    FUNC=VOLUME DUMP              DSN=
    TOVOL=        FRVOL=MIG215  DEVT=00000000  DSORG=     RECFM=      START=202659  END=202946
    BATCH REQUEST    WAIT SPECIFIED
    USER REQUEST

********************  TOD=202954  DATE=00095219  NAME=MWE COMPLETED  ******************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(DCLAS(DCLASS01))               UNIT(3390)'

********************  TOD=203000  DATE=00095219  NAME=MWE COMPLETED  ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(DVOL(DUTP1) AINC)              UNIT(3390)'

********************  TOD=203002  DATE=00095219  NAME=VOLUME RESTORE ENDED ********************
     FUNC=VOLUME OR DS RESTORE     DSN=
     TOVOL=MIG215  FRVOL=        DEVT=3010200E  DSORG=     RECFM=      START=203001  END=203002
     BATCH REQUEST    WAIT SPECIFIED
     USER REQUEST

********************  TOD=203004  DATE=00095219  NAME=MWE COMPLETED  ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(DGEN(1) AINC)                    UNIT(3390)'

********************  TOD=203006  DATE=00095219  NAME=VOLUME RESTORE ENDED  ********************
    FUNC=VOLUME OR DS RESTORE     DSN=
    TOVOL=MIG215  FRVOL=        DEVT=3010200E  DSORG=     RECFM=      START=203004  END=203006
    BATCH REQUEST    WAIT SPECIFIED
    USER REQUEST

********************  TOD=203007  DATE=00095219  NAME=MWE COMPLETED  ********************
MWE FUNC=09  FLGS=000044  UID=B342236
      CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(AINC) DATE(95/01/01)             UNIT(3390)'

********************  TOD=203009  DATE=00095219  NAME=VOLUME RESTORE ENDED ********************
    FUNC=VOLUME OR DS RESTORE     DSN=
    TOVOL=MIG215  FRVOL=        DEVT=3010200E  DSORG=     RECFM=      START=203008  END=203009
    BATCH REQUEST    WAIT SPECIFIED
    USER REQUEST
```

Figure 73. Example of a List from the ARCPEDIT Program

	OPERANDS THAT WERE SPECIFIED BY FUNCTION									
FUNCTION	DSNAME	WAIT	NOWAIT	VOLSER	UNITNAME	MLEVEL2	REPLACE	FROMVOL	DCOP REQ	DCOP FAIL
MIGRATE L0->L1	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
MIGRATE L1->L2	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
MIGRATE L0->L2	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RECALL L1->L0	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RECALL L2->L0	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
DELETE MDS	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
BACKUP DATA SET	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
SPILL BACKUP COPY	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RECOVER BACKUP COPY	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
RECYCLE DATA SET	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
DELETE BY AGE	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
RECYCLE MIG VOLUME	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
DUMP OF VOLUME	00000	00000	00003	00000	00000	00000	00000	00000	00009	00006
RESTORE VOLUME	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RESTORE DATA SET	00006	00000	00006	00000	00000	00000	00000	00000	00000	00000

Figure 74. Example of a Summary List from the ARCPEDIT Program

Restarting DFSMShsm after an Abnormal End

DFSMShsm automatically recovers after most abnormal terminations. However, if DFSMShsm could not recover from the abnormal end and the RESTART keyword was used in the PROC statement of the startup procedure, DFSMShsm will restart itself. The RESTART keyword allows you to specify a startup procedure—which can be the same or different from the original startup procedure—and to pass additional parameters to DFSMShsm. RESTART is only valid in an ESA environment.

Should an abnormal end occur that interrupts MVS processing, DFSMShsm can process any waiting requests as long as the extended common service area (ECSA) is not disturbed. When DFSMShsm is restarted, it will process any requests that are waiting in the ECSA.

Recovery from an abnormal end could possibly extend long enough to cause checking for the need to restart to occur outside the automatic processing window for migration, backup, or dump. Should such an incident occur and should you want the automatic function to restart, issue one of the following commands (as appropriate for the function that you want to start):

SETSYS PRIMARYSPMGMTSTART (*starttime,endingtime*)

SETSYS SECONDARYSPMGMTSTART (*starttime,endingtime*)

SETSYS AUTOBACKUPSTART (*starttime,latestarttime,quiescetime*)

SETSYS AUTODUMPSTART (*starttime,latestarttime,quiescetime*)

Leave *starttime* as it is for normal operations. Change *endingtime*, *latestarttime* and *quiescetime* to be later than the time that DFSMShsm is restarted. *Starttime* specifies the planned start time. *Endingtime* or *quiescetime* determine when DFSMShsm stops processing new volumes for the automatic function being processed. Because you do not change the *starttime* specification, DFSMShsm recognizes that you are not defining a new start window for the function. After the function has completed, you can issue a new SETSYS command to reset *endingtime, latestarttime,* and *quiescetime* to their former values.

If a DFSMShsm host fails in a sysplex environment where secondary host promotion is enabled, the unique functions of a primary host, or the secondary space management processing of any host are eligible to be taken over by a promoted host, when certain conditions are met.

If you are using the common recall queue (CRQ) when a host fails, all of the host's recall requests on the CRQ remain intact. They are eligible for processing by other active hosts, and the coupling facility notifies the remaining connected hosts of the failure. Also, requests that were in-process remain on the CRQ and are available for restart by other hosts.

Related reading: For additional information about secondary host promotion, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Running Automatic Functions Concurrently

Although the example system that has been used to explain the automatic functions of DFSMShsm used different times for automatic backup, dump, and migration, any or all of the functions can run concurrently, either in the same host or in different hosts.

When running the automatic functions concurrently, you must remember that the first thing that automatic backup does on the primary host is to optionally back up the control data sets. While the control data sets are being backed up, automatic dump, automatic volume space management and automatic secondary space management are delayed. Also, if automatic dump, automatic volume space management or automatic secondary space management are running when automatic backup begins to back up the control data sets, backup of the control data sets is delayed until migration and dump tasks complete the volumes they are processing and automatic secondary space management completes one of its level functions. If automatic backup starts first, none of the other processes can start until the control data sets are backed up.

Note: Functions such as recycle, tape copy, and audit perform similarly to automatic functions in regards to contention with CDS backup.

Example: In the example system, assume that DFSMShsm host 1 performs level functions and backup; DFSMShsm host 2 performs migration; and DFSMShsm host 3 performs dump. The commands provided are only those that would be particularly different from the examples that have been shown before.

First, the time when each of the DFSMShsm hosts begins processing must be defined. Issue the following commands:

For DFSMShsm host 1:

SETSYS AUTOBACKUPSTART(0001,0100,0200)

SETSYS SECONDARYSPMGMTSTART(0800,1600)

For DFSMShsm host 2:

SETSYS PRIMARYSPMGMTSTART(0030,0500)

For DFSMShsm host 3:

SETSYS AUTODUMPSTART(0030,0300,0500)

Automatic backup is started early in z/OS image 1 to allow it to back up the control data sets. After the control data sets are backed up, the other automatic functions are started.

For concurrent processing as it has been previously defined, the z/OS images that are to process each function must be defined to the storage groups. The MIGRATE SYSTEM/SYS GROUP NAME, BACKUP SYSTEM/SYS GROUP NAME, and DUMP SYSTEM/SYS GROUP NAME attributes are used in the storage groups to provide the definitions for SMS-managed volumes.

For example, the attributes in all storage groups would be set as follows:

```
BACKUP SYSTEM/SYS GROUP NAME  ===> PRC1

MIGRATE SYSTEM/SYS GROUP NAME ===> PRC2

DUMP SYSTEM/SYS GROUP NAME    ===> PRC3
```

Note: System group names can also be used for BACKUP SYSTEM/SYS GROUP NAME, MIGRATE SYSTEM/SYS GROUP NAME, and DUMP SYSTEM/SYS GROUP NAME.

Use the ADDVOL command to provide z/OS image definitions for non-SMS-managed volumes. For example, the following ADDVOL commands can be used to control the common volumes (GP0001, GP0002, GP0003).

For DFSMShsm host 1:

```
ADDVOL GP0001 PRIMARY(NOAUTOMIGRATION, AUTOBACKUP, -
  NOAUTODUMP)

ADDVOL GP0002 PRIMARY(NOAUTOMIGRATION, AUTOBACKUP, -
  NOAUTODUMP)

ADDVOL GP0003 PRIMARY(NOAUTOMIGRATION, AUTOBACKUP, -
  NOAUTODUMP)
```

For DFSMShsm host 2:

```
ADDVOL GP0001 PRIMARY(AUTOMIGRATION, NOAUTOBACKUP, -
  NOAUTODUMP)
```

ADDVOL GP0002 PRIMARY(AUTOMIGRATION, NOAUTOBACKUP, -
NOAUTODUMP)

ADDVOL GP0003 PRIMARY(AUTOMIGRATION, NOAUTOBACKUP, -
NOAUTODUMP)

For DFSMShsm host 3:

ADDVOL GP0001 PRIMARY(NOAUTOMIGRATION, NOAUTOBACKUP, -
AUTODUMP)

ADDVOL GP0002 PRIMARY(NOAUTOMIGRATION, NOAUTOBACKUP, -
AUTODUMP)

ADDVOL GP0003 PRIMARY(NOAUTOMIGRATION, NOAUTOBACKUP, -
AUTODUMP)

Chapter 18. Purposes of DFSMShsm Commands

This chapter presents a brief description of the commands you use to:

- Define DFSMShsm values and volumes
- Determine status
- Maintain and recover the control data sets and the DFSMShsm address space
- Maintain data sets and volumes
- Operate DFSMShsm

These commands are fully described in the *z/OS DFSMShsm Storage Administration Reference*.

Use the following commands to define the values and volumes for DFSMShsm:

Command	Purpose
ADDVOL	Adds a new volume to the set of volumes owned or managed by DFSMShsm and changes existing volume information defined with earlier ADDVOL commands.
AUTH	If the RACF FACILITY class is not active, DFSMShsm uses the AUTH command to protect storage administrator DFSMShsm commands.
DEFINE	Defines the structure of DASD level 2 volumes for DFSMShsm and defines a data set or volume pool related to non-SMS recalls. It also defines the automatic backup cycle, the secondary space management cycle, the dump cycle, dump classes, the volumes for aggregate recovery, and times for automatically demounting tapes currently in use by data set backup.
DELVOL	Removes a volume from the list of volumes managed by DFSMShsm and changes existing volume information.
SETSYS	Changes the values of various DFSMShsm parameters from their current values or defaults.

Use the following commands and parameters to determine status:

Command	Purpose
AUDIT	Detects and reports discrepancies about data set information in the computing system catalog, the aggregate backup and recovery ABR record, the DFSMShsm control data set records, and the VTOC of the DASD storage volume on which the data set resides. **Note:** This command also corrects discrepancies in the common recall queue.
LIST	Lists data set, user, volume, TTOC, and control data set record-serialization records from the DFSMShsm control data sets.
QUERY	Lists values of parameters specified with DFSMShsm commands, lists DFSMShsm statistics, or lists the status of pending requests.
REPORT	Generates reports based on the DFSMShsm statistics recorded in the MCDS.

© Copyright IBM Corp. 1984, 2004

Parameter	Purpose
MONITOR	This parameter of the SETSYS command lists the status of startup, space use, volume space management, backup, and recovery as these functions occur.
TAPELIST	This parameter of the RECYCLE command generates a list of tape volumes eligible for recycling.

Use the following commands to maintain and recover the control data sets and the DFSMShsm address space:

Note: The first four commands are only for diagnosis and repair. Refer to the *z/OS DFSMShsm Diagnosis Guide and Reference* for additional information about these commands.

Command	Purpose
DISPLAY	Displays the contents of locations within the DFSMShsm address space identified by an absolute address, a relative address, or a qualified address.
FIXCDS	Displays, modifies, and adds and deletes records in the MCDS, BCDS, and OCDS.
PATCH	Modifies storage within the DFSMShsm address space.
TRAP	Specifies whether DFSMShsm should write a log entry or cause a dump to be taken when the error specified with the TRAP command occurs.
UPDATEC	Combines the entries in the journal data set with a restored backup copy of one or all of the control data sets. See "Recovering Control Data Sets Using the Enhanced CDS Recovery Function" on page 463 for the recommended way to recover control data sets that run outside the DFSMShsm address space.

Use the following commands to directly manage data sets or volumes (as opposed to the automatic management functions of DFSMShsm):

Command	Purpose
ALTERDS	Changes the number of backup versions maintained and the frequency of creating backup versions for a non-SMS-managed data set.
ABACKUP	Backs up aggregate groups of user-defined data sets for recovery at another location.
ARECOVER	Recovers the data sets that were backed up by the ABACKUP command.
BACKDS	Creates a backup version of a specific data set.
BACKVOL	Requests that DFSMShsm back up eligible data sets on one or more volumes, in one or more storage groups, or on all primary volumes; or that it back up the control data sets. This command can also request that DFSMShsm invoke DFSMSdss to dump one or more volumes, or one or more storage groups.
BDELETE	Deletes all the backup versions or specific backup versions of a specific data set.
DELETE	Deletes a migrated data set without recalling the data set.
EXPIREBV	Deletes all eligible expired backup versions from all backup volumes.
FRBACKUP	Creates a fast replication backup version for each volume in a specified copy pool.

Command	Purpose
FRDELETE	Deletes one or more unneeded fast replication backup versions.
FREEVOL	Moves all migration copies meeting a specified age criteria between DFSMShsm migration or backup volumes.
FRRECOV	Uses fast replication to recover a single volume or a pool of volumes from the managed backup versions.
MIGRATE	Moves eligible data sets from DFSMShsm-managed volumes to migration volumes or from migration volumes to other migration volumes. It can also delete non-SMS-managed data sets from either DFSMShsm-managed volumes or migration volumes.
RECALL	Returns migrated data sets to DFSMShsm-managed volumes.
RECOVER	Obtains backed-up or dumped data sets and copies them to DFSMShsm-managed volumes.
RECYCLE	Consolidates onto fewer DFSMShsm-owned tapes valid data that is interspersed with invalid data on many DFSMShsm-owned tapes.
SETMIG	Changes the space management status of a specific data set or group of data sets on DFSMShsm-managed, non-SMS-managed volumes.
TAPECOPY	Makes copies of classes of DFSMShsm-owned tapes or of individual DFSMShsm-owned tapes and identifies the copies as alternate tapes to the original tapes.
TAPEREPL	Replaces references to original tapes with references to alternate tapes.

Use the following commands to operate DFSMShsm:

Command	Purpose
CANCEL	Cancels existing queued DFSMShsm requests for: • A specific request number • All requests for a particular user • All requests for a particular data set
HOLD	Prevents all or part of DFSMShsm's functions from running. Also causes the data set backup switchtapes action to occur immediately.
HSENDCMD	Issues DFSMShsm-authorized-user commands from a TSO terminal.
LOG	Enters information into the DFSMShsm log.
RELEASE	Releases for processing all or part of the DFSMShsm functions that have been held.
STOP	Causes an orderly shutdown of DFSMShsm.
SWAPLOG	Causes DFSMShsm to exchange its LOGX data set for its LOGY data set or exchange its PDOX data set for its PDOY data set.

Command Function Changes for SMS

In a non-SMS environment, DFSMShsm is controlled by commands and several parameters that describe the configuration and determine how and when DFSMShsm functions are performed. In an SMS environment, the storage management functions defined in the management class and the storage group control the SMS-managed data sets in place of the parameter or command definitions within DFSMShsm. The parameters formerly used on the SETSYS command and shown in Table 27 on page 570 do not apply to SMS-managed data sets.

Table 27. SETSYS Parameters Not Applicable to SMS-Managed Data Sets

DFSMShsm Parameter, Non-SMS	SMS Equivalent
DAYS	PRIMARY DAYS NON-USAGE
MIGRATIONLEVEL1DAYS	LEVEL 1 DAYS NON-USAGE
None	PARTIAL RELEASE
SETMIG (command)	COMMAND OR AUTO MIGRATE
None	ADMIN OR USER COMMAND BACKUP
None	# GDG ELEMENTS ON PRIMARY
None	ROLLED-OFF GDS ACTION
FREQUENCY	BACKUP FREQUENCY
None	GUARANTEED BACKUP FREQUENCY
VERSIONS	NUMBER OF BACKUP VERSIONS (DATA SET EXISTS)
None	NUMBER OF BACKUP VERSIONS (DATA SET DELETED)
None	RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED)
None	RETAIN DAYS EXTRA BACKUP VERSIONS
None	EXPIRE AFTER DAYS NON-USAGE
None	EXPIRE AFTER DATE/DAYS

The functions of certain DFSMShsm commands are assumed by management class and storage group attributes for SMS-managed data sets and volumes. Table 28 shows these commands and the SMS construct that assumes their functions.

Table 28. DFSMShsm Commands Not Used for SMS Storage Management

DFSMShsm Command, Non-SMS	SMS Class
DEFINE POOL	Storage group
DEFINE VOLUMEPOOL	Storage group
ADDVOL	Storage group
ALTERDS	Management class
SETMIG	Management class
SETMIG VOLUME	Storage group

Long-Running Commands

The AUDIT, EXPIREBV, LIST, RECYCLE, REPORT, and TAPECOPY commands are known as long-running commands. Long-running commands are limited to having only one of each such command being performed by DFSMShsm at any time.

Other commands can also run for a long time. If a command is running for a long time and is returning data to your terminal, you can be prevented from performing other operations at your terminal. An example might be a LIST or AUDIT command that is returning large amounts of data. To stop such a command from interfering with your use of your terminal, have the computer system operator issue a HOLD command for the function that is causing the interference. The operator can then issue a RELEASE command for the function, but the command

that was being performed does not restart. For example, if a LIST command returns data to your terminal and prevents you from using it, have the operator issue a HOLD LIST and RELEASE LIST command sequence. The list command that is causing you trouble does not restart.

Chapter 19. RACF Considerations

Throughout this document, the use of the Resource Access Control Facility (RACF) is recommended because DFSMShsm requests RACF authorization before it allows access to data sets by non-DFSMShsm-authorized users. RACF can also prevent access to DFSMShsm-owned data sets except by users with specific RACF authorization.

In addition, by utilizing FACILITY class profiles, RACF can control command-level access for end-users and storage administrators.

Beginning with RACF 1.9, you can use SECLABEL CLASS to control data set access. During initialization, DFSMShsm checks to see if SECLABEL CLASS is active. If SECLABEL CLASS or RACF is activated or deactivated after DFSMShsm startup, you must stop and restart DFSMShsm to reset the SECLABEL indicator.

System Authorization Facility

All RACF requests from DFSMShsm are made through the system authorization facility (SAF). SAF conditionally directs control to RACF (if present) or to a user-supplied processing routine. RACF is determined to be installed by inspection of control block fields. If a RACF-equivalent product sets these fields the same as RACF, DFSMShsm operates as if RACF were installed. Data management checks the user's authority during data set open when the user attempts to access the data.

Related reading:

- For additional information about FACILITY class profiles, see "Controlling Command Access with RACF FACILITY Class" on page 582 or the *z/OS DFSMShsm Implementation and Customization Guide*.
- For additional information about the system authorization facility, see the *z/OS MVS Programming: Assembler Services Guide*.

RACF Basics

To understand the relationship between RACF and DFSMShsm, you must understand the RACF concepts of RACF profiles, RACF indication, always-call support, and protect-all environment.

RACF Data Set Profiles

RACF controls data set access by identifying and verifying authorized users and by preventing unauthorized users from accessing data. RACF uses the information in a data set profile to determine whether a user is authorized to access the data set.

RACF has two types of RACF data set profiles: *generic* and *discrete*. The generic data set profile describes one or more data sets that have a similar name structure. For example, all data sets with the high-level name qualifier 'JJJ' can be described in a single generic data set profile. The discrete data set profile describes a specific data set on a particular volume.

© Copyright IBM Corp. 1984, 2004

Generic Profile

RACF TSO commands create and maintain generic data set profiles. Generic profiles are never dynamically created, copied, renamed, or deleted by the operating system or DFSMShsm. You must activate generic profile support with a RACF SETROPTS command before RACF can use generic profiles for authorization checking.

Generic profiles are designed to work with data sets that conform to a particular naming convention. A single access list then describes the access authorities to all those data sets.

Examples: The following are three generic profiles:

GRP12.LIB.*	Access list A
GRP12.*.CNTL	Access list B
GRP12.*	Access list C

When RACF searches a profile for data set GRP12.SRC.CNTL, it finds that the first profile does not match. The second profile does match; RACF verifies the user's authority with access list B. Note that the third profile also matches, but because it is less specific than the second, RACF does not use it.

In a generic profile definition, an * (asterisk) means any characters that are acceptable to the syntax of a data set name.

Generic profiles contain the following elements:

- Allow for security management by exception. In most cases, data sets that share the same naming conventions have the same security requirements. One profile can apply to many data sets with the same security requirements.
- Are more static than discrete profiles. They are not deleted when a data set is deleted.
- Make data set protection insensitive to DASD migration.
- Enhance performance as compared to discrete profiles. The most recently used generic profiles are kept in storage and are more often reused. Fewer updates to the RACF data set are needed.

Discrete Data Set Profiles

Discrete data set profiles provide a one-to-one relationship between a specific data set on a specific volume and its security authorization requirements. To maintain the one-to-one relationship, discrete profiles are:

- Optionally created when a DASD data set is allocated
- Renamed when the data set is renamed
- Deleted when the data set is scratched

RACF indication implies that a data set is protected by a discrete RACF data set profile.

Related reading: For additional information about defining RACF discrete and generic data set profiles, see the *z/OS Security Server RACF Security Administrator's Guide* and the *z/OS Security Server RACF Command Language Reference*.

RACF Indication

A RACF-indicated data set is one whose RACF indicator is on. For a non-VSAM data set, the RACF indicator is a bit in the volume table of contents (VTOC) of the

DASD volume on which the data set resides. For a VSAM data set, the RACF indicator is a bit in the catalog record. The indicator remains with the data set even if the data set is moved to another system. If the data set profile is not moved also or is somehow lost, a RACF security administrator must take some action before anyone can access the data set.

You can use the SETSYS RACFIND command to cause DFSMShsm to set the RACF indicator in the VTOC entry for migrated data sets, or for backup versions if the original data sets are RACF-indicated when they are migrated or backed up. Use the RACFIND parameter to RACF protect DFSMShsm-owned data sets when you use RACF discrete data set profiles.

Always-Call Support

A RACF always-call environment exists when all data accesses on behalf of non-DFSMShsm-authorized users result in the invocation of RACF for authorization checking, regardless of the setting of the RACF indicator in the VTOC entry or catalog record. Always-call is a system-wide convention for protection that is available to data sets that are allocated on DASD. With always-call support, resource managers in the operating system call RACF, regardless of the setting of the RACF indicator in the data set's DSCB. Some of the resource managers of the operating system are:

- OPEN/CLOSE/EOF
- DADSM ALLOCATE/EXTEND/RENAME/SCRATCH
- IEHMOVE

For details on the security facilities in your system, see your Security Administrator.

Without an always-call environment, generic profiles do not necessarily provide protection for the data sets they cover, because an unauthorized user might be able to access the data sets if the RACF indicator is not on. In this environment, use only discrete profiles. When the RACFIND parameter of the SETSYS command is in effect, DFSMShsm turns on the VTOC entry RACF indicator to prohibit access to the backup versions and migration copies of RACF-indicated data sets.

RACF Indicator with Always-Call

RACF's response to resource management authorization requests depends on the RACF indicator setting.

If the authorized requester (the caller) informs RACF that it believes the data set is protected by a discrete profile (by using the RACFIND=YES parameter), RACF first looks for a discrete profile. If none is found, RACF checks for a matching generic profile. If no generic profile is found, RACF rejects the request.

If the authorized requester (the caller) informs RACF that it believes the data set is not protected by a discrete profile (by using the RACFIND=NO parameter), RACF checks only for a generic profile.

A fundamental difference exists between RACF-indicated and non-RACF-indicated data sets. If no profile is found, RACF denies requests for RACF-indicated data sets but allows access for non-RACF-indicated data sets.

RACF Protect-All Environment

Beginning with RACF 1.7, installations have the option to fail access to data sets that are covered by neither a discrete nor a generic RACF profile. This

environment is specified with the RACF SETROPTS command and the PROTECTALL keyword. If this option is in effect, RACF returns a "not-authorized" return code when normal access checking or data set creation authorization checking is done. For the DFSMShsm user, it means that failure caused by a lack of profile does not appear to be different from failure caused by a lack of authorization.

DFSMShsm functions are affected as follows in a protect-all environment:

- Recall or deletion of migrated data sets
  - Automatic: DFSMShsm performs authority checking to ensure that the user initiating automatic recall is at least authorized to use the data set.
  - Command: Security checking is performed when the request originates with a non-DFSMShsm-authorized user. The request is failed if no profile covers the data set to be recalled or deleted. A generic profile can be created before requesting recall or deletion, or the ADDSD command can be used to simultaneously initiate an implicit recall and create a discrete profile.
- HALTERDS command: If the data set does not yet exist, a generic profile must be created, or the command fails when originated by a non-DFSMShsm-authorized user.
- HRECOVER command: If a backup profile exists, DFSMShsm creates a discrete profile. If no backup profile exists, create a discrete or generic profile before issuing the command, or the request fails.
- Remaining DFSMShsm commands (HBDELETE, HBACKDS, HMIGRATE): Create a discrete or generic profile before issuing the command, or the request fails.

RACF Environment for DFSMShsm

This discussion assumes the following environment:

- All data sets are cataloged in catalogs
- RACF security is based on protection with generic profiles
- RACF always-call support has been established

DFSMShsm and RACF Concepts

Migrated data sets and backup versions are stored in a DFSMShsm internal format. Data set names for migrated data sets and backup versions are different from the names of the corresponding data sets on the level 0 volume; thus the original RACF protection may not apply.

When DFSMShsm migrates a cataloged DASD data set, it scratches the data set on the level 0 volume. However, it does not delete the catalog entry; Instead DFSMShsm changes the original volume serial number to a dummy volume serial number named MIGRAT.

TSO users can explicitly recall (only with DFSMShsm) their data sets to a level 0 volume if they are authorized by RACF.

In addition to automatic DFSMShsm functions, TSO commands are available for users to selectively control DFSMShsm space and availability management activities for their data sets. These TSO commands can also be issued by batch jobs using the terminal monitor program (TMP). Your installation may require users to identify themselves to RACF with the USER and PASSWORD parameters on the JOB card.

When executing a DFSMShsm function started by a TSO command, DFSMShsm asks RACF for authorization for the user, specifying the original data set name on the level 0 volume. Thus, when DFSMShsm and TSO are used, backup and migration operations use the RACF-protection rules of their related data set on the level 0 volume. This occurs regardless of whether the migration or backup support is DASD or tape.

No access authorizations to the migrated or backed up data sets themselves are needed for users issuing DFSMShsm commands.

Migrated or backed up data sets should not be accessed as normal MVS data sets. RACF generic profiles should be set up to protect these data sets by their real names on DASD, and RACF tape protection must protect the corresponding tape volumes.

Only DFSMShsm should be authorized to use migrated and backed up data sets as MVS data sets. The RACF generic profiles for these data sets should contain an empty access list, and the default universal access authority (UACC) should be NONE. The DFSMShsm started task does not need to be on the access lists, because DFSMShsm sets itself up as a privileged user to RACF. It may be desirable to allow a few storage administrators on these access lists, to repair DASD or tape volumes. The authorization is automatically granted if these users have the RACF OPERATIONS attribute.

Failure to protect migrated and backed up data sets with generic profiles creates integrity exposures such as DFSMShsm having a record of a backup version or migration copy of a data set when the version or copy has been deleted.

RACF Authority Needed to Perform DFSMShsm Functions

DFSMShsm bypasses any security checking during automatic volume space management, automatic secondary space management, and availability management.

Undirected automatic recall is caused by reference from JCL, or under TSO, to a cataloged data set that is, in fact, migrated. In such a reference, the target volume is not specified. Once recall has occurred, standard RACF protection applies through OPEN. Table 29 shows the authority needed by TSO users to issue DFSMShsm commands.

Table 29. RACF Authorization Required for DFSMShsm Functions

DFSMShsm Function	RACF Resource Access Authority Required
Migrate a data set	UPDATE
Recall a data set	EXECUTE
Delete a migrated data set	ALTER
Back up a data set	UPDATE
Recover a backup version without specifying NEWNAME	ALTER
Recover a backup version and specify NEWNAME	READ to original data set; ALTER on the NEWNAME
Delete a backup version	ALTER
Change backup characteristics	ALTER
Aggregate backup	READ

Special considerations apply to the TSO user commands HBDELETE (to delete backup versions) and HALTERDS (to modify backup characteristics):

- HBDELETE: If the backed up data set does not exist and the data set was protected by a discrete profile, DFSMSdfp asked RACF to delete the profile when the data set was deleted. DFSMShsm knows that the backed up data set was RACF-indicated. DFSMShsm fails the request on this command if there is no backup profile or generic profile honoring access to the data set name, or if the requestor does not have ALTER authority on either the generic profile or the backup profile.

  If the backed up data set does not exist and if the data set was not RACF-indicated but was protected by a generic profile, DFSMShsm fails the request only if a generic profile that matches the original data set name does exist and the user does not have ALTER authority on that profile.
- HALTERDS: A user might set up backup characteristics for a data set that does not exist yet. As in the preceding item, DFSMShsm fails the request only if a generic profile that matches the original data set name exists and the requestor does not have ALTER authority on that profile. If the data set exists, DFSMShsm fails the request if no generic profile, discrete profile, or backup profile exists honoring access to the data set name or if the requestor does not have ALTER authority on the generic, the discrete, or the backup profile.

RACF Security Bypassing for DFSMShsm Commands

The previously discussed TSO commands can also be issued by using the operator console. DFSMShsm operator commands can be entered from the operator console or by a DFSMShsm-authorized user (a user who has received the DFSMShsm DATABASEAUTHORITY attribute) under TSO by using the HSENDCMD command. The command keywords of these DFSMShsm commands are identical to the corresponding TSO commands, except that the leading H of these commands is omitted. For example, for the TSO HDELETE command, the operator enters the DELETE command or the authorized TSO user enters the HSENDCMD DELETE command.

In general there is no RACF checking for DFSMShsm commands using the operator console. Therefore, the number of DFSMShsm-authorized users should be limited. An installation can use the ISPF-supplied SVC to run authorized, thus providing a way to issue requests to DFSMShsm. If DFSMShsm TSO commands are issued by a user who runs MVS-authorized, no RACF checking is done. If RACF FACILITY class is being used to protect aggregates, a user ID representing the console user must be defined and authorized to the aggregate group profiles before issuing the ABACKUP or ARECOVER commands from a console, or processing fails.

The RACF User ID for DFSMShsm

DFSMShsm must be able to bypass RACF during migration and backup of user data sets. You must, therefore, define DFSMShsm to RACF and give it the necessary level of authority. You do this by defining an entry in the RACF started-procedures module ICHRIN03, which contains the DFSMShsm startup procedure name and a user ID that you define to RACF by an ADDUSER command. Do not define this user ID with the automatic data set protection (ADSP) attribute.

The following is an example of the RACF command you can use to define DFSMShsm to RACF.

ADDUSER (*hsmid*) DFLTGRP (*grpname*)

In this example, *hsmid* is the user ID specified in the RACF started-procedure module ICHRIN03. The default group name is not important, but if it is not specified, RACF sets it to the current connect group of the user issuing the command.

Notes:

1. The user who issues the ADDUSER command must have the RACF SPECIAL attribute.
2. The UID parameter in the DFSMShsm startup procedure is not related to the RACF user ID.
3. If you are using remote sharing functions (RRSF) to propagate RACF commands to remote hosts, it is recommended that the RACF user ID for DFSMShsm be defined with the SPECIAL and OPERATIONS attributes on all recipient systems. An alternative approach for setting up for RRSF would be to mark DFHSM as TRUSTED in the RACF started procedures table (ICHRIN03) or STARTED class on all of the recipient systems.

RACF Started Task Table

Because a started task has a system-generated JOB card and cannot, therefore, supply USER, GROUP, and PASSWORD information, it must be identified to RACF by a different mechanism. SYS1.LPALIB contains a module, ICHRIN03, that contains the information that RACF requires for started tasks, and you must add the name of the DFSMShsm procedure (or procedures if there are more than one) to this module. You also need to add the name of the ABARS procedure to this module. In fact, this must be done for any started task that accesses DFSMShsm data sets (for example, a customized external writer to save the DFSMShsm log). The module is a nonexecutable table containing 32 bytes of character information for each entry. Each entry contains four 8-byte fields.

procname	*user ID*	*gpname*	Flag

The first three fields must be left-justified and padded with blanks. The *gpname* field can be left blank; RACF then uses the default specified on the ADDUSER command. The eight flag bytes are set to hexadecimal zeros. For example, if the DFSMShsm startup procedure is called DFHSM and you want to define a user ID of HSMUSR for DFHSM, the format of the entry you must create is:

DFHSM	HSMUSR		
8 × X'00'			

See your security administrator to have the correct entries made in ICHRIN03.

Note: If you are using remote sharing functions (RRSF) to propagate RACF commands to remote hosts, see the notes under "The RACF User ID for DFSMShsm" on page 578.

Always-Call Support for Migrated Data Sets and Backup Copies

The DFSMShsm option NORACFIND in the SETSYS command requires that you protect DASD migration copies and backups with generic profiles in an always-call environment. DFSMShsm does not set the RACF indicator on for these data sets.

The SETSYS command has MIGRATEPREFIX and BACKUPPREFIX parameters that allow all migration data sets and backup data sets to have a high-level qualifier selected by the installation.

When the SETSYS MIGRATEPREFIX and BACKUPPREFIX parameters are used, you can define one or two generic profiles to protect all the migrated and backed up data sets. These profiles should have an empty (or nearly empty) access list and a UACC value of NONE. The following example shows how RACF definitions can be set up to protect the migrated and backed up data sets.

```
SETROPTS GENERIC (*)
ADDGROUP (prefix)
ADDGROUP (bprefix)
ADDSD 'prefix.*' UACC(NONE)
ADDSD 'bprefix.*' UACC(NONE)
```

For an explanation of *prefix* and *bprefix*, see "Specifying the First Qualifier for Names of Migrated Data Sets" on page 37 and "Specifying the First Qualifier for Names of Backup Versions" on page 173.

- Set the system-wide RACF options to include generic profiles.
- If you choose MIGPREF as the migration prefix and BAKPREF as the backup prefix, generic profiles MIGPREF.* and BAKPREF.* protect all the DFSMShsm migrated and backed up data sets.
- The ADDSD command builds a generic profile for MIGPREF and another for BAKPREF.

Migration and Recall from DASD

When performing DFSMShsm TSO commands for a user, DFSMShsm never uses the real migrated data set name to ask RACF for authorization. It always asks for authorization on the original data set name on the original volume.

If the original data set was protected by a generic profile, DFSMShsm asks for authorization with the RACFIND=NO option, causing RACF to check for generic profile protection. The same protection that was in effect for the original data set applies, regardless of whether a discrete or generic profile was in effect.

The same protection also continues to apply when a data set is moved to an ML2 volume.

Backup and Recovery from DASD

As with migration, DFSMShsm refers to the RACF profile of the active data set for all TSO user interactions with their backups.

To recover a RACF-indicated data set, the corresponding profile must exist. This can be a problem with discrete profiles because deleting the data set also deletes the discrete profile. When the discrete profile has been deleted, a new one must be rebuilt by an appropriately authorized person before the data set can be recovered by a non-DFSMShsm-authorized user.

DFSMShsm provides the capability to back up RACF discrete profiles when a RACF-indicated data set is backed up. If you specify the SETSYS PROFILEBACKUP command, the discrete profiles are backed up. Non-DFSMShsm-authorized users can then recover deleted data sets that had

discrete profiles, because DFSMShsm recovers the backup profile if no discrete profile exists for a RACF-indicated data set.

When a cataloged non-VSAM data set is being recovered to another volume, DFSMShsm RECOVER gives some assistance by updating the volume serial information of the original RACF discrete profile, if it still exists, or of the recovered backup profile.

When you specify the NEWNAME and REPLACE parameters to recover a VSAM data set, DFSMShsm deletes any information about newly named data sets from the catalog before it recovers the data set. The newly named data set gets the catalog information that the original data set had, including password and RACF-indication. You must provide the proper RACF protection for the newly named data set.

With generic profiles, most of these considerations do not apply, and no extensive security administration is required. If a generic profile exists that covers data sets that were backed up with discrete profiles, the data sets can be recovered by non-DFSMShsm-authorized users whether or not the discrete or backup profiles exist, provided the user is authorized access to the data sets.

Aggregate Backup and Recovery Data Set Security

When an ABACKUP or ARECOVER command is issued, DFSMShsm passes the RACF FACILITY class profile name for the command, along with the user's or console operator's user ID, to RACF for authorization checking. If authorization is denied by RACF, DFSMShsm rejects the request to issue the command. If RACF responds that no profile is defined for the command, DFSMShsm processes the request only if the user is a DFSMShsm-authorized user or a console operator. If the user is not a DFSMShsm-authorized user, DFSMShsm fails the request to issue the command. See "Checking Authorization during Aggregate Backup" on page 304 for a discussion of comprehensive and restricted ABACKUP and ARECOVER commands.

Backing Up RACF-Indicated Data Sets

When an existing data set with a RACF discrete profile is backed up to DASD and the NORACFIND parameter of the SETSYS command has been specified, DFSMShsm does not set the RACF indicator on in the VTOC entry of the backup version.

Authorizing the Use of Concurrent Copy for Data Set Backup Commands

DFSMShsm uses the STGADMIN.ADR.DUMP.CNCURRNT FACILITY class resource to limit concurrent copy use to those users who are authorized to use the CC option.

By default, all users can use the CC option. However, if RACF indicates a lack of authority, DFSMShsm fails the data set backup request if the concurrent copy request is specified as REQUIRED. If the concurrent copy request is not specified as REQUIRED and RACF indicates a lack of authority, DFSMShsm continues to back up the data set as if the concurrent copy keyword were not specified on the backup command.

Small Data Set Packing Data Set Security

Because data sets that migrate to small data set packing (SDSP) data sets are treated as records in the SDSP data set, they must be protected by applying security to the SDSP data set. The security applied to the SDSP data set prevents non-DFSMShsm access to the data in the data sets treated as records.

To maintain data set security on your system's small data set packing data sets, RACF-protect them with a discrete or generic profile. Erase-on-scratch should be specified in the discrete or generic profile if erase-on-scratch is specified for DFSMShsm data sets not in SDSP data sets.

If data sets on volumes managed by DFSMShsm are protected by a password and you specified SETSYS SMALLDATASETPACKING, protect the SDSP data sets on ML1 volumes with a discrete or generic profile to extend the protection of the data sets contained in the SDSP data sets. DFSMShsm can still access the data sets in a protected SDSP data set without being given access authority to the SDSP data set.

Uncataloging Migrated Data Sets

The DELETE *data set name* NOSCRATCH command can be used to uncatalog migrated data sets. For this command to be used when RACF protection is used for DFSMShsm-owned data sets, two conditions must exist:

- A RACF group of ARCCATGP must exist.
- The user who issues the DELETE NOSCRATCH command must be authorized to group ARCCATGP.

Controlling Command Access with RACF FACILITY Class

RACF FACILITY class profiles, both generic and discrete, can be used to allow access to all DFSMShsm commands. If the RACF FACILITY class is active, DFSMShsm always uses it to protect all DFSMShsm commands. If the RACF FACILITY class is not active, DFSMShsm uses the AUTH command to protect all DFSMShsm commands. The storage administrator must work with the security administrator to give authorization to user and storage administrator commands.

Related reading: For additional information about using the RACF FACILITY class environment and the AUTH command environment, see the *z/OS DFSMShsm Implementation and Customization Guide*.

Part 5. DFSMShsm Data Recovery Scenarios

The data recovery scenarios that are presented in this document are real-life examples of data loss situations. Someday, some other computer installations are going to have these same data loss situations. When that happens, storage administrators can refer to this document and see what was done to restore the data, and apply that information to their own systems.

Not all possible data loss situations are covered in this document. However, if you have had a memorable experience with data loss and recovery, and think that other DFSMShsm users could benefit from your experience, send us a Readers' Comment form with the details. You may find your story in the next edition of this book.

Note: Many data loss situations can be avoided by performing administrative steps prior to DFSMShsm implementation.

This part contains one chapter. This chapter describes a variety of situations where data is lost, and what steps to take to recover that data. It also includes situations where the lost data is a control data set or journal.

© Copyright IBM Corp. 1984, 2004

Chapter 20. Data Recovery Scenarios

In the data loss and recovery scenarios that follow, a description of a problem is presented, followed by the corrective measures that you can use for that problem. When you have a data-loss situation, you can quickly scan our examples to see if one of the examples matches your problem. If so, you may be on your way to a speedy recovery.

Case 1: Damaged CDS, Full Journal

It's a cold Monday morning when Rick, a storage administrator, arrives at work to find the following note on his desk:



What does Rick do after he regains consciousness?

The first thing Rick does is review the DFSMShsm command activity log to see what caused the ARC0026E journaling error. He sees that the ARC0026E error specifies an EOV problem; the journal has been completely filled up. Rick then examines the backup activity log to find out what caused the journal to completely fill up. He sees that control data set (CDS) backups fail each time the BACKVOL

© Copyright IBM Corp. 1984, 2004

CDS command is issued. Since the BACKVOL CDS command is failing, the journal is not getting backed up. Rick also sees that IDCAMS has issued error messages indicating out-of-sequence records in the backup control data set (BCDS). The failures recorded in the backup activity log indicate a "broken" BCDS.

To review the backup activity log, Rick used the DFSMShsm RELEASE HARDCOPY command. HARDCOPY is an optional parameter of the RELEASE command that is used to specify that DFSMShsm close any of the four activity logs if they contain data, and either print the logs or send them to DASD. The DFSMShsm SETSYS ACTLOGTYPE command determines whether the logs are printed or sent to DASD.

When the activity logs are closed, new copies of the logs are allocated for use.

Once Steve had reset the journal, DFSMShsm was able to operate because DFSMShsm addresses the virtual storage access method (VSAM) control intervals directly by key. Backup of the CDS failed because a valid VSAM sequence set is required to read CDSs in key sequence.

What happened to the BCDS? Two MVS images, both having DFSMShsm installed, updated the shared BCDS concurrently. When this happened, duplicate records were created in the BCDS, and the index structure of the BCDS was damaged.

After Rick figures out what happened and what damage was done, he plans his salvage operation. Although the journal records were lost when Steve deleted and reallocated the journal, the same information still exists in the BCDS. The BCDS, however, has duplicate records. So the first thing Rick must do is to delete the duplicate records in the BCDS and back up the CDSs. To delete the duplicate records in the BCDS and back up the CDSs, Rick performs the following tasks:

1. Stops DFSMShsm on all images.

   _____

2. Uses the IDCAMS REPRO command to copy the data component of the BCDS to a sequential file. This can be performed by one of the following methods:
   a. Direct the IDCAMS REPRO command to make a copy of the data component of the BCDS, as in the following example:
      ```
      REPRO IDS(HSM.BCDS.DATA) ODS(RICK.BCDS.DATA)
      ```
   b. Direct the IDCAMS REPRO command to make a copy of the BCDS by relative byte address, as in the following example:
      ```
      REPRO IDS(HSM.BCDS) ODS(RICK.BCDS.DATA) FADDR(0)
      ```

   _____

3. Uses DFSORT to perform the following:
   a. Sort the sequential file by key, eliminating records with duplicate keys. The record that was most recently updated, where there are duplicates, is the one that is retained.
   b. Reload the BCDS. The damaged BCDS must be deleted and a new BCDS defined.

   _____

4. Starts DFSMShsm, and issues the DFSMShsm BACKVOL CDS command to create a good copy of the BCDS and nullify the journal.

   _____

Rick used the following JCL to accomplish the preceding steps:

```
//RECOVER JOB
//*******************************************************************
//*PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//*?USERID -TO YOUR USER ID
//*?DFSMSHSM -TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//*******************************************************************
//*THE FIRST STEP WILL CREATE A COPY OF ALL OF THE RECORDS CONTAINED
//*IN THE DATA COMPONENT OF THE BCDS.THESE RECORDS WILL NOT BE IN
//*ORDER,AND MAY CONTAIN MULTIPLE RECORDS WITH THE SAME KEY.IN
//*ADDITION,A RECOVERY BCDSWILL BE CREATED IN THIS STEP.THE USER
//*MUST SUPPLY THE PROPER SPACE INFORMATION,VOLUME INFORMATION,AND
//*DATA SET INFORMATION FOR THE DEFINE CLUSTER.
//*******************************************************************
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DDIN1 DD DISP=SHR,DSN=?DFSMSHSM.BCDS.DATA
//DDOUT1 DD DISP=(,CATLG),UNIT=SYSDA,
// DCB=(LRECL=2052,BLKSIZE=20524,RECFM=VB),
// SPACE=(CYL,(50,10),RLSE),DSN=?USERID.BCDS.RECORDS
//SYSIN DD *
 REPRO INFILE(DDIN1)OUTFILE(DDOUT1)FADDR(0)
 DEFINE CLUSTER(NAME(?USERID.RECOVER.BCDS)-
 CYL(004)SPEED BUFFERSPACE(530432)-
 VOL(XXXXXX)-
 CISZ(12288)-
 FREESPACE(0 0)-
 KEYS(44 0)-
 RECORDSIZE(2040 2040))-
 DATA(NAME(?USERID.RECOVER.BCDS.DATA))-
 INDEX(NAME(?USERID.RECOVER.BCDS.INDEX))
/*
//*******************************************************************
//*THE NEXT STEP USES DFSORT AS FOLLOWS:
//*-THE SORT OPERATOR WILL SORT THE RECORDS EXTRACTED FROM
//*THE BCDSDATA COMPONENT.THEY WILL BE SORTED BY THEIR
//*VSAM KEYS,AND BY THE TIME STAMP OF THE LAST UPDATE MADE
//*TO THE RECORD.THE KEYS WILL BE SORTED IN ASCENDING ORDER,
//*AND THE UPDATE TIME IN DESCENDING ORDER;THE MOST CURRENT
//*UPDATED RECORD BEING FIRST IN THE OUTPUT DATA SET.
//*-THE SELECT OPERATOR (NEW IN DFSORT R12)WILL SORT THE
//*RECORDS BY THEIR VSAM KEYS.THE ORDER ESTABLISHED BY THE
//*SORT OPERATOR,WHERE THE MOST RECENTLY UPDATED VERSION OF
//*A DUPLICATE RECORD IS PLACED FIRST IN THE FILE,IS MAINTAINED.
//*THE DUPLICATE RECORD THAT IS NOT THE MOST RECENTLY UPDATED
//*VERSION IS DISCARDED.THE OUTPUT IS LOADED INTO THE RECOVERY
//*COPY OF THE BCDSTHAT WAS DEFINED IN THE FIRST STEP.
//*******************************************************************
//STEP2 EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDS.RECORDS
//SORTOUT DD DISP=(,PASS),DSN=&&TFILE1,
// UNIT=SYSDA,SPACE=(CYL,(50,10),RLSE),
// DCB=(LRECL=2052,BLKSIZE=0,RECFM=VB,DSORG=PS)
//*SORT THE OLD BCDS BY ASCENDING SAM KEY AND
//*DESCENDING TIME STAMP OF LAST UPDATE.
//SYSIN DD *
 SORT FIELDS=(5,44,CH,A,53,8,BI,D)
/*
//STEP3 EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=&&TFILE1
//SORTOUT DD DISP=OLD,DSN=?USERID.RECOVER.BCDS
//*LOAD ONLY THE MOST RECENTLY UPDATED RECORDS INTO THE NEW BCDS
//SYSIN      DD *
 SORT FIELDS=(5,44,CH,A),EQUALS
 SUM FIELDS=(NONE)
/*
```

Case 2: Damaged Journal, Undamaged CDS

A damaged journal is often the result of a hardware-caused error on the journal volume. DFSMShsm issues error message ARC0026E when it encounters an input/output (I/O) error in the journal data set. Message ARC0026E states "JOURNALING DISABLED DUE TO I/O ERROR. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPEREPL, RECYCLE, ARECOVER, AUDIT, AND EXPIREBV HELD."

Once the problem occurs, follow this procedure:

1. Verify the DFSMShsm CDSs are structurally sound.

   Using IDCAMS, run EXAMINE against each of the CDSs. The procedure runs very quickly, and does not require that you bring DFSMShsm down. The sample JCL to perform this procedure (which follows) comes from *z/OS DFSMS Access Method Services for Catalogs*.

```
//EXAMCDS   JOB
//STEP1     EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *
   EXAMINE -
     NAME(?hlqual.MCDS) -
     ERRORLIMIT(0)
//STEP2     EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *
   EXAMINE -
     NAME(?hlqual.BCDS) -
     ERRORLIMIT(0)
//STEP3     EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *
   EXAMINE -
     NAME(?hlqual.OCDS) -
     ERRORLIMIT(0)
```

   If there are no errors reported by EXAMINE, continue to the next step.

   If errors ***are*** reported by EXAMINE, do not proceed to the next step. Go to "Case 1: Damaged CDS, Full Journal" on page 585.

   _____

2. Stop DFSMShsm on all images.

   _____

3. Do ***not*** copy the journal.

   _____

4. Rename the current journal.

   _____

5. Allocate a new journal with the old name. The following JCL allocates a new journal with the old name.

```
//JRNLDEF  JOB
//STEP1   EXEC PGM=IEFBR14
//JRNLDD   DD  DISP=(,CATLG),UNIT=?3390,VOL=SER=?JNLVOL,
//             SPACE=(CYL,(100),,CONTIG),DSN=?hlqual.JRNL
```

   _____

6. Start DFSMShsm in EMERG=YES mode, a proclib option.

   _____

7. Issue the DFSMShsm BACKVOL CDS command to keep your CDSs in synchronization with the new journal.

   _____

8. Delete the old journal.

   _____

9. Issue the DFSMShsm SETSYS NOEMERGENCY command and then the DFSMShsm RELEASE command to resume normal processing.

Case 3: Full Journal, Undamaged CDS

The problem of a full journal with an undamaged CDS occurs because the journal is not large enough or the CDSs are not being backed up often enough. DFSMShsm issues error message ARC0909E when the DFSMShsm journal threshold is met, and continues to issue the error message each time an additional one percent (or more) of the journal is filled. When the journal is full, ARC0909E states: "JOURNAL DATA SET IS ABOUT 100% FULL." When the journal is 100% full, message ARC0026E is also issued, which states: "JOURNALING DISABLED DUE TO EOV ERROR ON JOURNAL. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPEREPL, RECYCLE, ARECOVER, AUDIT, AND EXPIREBV HELD."

Once the problem has occurred, take this step:

- Issue the BACKVOL CDS command. The BACKVOL CDS command will back up the journal and the control data sets, and null the journal.

Case 4: Structurally Damaged CDS, Missing Journal Records

A structurally damaged CDS is usually caused by multiple updates occurring at the same instant in time within the same CDS. Simultaneous updates to the same CDS can occur in a multiple-MVS-image environment, if a job running on one image reorganizes the CDS while DFSMShsm is still running on another image. Without proper share options and serialization methods, the problem of a structurally damaged CDS along with missing journal records can easily happen.

Problems arise when there are no CDS and journal backup versions prior to the damage. This commonly occurs when damage to a CDS is not immediately detected. As new CDS and journal backups are being created, the older versions are being deleted. By the time the damaged CDS is detected, the backup versions that were created prior to the error have been deleted.

When you have a structurally damaged CDS and missing journal records, you must work with the records currently in your damaged CDS. To repair your damaged CDS, follow the same steps that Rick followed in "Case 1: Damaged CDS, Full Journal" on page 585, substituting your damaged CDS for Rick's backup control data set.

Case 5: Time Sequence Gaps in CDS Records, Journal Records Also Missing

During a reorganization, user error may cause time sequence gaps in the CDS records, along with missing journal records.

Take the case where an installation uses generation data sets (GDSs) when reorganizing the CDSs. The user attempts to use IDCAMS REPRO to create an output data set for a migration control data set (MCDS). The MCDS is not successfully copied to the output data set, but the user does not notice. Then the user employs IDCAMS REPRO to create a new, reorganized MCDS, not knowing that the input data set is actually the last version of the MCDS successfully copied and is not the current MCDS at all.

After the reorganization completes, the administrator thinks that everything went as planned, and continues with everyday processing.

DFSMShsm may run with an old copy of the MCDS for a week before a high number of data set recall failures signal that there is a problem. The old MCDS that is being used, of course, does not contain the more recently migrated data sets.

When the storage administrator finally discovers what has happened, he looks to see if he can recover from a backup version. He backs up once a day, and keeps three backup versions. The reorganization, however, was performed seven days ago. That means he doesn't have any backups prior to when the reorganization occurred, for either the journal or the MCDS.

Recovery in this instance is very complicated. The best course of action is to call your IBM Area Storage Specialist for recovery assistance.

Case 6: Overwritten Migration Tape

This recovery scenario applies only to an overwritten header label. If the damage to the tape is more extensive than an overwritten label, the following recovery procedure will not work. If the tape cannot be recovered due to hardware problems related to the drive itself, consider sending it to your tape drive manufacturer for recovery.

If you have a duplex or a tape copy, perform a tape replace. You can reference this procedure in the *z/OS DFSMShsm Storage Administration Reference*.

In either case, you should route the recovery assistance through IBM Consultline. You may open a software problem report to diagnose the root cause of the damage, if it is not known.

Ensure that you have a backup copy of any data set that is going to tape (Migration Level 2). If a backup copy is available, you can recover from a backup version without having to recover from the damaged tape.

The following information is relevant to know or have available when you open a problem record:

1. Specific failure indications—full message text of any related message that was issued
2. Type of drives that you are using
3. Microcode levels for the drive and controller
4. Reasonably available history for the tape
5. Pervasiveness of the problem
6. Attempted recovery actions taken–RECALL, RECYCLE, DELVOL, and so forth.
7. Customers should gather the following information when the problem arises:

- List TTOC of the affected volume
- Library Manager Log for an ATL
- CTRACE (if you are already writing to an external writer)
- SYSLOG
- DFSMShsm PDA trace
- A dump on the failure, if it can be trapped using a slip (ABEND637, for example).

In some cases, you cannot recover some or any of the data (depending on the extent of the damage). A backup of the tape ensures that this is not a concern. Back up the data, if the data matters, before going to Migration Level 2 (ML2).

The main reason for this type of damage is that the wrong tape volume is mounted to satisfy a mount request for a scratch tape. The problem usually remains undiscovered unless you need the overwritten tape volume to recall a data set.

Use the following recovery procedure to restore access to DFSMShsm data on tape cartridges. This procedure will not work for reel-type tape because of a difference in file format.

1. Copy the damaged tape volume.

   (This step is optional.) If the user data that was written over the DFSMShsm data is wanted, you will need to create a copy of that data. The damaged tape volume will be returned to scratch or to DFSMShsm for its later use, and the user data that was written on this volume will be lost at the end of this procedure.

   ---

2. Reinitialize the damaged tape volume.

   Reinitialize the tape volume with the data set name that is used by DFSMShsm for all of its tape volumes. Write a dummy data set with the first file on the volume named as below:

   prefix.**HMIGTAPE.DATASET**

   The *prefix* can be any valid qualifier. For example, you can use the migration prefix as specified on the SETSYS MIGRATEPREFIX command, or the UID. This step will create a good tape label on the damaged tape. Once the tape label is valid, the volume can be processed by RECYCLE in step 6 on page 592 of this procedure.

   IEBGENER can be used to perform the reinitialization task.

   The sample JCL that follows can be used to reinitialize a damaged migration volume. Modify the JCL to fit your needs by doing the following tasks:

   - Change the UNIT parameter on the SYSUT2 DD statement, if necessary.
   - Change *prefix* to the appropriate prefix.
   - Change *ml2volser* to your volume serial number.

   Included in the example that follows are the various messages that are generated from this job and the appropriate replies.

   Warning: Do not perform the AUDIT MEDIACONTROLS function. AUDIT MEDIACONTROLS will remove all of your data.

```
//GENR     JOB
//STEP1    EXEC PGM=IEBGENER
//SYSUT1   DD DUMMY,LRECL=16384,BLKSIZE=16384,RECFM=F
//SYSUT2   DD DISP=(NEW,KEEP),DCB=*.SYSUT1,UNIT=?3480,
//            DSN=prefix.HMIGTAPE.DATASET,VOL=SER=?ml2volser
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
/*
                     J E S 2  J O B  L O G

 JOB09371  $HASP373 HAMMERTA STARTED - INIT 3 - CLASS A - SYS ..
 JOB09371 *IEF233A M 5A8,HAMMRT,,HAMMERTA,STEP1
 JOB09371  IEC512I LBL ERR 5A8,SMFDMP,SL,HAMMRT,SL,HAMMERTA,STEP1
 JOB09371 *IEC507D E 5A8,SMFDMP,HAMMERTA,STEP1,DMP.T241T240.DATA
 JOB09371  IEC507D REPLY 'U'-USE OR 'M'-UNLOAD
 JOB09371  R 78,U
 JOB09371 *IEC534D A 5A8,SMFDMP,SL,HAMMERTA,STEP1
 JOB09371  IEC534D REPLY 'U'-USE OR 'M'-UNLOAD
 JOB09371  R 84,U
 JOB09371 *IEC704A L 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
 JOB09371  IEC704A REPLY 'VOLSER,OWNER INFORMATION','M'OR'U'
 JOB09371  R 87,'HAMMRT,DFSMSHSM'
 JOB09371  IEC705I TAPE ON 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
 JOB09371  IEF234E K 5A8,HAMMRT,PVT,HAMMERTA,STEP1
 JOB09371  HAMMERTA STEP1  : 00:17:30/ 00:00:00/    53 - R000
 JOB09371  $HASP395 HAMMERTA ENDED
```

3. Determine if the first data set on the damaged tape volume spanned from another volume.

   Issue the LIST TTOC(*volser*) ODS(*dsname*) command to determine if your damaged tape volume is associated with a previous tape volume. If your damaged tape is associated with a previous tape, it will be listed under the heading "PREV VOL." When there is a previous tape, it means that the first data set on your damaged tape began on the previous tape.

   To verify whether your damaged tape is associated with a previous tape, issue the LIST TTOC(*prev_volser*) command and check to see that the last data set name on the previous tape is the same as the first data set name on your damaged tape.

4. Delete the first data set on your damaged tape volume if it spanned from a previous tape volume.

   To delete the first data set on a damaged migration tape, issue the following command:

   ```
   DELETE dsn
   ```

5. Issue the following command to recover the first data set:

   ```
   RECOVER dsn
   ```

6. Depending on the type of error that you are recovering from, it may be necessary to run RECYCLE several times. Issue the following command:

   ```
   RECYCLE VOLUME(volser) EXECUTE
   ```

   This step should be repeated to ensure that all available data sets are moved without the use of the FORCE parameter of the RECYCLE command.

   Note: Readable, complete, and valid data sets are subsequently recycled to a new volume.

7. Identify data sets that have been overwritten and build appropriate recovery commands.

   Create a list showing which migrated data sets you need to recover (because the RECYCLE command was not able to remove them from the volume), by issuing the following command:

   `LIST TTOC(volser) ODS(dsn)`

   Build RECOVER commands for each data set on the list, but do not process the commands until step 9 on page 594.

   An alternate to using the LIST TTOC command is to use DFSORT. The following is an example of a DFSORT job that identifies and builds RECOVER commands for each migrated data set that was overwritten. This job only builds the RECOVER commands; it does not process them.

```
//RCVCMDS JOB
//*****************************************************************
//* Please change all the following to the appropriate values:
//* ?USERID - to your user id
//* ?DFSMShsm - to the DFSMShsm prefix for your MCDS
//* ?VLSER - to the volser of your damaged tape volume
//*****************************************************************
//* This step uses SORT utility to build RECOVER
//* commands.
//*****************************************************************
//STEP1 EXEC PGM=SORT,REGION=4096K
//*****************************************************************
//* The INCLUDE statement selects MCD records (X'00'
//* at offset 51), that have the volser of your damaged tape and
//* have the MCDFASN bit on.
//* The OUTREC statement builds a RECOVER command for each
//* migrated data set found.
//*****************************************************************
//SYSOUT DD SYSOUT=*
//MCDS DD DSN=?DFSMSHSM.MCDS,DISP=SHR
//CMDS DD DSN=?USERID.TEST.SORTMCD,DISP=(NEW,CATLG),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=VB),
// SPACE=(CYL,(5,1),RLSE),UNIT=SYSDA
//SYSIN DD *
SORT FIELDS=(COPY)
INCLUDE COND=((51,1,BI,EQ,X'00'),AND,
                            (69,6,CH,EQ,C'?VLSER'),AND,
                            (75,1,BI,EQ,B'1.......'))
OUTREC FIELDS=(1,4,C' HSEND RECOVER ',5,44)
/*
```

8. Issue the RECYCLE VOLUME(*volser*) FORCE EXECUTE command for the damaged volume.

   The FORCE parameter of the RECYCLE command causes all data sets with errors to be deleted from the control data sets and the catalog. The deletion makes the tape empty, and the tape is removed from DFSMShsm's inventory of tapes that contain data.

   If the RECYCLE command fails with return code 16, you must issue a second RECYCLE VOLUME command to recycle the valid data sets while processing in single buffer mode. For an explanation of return code 16, use LookAt or see *z/OS MVS System Messages, Vol 2 (ARC-ASA)* and the table in the appendix titled "Reason Codes for Message ARC0734I when the Action is RECYCLE and Return Code is 16." For a description of LookAt, see "Using LookAt to look up message explanations" on page xvii.

9. Issue the RECOVER commands for the migrated data sets that were lost. The RECOVER commands were generated in step 7 on page 593.

Case 7: Overwritten Backup Tape

This recovery scenario applies only to an overwritten header label. If the damage to the tape is more extensive than an overwritten label, the following recovery procedure will not work. If the tape cannot be recovered due to hardware problems related to the drive itself, consider sending it to your tape drive manufacturer for recovery.

If you have a duplex or a tape copy, perform a tape replace. You can reference this procedure in the *z/OS DFSMShsm Storage Administration Reference*.

In either case, route the recovery assistance through IBM Consultline. Open a software problem report to diagnose the root cause of the damage, if it is not known.

When you open a problem record, you should be aware of the following information:

1. Specific failure indications—full message text of any related message issued
2. Type of drives
3. Microcode levels for the drive and controller
4. Reasonably available history for the tape
5. Pervasiveness of the problem
6. Attempted recovery actions taken–RECALL, RECYCLE, DELVOL, and so forth.
7. Customers should gather the following information when the problem arises:
   - List TTOC of the affected volume
   - Library Manager Log for an ATL
   - CTRACE (if already writing to an external writer)
   - SYSLOG
   - DFSMShsm PDA trace
   - A dump on the failure if it can be trapped using a slip (for example, ABEND637).

The main reason for this type of damage is that the wrong tape volume gets mounted to satisfy a mount request for a scratch tape. The problem usually remains undiscovered unless you need the overwritten tape volume to recover a data set.

Use the following recovery procedure to restore access to DFSMShsm data on single-file-format backup tapes that were partially overwritten by user data.

1. Copy the user data to a safe place.

   (This step is optional.) If the user data that was written over the DFSMShsm data is wanted, you will need to create a copy of that data. The damaged tape volume will be returned to scratch or to DFSMShsm for its later use, and the user data that was written on this volume will be lost at the end of this procedure.

2. Reinitialize the damaged tape volume.

Reinitialize the tape volume with the data set name that is used by DFSMShsm for all of its tape volumes. Write a dummy data set with the first file on the volume named as below:

prefix.**BACKTAPE.DATASET**

The *prefix* is one of the following:

- The backup prefix as specified on the SETSYS BACKUPPREFIX command
- The UID if prefixes are not specified with the SETSYS commands

This step will create a good tape label on the damaged tape. Once the tape label is valid, the volume can be processed by RECYCLE in step 5 on page 596 of this procedure.

IEBGENER can be used to perform the reinitialization task.

The sample JCL that follows can be used for a damaged backup volume. Modify the JCL to fit your needs by doing the following tasks:

- Change the VOL=SER parameter, as needed.
- Change the *prefix*, as needed.

Included in the example that follows are the various messages that are generated from this job and the appropriate replies.

```
//GENR     JOB
//STEP1    EXEC PGM=IEBGENER
//SYSUT1   DD DUMMY,LRECL=16384,BLKSIZE=16384,RECFM=F
//SYSUT2   DD DISP=(NEW,KEEP),DCB=*.SYSUT1,UNIT=?3480,
//            DSN=prefix.BACKTAPE.DATASET,VOL=SER=?ml2volser
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
/*
                     J E S 2  J O B  L O G

 JOB09371  $HASP373 HAMMERTA STARTED - INIT 3 - CLASS A - SYS ..
 JOB09371 *IEF233A M 5A8,HAMMRT,,HAMMERTA,STEP1
 JOB09371  IEC512I LBL ERR 5A8,SMFDMP,SL,HAMMRT,SL,HAMMERTA,STEP1
 JOB09371 *IEC507D E 5A8,SMFDMP,HAMMERTA,STEP1,DMP.T241T240.DATA
 JOB09371  IEC507D REPLY 'U'-USE OR 'M'-UNLOAD
 JOB09371  R 78,U
 JOB09371 *IEC534D A 5A8,SMFDMP,SL,HAMMERTA,STEP1
 JOB09371  IEC534D REPLY 'U'-USE OR 'M'-UNLOAD
 JOB09371  R 84,U
 JOB09371 *IEC704A L 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
 JOB09371  IEC704A REPLY 'VOLSER,OWNER INFORMATION','M'OR'U'
 JOB09371  R 87,'HAMMRT,DFSMSHSM'
 JOB09371  IEC705I TAPE ON 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
 JOB09371  IEF234E K 5A8,HAMMRT,PVT,HAMMERTA,STEP1
 JOB09371  HAMMERTA STEP1  : 00:17:30/ 00:00:00/    53 - R000
 JOB09371  $HASP395 HAMMERTA ENDED
```

3. Determine if the first data set on the damaged tape volume spanned from another volume.

   Issue the LIST TTOC(*volser*) ODS(*dsname*) command to determine if your damaged tape volume is associated with a previous tape volume. If your damaged tape is associated with a previous tape, it will be listed under the heading "PREV VOL." When there is a previous tape, it means that the first data set on your damaged tape began on the previous tape.

   To verify whether your damaged tape is associated with a previous tape, issue the LIST TTOC(*prev_volser*) command and check to see that the last data set name on the previous tape is the same as the first data set name on your damaged tape.

4. Delete the first data set on your damaged tape volume if it spanned from a previous volume.

   Issue the following command:

   ```
   BDELETE dsn VERSIONS(n)
   ```

   The original data set name and the backup generation number will be required. To get this information, issue the FIXCDS command to display the MCC record for the backup version. The backup version generation number is a 2-byte field contained in the MCCGEN field at offset X'46' in the FIXCDS display output. For example, issuing the following command:

   ```
   HSEND FIXCDS C DFHSM.BACK.T500718.TONY3.AHRENS.I3006
   ```

   gives you the following results:

```
MCH=  01602400 A6F7B92B F4BBC221 A6DAB593 9C8BBA03
*     7  4 B                    *
+000  E3D6D5E8 F34BC1C8 D9C5D5E2 4BC2C3C4 E2404040 40404040 40404040 40404040
*TONY3.AHRENS.DATA              *
+020  40404040 40404040 40404040 E7E8E9F9 F0F50003 78048081 18075106 0093006F
*            XYZ905             *
+040  00081000 0000001C 000004DD 0121C0E2 0000243C 00003060 0000E3D6
D5E8F0F5
                         .
                         .
```

   Issue the following command to show all versions of the data set if the tape spanned before BDELETE:

   ```
   LIST dsn
   ```

   The output gives you the original data set name: `TONY3.AHRENS.BCDS` and the version number: `+040  00081000 000000 1C 000004DD`

   The deletion would incorporate the information as follows:

   ```
   BDELETE DSN(TONY3.AHRENS.DATA) VERSION(28)
   ```

   Note: The version number gets converted from 1C to decimal (28).

5. Depending on the type of error that you are recovering from, it may be necessary to run RECYCLE several times. Issue the following command:

   ```
   RECYCLE VOLUME(volser) EXECUTE
   ```

   This step should be repeated to ensure that all available data sets are moved without the use of the FORCE parameter of the RECYCLE command.

   Note: Readable, complete, and valid data sets are recycled to a new volume.

6. Identify data sets that were overwritten.

   Build BACKDS commands for all the backup data sets that were overwritten and that are the most recent backup version. The following DFSORT job can identify and build the BACKDS commands for these data sets. This job only builds the BACKDS commands; it does not process them.

```
99//BDSCMDS JOB
//*******************************************************************
//*PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//*?USERID -TO YOUR USER ID
//*?DFSMSHSM -TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//*?VLSER -TO THE VOLSER OF YOUR DAMAGED BACKUP VOLUME
//*******************************************************************
//*THIS STEP USES SORT UTILITY TO FIND ALL
//*ASSOCIATED MCB AND MCC RECORDS,IDENTIFYING THE MOST RECENT
//*BACKUP VERSION THAT WAS OVERWRITTEN ON THE DAMAGED BACKUP
//*VOLUME.IT ALSO GENERATES THE BACKDS COMMANDS TO CREATE A
//*NEW BACKUP VERSION.
//*******************************************************************
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?DFSMSHSM.BCDS,AMP='BUFND=50'
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.BCDSSRT1,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=2048,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
 SORT FIELDS=(COPY)
 INCLUDE COND=(1,4,BI,GE,X'00C1')
/*
//STEP2 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDSSRT1
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.BCDSSRT2,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=2048,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
 SORT FIELDS=(COPY)
 INCLUDE COND=(51,1,BI,EQ,X'20')
 OUTREC FIELDS=(1,4,5,44,X,149,44,X'0001')
/*
//STEP3 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDSSRT1
//SORTOUT DD DISP=(MOD,CATLG),DSN=?USERID.BCDSSRT2
//SYSIN DD *
 SORT FIELDS=(COPY)
 INCLUDE COND=(51,1,BI,EQ,X'24',AND,113,6,CH,EQ,C'?VLSER')
 OUTREC FIELDS=(1,4,69,44,X,5,44,X'0001')
/*
//STEP4 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDSSRT2
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.MERGED.BCDS.RECORDS,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
 SORT FIELDS=(50,44,CH,A)
 SUM FIELDS=(94,2,BI)
/*
//STEP5 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=?USERID.MERGED.BCDS.RECORDS,DISP=SHR
//BKDSREC DD DSN=?USERID.BKDS.RECORDS,DISP=(NEW,CATLG),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//*BUILD HSEND BACKDS COMMANDS.
//*******************************************************************
//*THE INCLUDE STATEMENT SELECTS MCB RECORDS THAT CONTAIN
//*THE VOLSER OF YOUR DAMAGED ML1 VOLUME.
//*THE OUTREC STATEMENT BUILDS AN HSEND BACKDS COMMAND
//*FOR EACH DATA SET FOUND.
//*******************************************************************
//SYSIN DD *
 SORT FIELDS=(COPY)
 OUTFIL FNAMES=BKDSREC,
  INCLUDE=(94,2,BI,EQ,X'0002'),CONVERT,
  OUTREC=(C'HSEND BACKDS ',5,44,255:X)
/*
```

7. Issue the DELVOL *volser* BACKUP(PURGE) command. The DELVOL ... (PURGE) command will perform the same functions as a RECYCLE ... FORCE command against the volume without causing a tape mount. The damaged backup tape volume will be deleted from DFSMShsm inventory and any other control records describing the contents of the volume. DFSMShsm deletes any association with a day in the backup cycle or with a set of spill backup volumes.

8. Issue the BACKDS commands that were generated in step 6 on page 596.

   Note: If you are backing up a data set (with the BACKDS command) that resides on an ML2 tape volume, you need to use the RECALL command to recall the data set prior to issuing the BACKDS command.

Case 8: Damaged ML1 Volumes

To handle a damaged Migration Level 1 (ML1) volume, perform the following steps:

1. Delete the ML1 volume entry of the damaged volume with the following command:

   ```
   DELVOL volser MIGRATION(PURGE)
   ```

2. Initialize another DASD volume of the same device type as the damaged volume, using the volume serial number of the damaged volume.

3. Restore the damaged ML1 volume from a dump copy. If DFSMShsm was used to create the dump copy, you can use the following command to restore the ML1 volume:

   ```
   RECOVER * TOVOLUME(volser) UNIT(unittype) FROMDUMP
   ```

4. If there was a small data set packing (SDSP) data set on the damaged ML1 volume, check to see if there is a backup copy of the SDSP data set that is more recent than the dump copy that is used to restore the volume. If so, recover the backup copy of the SDSP. If an IDCAMS EXPORT command was used to create the backup copy, use the IDCAMS IMPORT command to recover the data set.

5. Add the restored ML1 volume to the list of volumes that DFSMShsm owns. Use the ADDVOL command, specifying the DRAIN parameter. The DRAIN parameter prevents the volume from being selected for migration output. For example, issue the following command:

   ```
   ADDVOL volser UNIT(unittype) MIGRATION(ML1 DRAIN)
   ```

   If there was an SDSP data set on your ML1 volume, be sure to specify the SDSP parameter on your ADDVOL command. For example, issue the following command:

   ```
   ADDVOL volser UNIT(unittype) MIGRATION(ML1 DRAIN SDSP)
   ```

6. Move all valid migration data sets and VTOC copy data sets off the ML1 volume with the following FREEVOL command:

   ```
   FREEVOL MIGRATIONVOLUME(volser) AGE(0) TARGETLEVEL(ML1)
   ```

It is possible to generate tape mounts for ML2 tape volumes if there are valid, migrated, SMS-managed data sets that have reached the criteria for migrating to ML2.

---

7. Move all valid backup versions to backup volumes with the following command:

   ```
   FREEVOL ML1BACKUPVERSIONS
   ```

   The FREEVOL command causes all valid backup versions from all ML1 volumes to be moved to backup volumes.

---

8. Identify all lost migrated data sets.

   The following is an example of a DFSORT job that identifies migrated data sets that are lost, and builds HDELETE and HRECOVER commands for each data set. This job only builds the HDELETE and HRECOVER commands; it does not process them.

```
//SMCD JOB
//*****************************************************************
//*PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//*?USERID -TO YOUR USER ID
//*?DFSMSHSM -TO THE DFSMSHSM PREFIX FOR YOUR MCDS
//*?VLSER -TO THE VOLSER OF YOUR DAMAGED ML1 VOLUME*****************
//*****************************************************************
//*THIS STEP USES DFSORT TO LOCATE ALLHESE RECORDS WILL NOT BE IN
//*MIGRATED DATA SETS LOST FROM THE DAMAGED ML1 VOLUME,ANDY.IN
//*BUILD HDELETE AND HRECOVER COMMANDS FOR EACH DATA SET FOUND.R
//*THE HDELETE COMMANDS ARE PLACED IN THE DATA SET NAMEDMATION,AND
//*USERID.HDELETE.DATA.THE HRECOVER COMMANDS ARE PLACED
//*IN THE DATA SET NAMED USERID.HRECOVER.DATA.**********************
//*****************************************************************
//STEP1 EXEC PGM=SORT,REGION=4096K
//*****************************************************************
//*THE INCLUDE STATEMENT SELECTS MCD RECORDS ('00'AT +51)
//*THAT CONTAIN THE VOLSER OF YOUR RECOVERED ML1 VOLUME
//*(AT OFFSET 69),AND HAVE THE MCDFASN BIT ON (A VALUE GREATER
//*THAN X'80'AT OFFSET 75).
//*THE OUTREC STATEMENT BUILDS AN HDELETE COMMAND FOR EACH MCD
//*RECORD SELECTED.THE DATA SET NAME,WHICH COMES FROM THE
//*MCD RECORD AT OFFSET 5,IS PLACED FOLLOWING THE
//*CHARACTERS HDELETE.
//*****************************************************************
//*THE INCLUDE STATEMENT SELECTS MCD RECORDS ('00'AT +51)
//*THAT CONTAIN THE VOLSER OF YOUR RECOVERED ML1 VOLUME
//*(AT OFFSET 69),AND HAVE THE MCDFASN BIT ON (A VALUE GREATER
//*THAN X'80'AT OFFSET 75).
//*THE OUTREC STATEMENT BUILDS AN HRECOVER COMMAND FOR EACH MCD
//*RECORD SELECTED.THE DATA SET NAME,WHICH COMES FROM THE
//*MCD RECORD AT OFFSET 5,IS PLACED FOLLOWING THE
//*CHARACTERS HRECOVER.
//*****************************************************************
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=?DFSMSHSM.MCDS,DISP=SHR
//HDELETE DD DSN=?USERID.HDELETE.DATA,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=255,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE)
//HRECOVER DD DSN=?USERID.HRECOVER.DATA,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=80,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE)
//*BUILD HDELETE COMMANDS.
//*BUILD HRECOVER COMMANDS.
//SYSIN DD *
 SORT FIELDS=(COPY)
 OUTFIL FNAMES=HDELETE,
                INCLUDE=((51,1,BI,EQ,X'00'),AND,
                         (69,6,CH,EQ,C'?VLSER'),AND,
                         (75,1,BI,EQ,B'1.......')),
                CONVERT,OUTREC=(C'HDELETE ',5,44,255:X)
 OUTFIL FNAMES=HRECOVER,
                INCLUDE=((51,1,BI,EQ,X'00'),AND,
                         (69,6,CH,EQ,C'?VLSER'),AND,
                         (75,1,BI,EQ,B'1.......')),
                CONVERT,OUTREC=(C'HRECOVER ',5,44,255:X)
/*
```

9. Identify all lost backup versions.

   The following is an example of a DFSORT job that identifies backup versions of data sets that are lost and builds HSEND BDELETE commands for those data sets. This job only builds the HSEND BDELETE commands; it does not process them.

```
//SMCC     JOB
//*******************************************************************
//*  Please change all the following to the appropriate values:
//*        ?USERID  -  to your user id
//*        ?DFSMShsm  -  to the DFSMShsm prefix for your BCDS
//*        ?VLSER   -  to the volser of your damaged ML1 volume
//*******************************************************************
//*  This job extracts data from MCC records to create
//*  HSEND BDELETE commands for all backup versions that were lost
//*  from the ML1 volume.
//*  The HSEND BDELETE commands are placed in the data set named
//*  userid.BDEL.DATA.
//*******************************************************************
//* The first step uses the SORT utility to extract
//* data from MCC records to produce skeleton HSEND BDELETE
//* commands and backup version values.
//*******************************************************************
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?DFSMSHSM.BCDS,AMP='BUFND=50'
//SORTOUT DD DISP=(NEW,PASS),DSN=&&SOURTOUT1,
// UNIT=SYSDA,SPACE=(TRK,(15,5),RLSE),
// DCB=(LRECL=2048,BLKSIZE=23476,RECFM=VB)
//SYSIN  DD *
SORT FIELDS=(COPY)
INCLUDE COND=(51,1,BI,EQ,X'24',AND,113,6,CH,EQ,C'?VLSER')
OPTION LSHRT
//*******************************************************************
//*******************************************************************
//STEP2 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=(OLD,PASS),DSN=&&SOURTOUT1
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.MCC.VERS,
// UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN  DD *
SORT FIELDS=(COPY)
OUTREC FIELDS=(1,4,C' HSEND BDELETE ',69,44,C'VERSIONS(0000) ',5,44)
//*******************************************************************
//* This converts the binary 'version' field to display and formats for merge
//*******************************************************************
//STEP3 EXEC PGM=SORT
//SYSOUT DD SYSOUT=*
//SORTIN  DD DISP=(OLD,PASS),DSN=&&SOURTOUT1
//SORTOUT DD DISP=(NEW,PASS),DSN=&&VERS1,UNIT=SYSDA,
// SPACE=(TRK,(10,1),RLSE)
//SYSIN DD *
 SORT FIELDS=COPY
 OUTFIL CONVERT,OUTREC=(X,5,44,3X,139,2,BI,EDIT=(+TTTTTTTTTTTTTT),121:X)
//*******************************************************************
```

(continued on next page)

```
//SMCC     JOB
//*****************************************************************
STEP4 EXEC PGM=SORT,REGION=4096K
//SYSOUT    DD SYSOUT=*
//SORTIN     DD DISP=(OLD,PASS),DSN=&&VERS1,UNIT=SYSDA,
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.MCC.VERS2,
// UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
// DCB=(LRECL=251,BLKSIZE=0,RECFM=FA,DSORG=PS)
//SYSIN DD *
 SORT FIELDS=COPY
OUTREC FIELDS=(69X,61,4,2X,2,44,251:X)
//*****************************************************************
//* This step uses IDCAMS to convert the fixed-length records of
//* ?USERID.MCC.VRS2 to variable-length records compatible with
//* ?USERID.MCC.VRS so these two data sets can be further
//* processed together.
//*****************************************************************
//STEP5 EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//IN DD DISP=OLD,DSN=?USERID.MCC.VERS2
//OUT DD DISP=(NEW,CATLG),DSN=?USERID.VB2.VERSION,
// UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
// DCB=(LRECL=255,BLKSIZ =23476,RECFM=VB)
//SYSIN DD *
REPRO INFILE(IN) OUTFILE(OUT)
/*
//*****************************************************************
//* This step uses SORT utility to combine the BDELETE
//* command skeleton in ?USERID.MCC.VRS with the decimal value
//* for the backup version in ?USERID. B2.VRSION, to produce the
//* final HSEND BDELETE commands in ?USERID.BDEL.DATA.
//*****************************************************************
//STEP6 EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
//SYSOUT  DD SYSOUT=*
//SORTIN   DD DISP=OLD,DSN=?USERID.MCC.VERS
//                 DD DISP=OLD,DSN=?USERID.VB2.VERSION
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.BDEL.DATA,
// UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE),
// DCB=*.COMBIN
//SYSIN    DD *
SORT FIELDS=(80,44,CH,A)
SUM FIELDS=(74,4,ZD)
OUTREC FIELDS=(1,4,5,75)
OPTION ZDPRINT
/*
```

10. Identify which of the lost backup-version data sets were the most recent backup versions of the original data sets.

    The following is an example of a DFSORT job that identifies these data sets, and also builds HSEND BACKDS commands for each of the identified data sets. This job only builds the HSEND BACKDS commands; it does not process them.

    You will have to assess whether to create new backup versions at this time. You may also want to inform the owners of the data sets that the most recent backup versions of their data sets were lost.

```
//SMCB JOB
//*****************************************************************
//*PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//*?USERID, -TO YOUR USER ID
//*?DFSMSHSM -TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//*?VLSER -TO THE VOLSER OF YOUR DAMAGED ML1 VOLUME
//*****************************************************************
//*THIS STEP USES SORT UTILITY TO LOCATE THE MOST
//*RECENT OF ANY BACKUP VERSIONS THAT WERE LOST FROM THE DAMAGED
//*ML1 VOLUME.
//*IT BUILDS HSEND BACKDS COMMANDS FOR EACH BACKUP VERSION FOUND.
//*THE HSEND BACKDS COMMANDS ARE PLACED IN THE DATA SET NAMED
//*USERID.BACKDS.CMDS.
//*****************************************************************
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=?DFSMSHSM.BCDS,DISP=SHR
//CMDS DD DSN=?USERID.BACKDS.CMDS,UNIT=SYSDA,DISP=(NEW,CATLG),
//  DCB=(LRECL=255,BLKSIZE=0,RECFM=FB,DSORG=PS),
//  SPACE=(TRK,(5,1),RLSE)
//*BUILD HSEND BACKDS COMMANDS.
//*****************************************************************
//*THE INCLUDE STATEMENT SELECTS MCB RECORDS THAT CONTAIN
//*THE OUTREC STATEMENT BUILDS AN HSEND BACKDS COMMAND
//*FOR EACH DATA SET FOUND.
//*****************************************************************
//SYSIN DD *
 SORT FIELDS=(COPY)
  OUTFIL FNAMES=CMDS,
    INCLUDE=((51,1,BI,EQ,X'20'),AND,
            (69,6,CH,EQ,C'?VLSER')),CONVERT,
    OUTREC=(C' HSEND BACKDS ',5,44,255:X)
/*
```

11. Delete all lost migrated data sets by issuing the HDELETE commands that were generated in step 8 on page 599. This step corrects the MCDS so it no longer lists the lost migration data sets as residing on the ML1 volume, and deletes the catalog entries for the data sets.

12. Recover all lost migrated data sets by issuing the HRECOVER commands that were generated in step 8 on page 599. This step recovers the migrated data sets that were lost on the damaged ML1 from backup copies. The data sets are recovered to a level-0 volume.

13. Delete all backup versions that were lost by issuing the HSEND BDELETE commands that were generated in step 9 on page 600. This step corrects the BCDS so it no longer lists the lost backup version data sets as residing on the ML1 volume.

14. Identify all lost VTOC-copy data sets.

    ISMF can be used to easily identify which VTOC-copy data sets are lost. The data sets with the following message are no longer on the ML1 volume:

    `DATA SET NOT FOUND ON VOLUME: volser`

    The backup VTOC-copy data set is named as below:

    bprefix.VTOC.T*ssmmhh*.V*volser*.D*yyddd*

    The dump VTOC-copy data set is named as below:

    bprefix.DUMPVTOC.T*ssmmhh*.V*volser*.D*yyddd*

15. Delete the catalog entries for those VTOC-copy data sets that no longer exist on the ML1 volume.

16. Reinitialize the ML1 volume. This will delete all residual data.

17. Define an SDSP data set, if you want one on the ML1 volume.
    a. If you had an SDSP data set on the damaged volume, you need to delete the catalog entry for that data set by issuing a DELETE NOSCRATCH command.
    b. Issue the IDCAMS DEFINE command to define a new SDSP data set for the ML1 volume.

18. Change the ML1 volume so it can be selected for migration output by issuing the ADDVOL command with the NODRAIN parameter. For example, issue this command:

    ```
    ADDVOL volser UNIT(unittype) MIGRATION(ML1 NODRAIN)
    ```

    Be sure to specify other parameters as needed, such as SDSP and AUTODUMP.

Case 9: Reestablish Access to Previously Deleted Migrated Data Sets (No Backup Exists, ML2 Only)

To reestablish access to previously deleted migrated data sets, perform the following steps:

1. Create a catalog entry for the data set. Use the IDCAMS DEFINE NONVSAM command with the VOLSER(MIGRAT) and UNIT(*tapeunit*) parameters.
   a. If the data set is NON-VSAM, run IDCAMS DEFINE data set to VOLSER of MIGRAT to create the catalog entry.

      Example JCL:

```
//IDCAMS   JOB ,CLASS=A,MSGCLASS=H
//STEP   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE NONVSAM -
             (NAME(dsname) -
             DEVT(device type) -
             VOL(MIGRAT))
/*
```

   b. If the data set is VSAM, use IDCAMS DEFINE base and data component to VOLSER of MIGRAT to create the catalog entry.

      Example JCL:

```
//IDCAMS   JOB ,CLASS=A,MSGCLASS=H
//STEP   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE NONVSAM -
              (NAME(xxxxx.ksds) -
              DEVT(device type) -
              VOL(MIGRAT))
  DEFINE NONVSAM -
              (NAME(xxxxx.ksds.data) -
              DEVT(device type) -
              VOL(MIGRAT))
/*
```

Note: For a VSAM data set, define the NONVSAM entry for the base and data component.

---

2. The MCD record must be altered, if it exists, or created, if it does not exist.
   a. Use the following command to display the MCD record, if it exists:

      ```
      FIXCDS D data.set.name DISPLAY
      ```

      Then issue the ASSIGNEDBIT command to turn on the bit, which indicates that the data set has migrated:

      ```
      FIXCDS D data.set.name ASSIGNEDBIT(ON)
      ```

   b. If the MCD record does not exist (the FIXCDS DISPLAY command failed), the following AUDIT command will create it. The ML2 volser is needed for the audit command. If you do not know the volser, check in one of your journal backup versions to locate the journal entry of the MCD record for the migrated data set. Note the migration volume serial number, which is found at offset X'40' in the MCD record. Run the following command with the volser parameter of the ML2 volume:

      ```
      AUDIT MEDCTL VOLUMES(ML2volser) FIX SERIALIZATION(CONTINUOUS)
      ```

   c. For VSAM data sets, if AUDIT issues ERR169 with the data component name, then the MCD was created with the data component name instead of the base cluster name. As explained in the ERR169 message, change this name to the base cluster name before you perform a recall. Issue the following command to change it:

      ```
      FIXCDS D data.component.name NEWKEY(base.name)
      ```

---

3. If the data set spans across tape volumes, you must repeat the AUDIT MEDCTL command for each ML2 volser and issue the following FIXCDS commands for the MCD record:
   a. Expand the MCD record:

      ```
      FIXCDS D data.set.name EXPAND(z)
      ```

      where z = number of volumes to which the data set spans $\times$ 6

   Example: Data set starts on volume **A** and spans to **B** and **C**; $z = 2 \times 6 = 12$. (2 = additional volumes, 6 = 6 characters for each *volser*)

   b. Add additional volsers to the MCDAVSN array at offset X'194'

      ```
      FIXCDS D data.set.name PATCH(X'194' volserxvolsery)
      ```

      where *volserx* = the first additional volume and *volsery* equals the second additional volume.

c. Alter MCDNVSN with the number of additional tape volumes and MCDNVSNO with the offset of the list of volsers:

```
FIXCDS D data.set.name PATCH(X'D4' X'zzzz0194')
```

where *zzzz* = the number of *volsers* and 0194 equals the offset to the list of volsers as determined in the note below the following example.

Example:

```
FIXCDS D data.set.name PATCH(X'D4'X'00020194')
```

Note: To determine if a data set has spanned to additional volumes, inspect the MCD record found in the journal backup. The number of "spanned to" volumes is found in a two-byte field starting at offset X'D4'. The location of the list of "spanned to" volumes is found at offset X'D6'. Offsets are given from the beginning of the MCD record. The MCD record found in the backup copy of the journal is appended at the beginning by a 10 byte MLOG, a 44 byte MCK, and a 20 byte MCH header record.

4. Recall the data set.

5. If you follow all steps correctly and you are still unable to recall the data set, contact IBM Support.

Case 10: Restoring Stacked Output Tapes with a Down-Level MVS Image

The DFSMShsm ABARS file stacking function, which was introduced with DFSMS/MVS Version 1 Release 4, allows users to place (stack) the ABARS ABACKUP output files from a single aggregate group on a minimum number of tape cartridges. The output tapes are intended to be recovered by ARECOVER with an MVS image running DFSMS/MVS Version 1 Release 4, or a subsequent release.

If, for some reason, you need to ARECOVER the stacked output tapes with an MVS image running DFSMS/MVS Version 1 Release 3, use the following procedure:

1. Ensure that the ABACKUP output files are cataloged at the recovery site.

   Either bring your catalogs from the ABACKUP site, or manually catalog the files. This is required for the ARECOVER processing to work. Although this takes some work, it is much quicker than doing a DFSMSdss COPYDUMP or a REPRO.

2. Issue an ARECOVER command for the stacked output tapes specifying the DATASETNAME and XMIT parameters.

   The XMIT parameter causes ABARS to get the ABACKUP output file information out of the catalog and pick up the appropriate volser and file sequence number from the catalog. This enables DFSMShsm running at the DFSMS/MVS Version 1 Release 3 level to recover the stacked output.

3. Recall the data set.

Case 11: Correcting Errors within the Common Recall Queue

The following indicate the corruption of the common recall queue (CRQ):

- DFSMShsm issues message ARC1506E.
- DFSMShsm issues message ARC1187E.
- DFSMShsm does not select recall requests for processing.

  For this last circumstance, issue the QUERY ACTIVE command and examine message ARC1541I. This message displays the factors that affect the selection of requests from the CRQ. A status of anything other than CONNECTED or a hold level of anything other than NONE is a probable reason why certain recall requests are not selected.

Note: An unexpected loss in connectivity to the CRQ may also introduce errors, but in most cases DFSMShsm will be able to automatically correct those errors. When this occurs, you may see the multiple issuance of message ARC1102I. DFSMShsm issues message ARC1102I when it attempts to recall a data set that was previously recalled. Occurrences of this message at the time of a loss in connectivity is normal and not a problem with the CRQ list structure.

If you believe that the CRQ has become corrupted, issue the AUDIT COMMONQUEUE(RECALL) FIX command. When DFSMShsm issues this command, it scans the entries in the CRQ and attempts to correct logical inconsistencies within the structure. Note that the audit function cannot correct all types of errors and may report zero errors, even though there are errors in the structure.

If messages ARC1506E and ARC1187E recur, or if certain data sets are not selected for recall processing after running the audit function, then perform the following procedure:

1. Capture a dump of the CRQ structure.
   a. As a general rule for dumps, include COUPLE and XESDATA information in SDUMPs.

      To display the current options, issue the following command:

      ```
      D D,O
      ```

      If DFSMShsm does not list COUPLE or XESDATA as options for SDUMP, then issue the following command:

      ```
      CD SET,SDUMP=(COUPLE,XESDATA)
      ```

   b. Issue this dump command:

      ```
      DUMP COMM=(CRQ LIST STRUCTURE)
      ```

      See: * *id* IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

   c. Issue the following command:

      ```
      R id,STRLIST=(STRNAME=SYSARC_basename_RCL,(LNUM=ALL,ADJ=CAP,EDATA=UNSER),CONT
      ```

      where *basename* is the base name of the CRQ structure that the following command specified:

      ```
      SETSYS COMMONQUEUE(RECALL(CONNECT(basename)))
      ```

      See: * *id*2 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

   d. Issue the following command:

      ```
      R id2,LOCKE,(EMC=ALL),ACC=NOLIM),END
      ```

---

2. Reallocate the structure to remove the errors.
   a. Issue the SETSYS COMMONQUEUE(RECALL(DISC)) command on all DFSMShsm hosts that are connected to the CRQ structure. This will move all recall requests from the CRQ back to the local DFSMShsm hosts for processing. Each host will issue message ARC1502I after it disconnects from the structure.

      The errors within the CRQ may prevent a particular DFSMShsm host from disconnecting. If this is the case, DFSMShsm cannot perform the remainder of the steps until that DFSMShsm host shuts down.

   Recommendation: Discontinue using the CRQ until the DFSMShsm host can be shut down with minimal impact on other DFSMShsm functions. At this point, no further action is required to prevent DFSMShsm from attempting to place new requests onto the CRQ.

   b. Once all DFSMShsm hosts disconnect from the common recall structure, delete the structure.

      To determine if all DFSMShsm hosts have disconnected, issue the following command:

      ```
      D XCF,STR,STRNAME=SYSARC_basename_RCL
      ```

      The number of connections reported should be zero.

      To delete the structure, issue the following command:

      ```
      SETXCF FORCE,STR,STRNAME=SYSARC_basename_RCL
      ```

   c. Issue the following command on each DFSMShsm host that was previously connected:

      ```
      SETSYS COMMONQUEUE(RECALL(CONNECT(basename)))
      ```

      This command will cause each host to automatically move all of its local recall requests back onto the CRQ and to reallocate the structure.

---

3. Capture the Problem Determination Aid (PDA) data.

   Refer to the *z/OS DFSMShsm Diagnosis Guide and Reference* for information on how to perform this action. The collected PDA includes the data from the time that a problem first occurred, through the time that the audit was performed.

---

4. Contact IBM Support to report the problem.

Case 12: Recovering a Deleted ML1 Data Set without a Backup

This procedure assumes that you have a DUMP of the ML1 volume and that you know when the ML1 data set was deleted. You must determine the ML1 volume and the HSM-generated name from either the journal or activity log. This scenario does not account for data sets in an SDSP on a ML1 volume.

To reestablish access to previously deleted migrated data sets, perform the following steps:

1. Determine the ML1 volume and HSM-generated name from the journal. Use the following jobs to obtain this information:
   a. Copy the journal using IEBGENER before you use SORT.

```
//COPY  JOB ,
//
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,
//          DSN=journal name (the journal data set where the information
//                            about the deleted migrated data set resides)
//SYSUT2 DD DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=6556,BLKSIZE=13112,RECFM=VB),
//          SPACE=(CYL,(10,5),RLSE),
//          DSN= copied journal name
/*
```

b. Process the SORT job.

```
//SORTANY  JOB ,
//
//SORT1   EXEC PGM=SORT
//SYSPRINT DD  SYSOUT=*
//SORKWK01 DD  UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORKWK02 DD  UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTIN   DD  DISP=SHR,
//             DSN=copied journal name
//SORTOUT  DD  DISP=(NEW,CATLG),UNIT=SYSDA,
//*            DCB=(LRECL=121,BLKSIZE=27951,RECFM=FB),
//*            DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//             SPACE=(CYL,(500,5),RLSE),
//             DSN=output name
//MSGOUT1  DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  *
     OPTION VLSHRT
     MERGE FIELDS=COPY
     RECORD TYPE=V
     INCLUDE COND=((63,1,BI,EQ,X'00'),and,(17,len,CH,EQ,C'dsn'))
     END
/*
```

Substitute the length of the user data set name for *len* and the data set name for *dsn*. This job will retrieve the MCD records for the data set that was erroneously deleted.

You will have the journal records for the deleted data set in the sorted output data set. The ML1 volume can be determined by going to +76 into the journal record. The HSM-generated name will be at +232. Record the volser of the ML1 volume and the HSM-generated name.

---

2. Create a NON-VSAM catalog entry that specifies a DEVT(*device type*) and VOL(MIGRAT) where device type is the device type of the ML1 volume.

```
//IDCAMS   JOB ,CLASS=A,MSGCLASS=H
//STEP   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
   DEFINE NONVSAM -
             (NAME(dsname) -
             DEVT(device type) -
             VOL(MIGRAT))
/*
```

Note: For VSAM data sets, you must create a NON-VSAM entry for both the base and data components.

---

3. Run the following command to determine which dump volumes you should use:

   ```
   LIST VOLUME(ml1volser) BCDS ALLDUMPS
   ```

   This will list all of the tape volumes that this ML1 volume was dumped to, and when.

---

4. Restore the data set. Use the list of volumes created by Step 3 as the source volumes to a DFSMSdss RESTORE job specifying the DFSMShsm-generated name in DATASET(INCL(dsn)).

---

5. Run the following command to rebuild the MCD and MCA records for the volume:

   ```
   AUDIT MEDIACONTROLS VOLUMES(ml1volser) FIX
   ```

---

6. RECALL the data set.

Appendix A. Consolidated Commands, Storage Group Definitions, and Management Classes for the Example System

This appendix presents the consolidation of all of the DFSMShsm commands, storage group attributes, and management class attributes that were used to show the automatic storage management of the example system. The examples are separated by the kind of command (ADDVOL, DEFINE, SETSYS, and so forth) and by storage group attributes and management class attributes.

ADDVOL Command

The ADDVOL command is used to define DFSMShsm-owned volumes and non-SMS-managed volumes to DFSMShsm. It is not used for SMS-managed volumes.

Note: A maximum of eight processing units can ADDVOL a primary or migration volume at the same time.

The consolidated ADDVOL command for the example system is:

```
/* DEFINING ML1 VOLUMES TO DFSMShsm IN ALL PROCESSING UNITS */
   ADDVOL ML1001 UNIT(3390) -
     MIGRATION(MIGRATIONLEVEL1 -
     SMALLDATASETPACKING) THRESHOLD(85)

   ADDVOL ML1002 UNIT(3390) -
     MIGRATION(MIGRATIONLEVEL1 -
     SMALLDATASETPACKING) THRESHOLD(90)

   ADDVOL ML1003 UNIT(3390) -
     MIGRATION(MIGRATIONLEVEL1 -
     NOSMALLDATASETPACKING) THRESHOLD(90)

 /* DEFINING AUTOMATIC DUMP OF ML1 VOLUMES IN PROCESSING */
 /* UNIT 2 */
   ADDVOL ML1001 UNIT(3390) MIGRATION(AUTODUMP -
     (ONEWEEK,TWOWEEK))

   ADDVOL ML1002 UNIT(3390) MIGRATION(AUTODUMP -
     (ONEWEEK,TWOWEEK))

   ADDVOL ML1003 UNIT(3390) MIGRATION(AUTODUMP -
     (ONEWEEK,TWOWEEK))

 /* DEFINING NOAUTOMATIC DUMP OF ML1 VOLUMES IN PROCESSING */
 /* UNITS 1 AND 3 */
   ADDVOL ML1001 UNIT(3390) MIGRATION(NOAUTODUMP)

   ADDVOL ML1002 UNIT(3390) MIGRATION(NOAUTODUMP)

 /* DEFINING PRIMARY VOLUMES FOR MIGRATION AND BACKUP */
 /* IN PROCESSING UNIT 1 */
   ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
     AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

   ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
     AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

   ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
     AUTORECALL MIGRATE(12)) THRESHOLD(95 80)
```

© Copyright IBM Corp. 1984, 2004

```
  ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(96 80)

  ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

/* DEFINING PRIMARY VOLUMES FOR MIGRATION AND BACKUP */
/* IN PROCESSING UNIT 2 */
  ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

/* DEFINING PRIMARY VOLUMES FOR MIGRATION AND BACKUP */
```

```
/* IN PROCESSING UNIT 3*/
  ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(96 80)

  ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

  ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

/* DEFINING PRIMARY VOLUMES FOR DUMP IN PROCESSING */
/* UNIT 2 */
  ADDVOL GP0001 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL GP0002 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL GP0003 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL UG1001 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL UG1002 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL UG1003 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL UG2001 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL UG2002 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL UG2003 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,TWOWEEK))

  ADDVOL VP0001 UNIT(3390) PRIMARY(AUTODUMP -
```

```
    (ONEWEEK,FOURWEEK))

  ADDVOL VP0002 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,FOURWEEK))

  ADDVOL VP0003 UNIT(3390) PRIMARY(AUTODUMP -
    (ONEWEEK,FOURWEEK))

/* DEFINING PRIMARY VOLUMES FOR DUMP IN PROCESSING */
/* UNITS 1 AND 3 */
  ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTODUMP)

  ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTODUMP)
```

DEFINE Command

The DEFINE command is used to define pools, dump classes, cycles, and migration level 2 key ranges. You need the DEFINE commands for pools in the ARCCMD*xx* member. The definitions of dump classes, cycles, and migration level 2 key ranges are retained in the DFSMShsm control data sets and are not required (although they are useful) in the ARCCMD*xx* member. The DEFINE commands for migration level 2 key ranges should *not* be included in the ARCCMD*xx* member.

The consolidated DEFINE command for the example system is:

```
/* SPECIFYING PRIMARY SPACE MANAGEMENT CYCLE */
/* ONLY FOR PROCESSING UNIT 1 */
  DEFINE PRIMARYSPMGMTCYCLE(YNNNYNN -
    CYCLESTARTDATE(95/09/15))

/* SPECIFYING SECONDARY SPACE MANAGEMENT CYCLE */
/* ONLY FOR PROCESSING UNIT 2 */
  DEFINE SECONDARYSPMGMTCYCLE(YNNNYNN -
    CYCLESTARTDATE(95/09/15))

/* DEFINING POOLS */
  DEFINE POOL(GROUP1 VOLUMES(UG1001 UG1002 UG1003))
  DEFINE POOL(GROUP2 VOLUMES(UG2001 UG2002 UG2003))
  DEFINE VOLUMEPOOL(SAMPLE VOLUMES(VP0001 VP0002 VP0003))

/* DEFINING POOLS FOR AGGREGATE RECOVERY*/
  DEFINE ARPOOL(* ML1VOLS(UG1001 UG1004) L0VOLS(UG2002))

/* DEFINING THE DUMP CYCLE */
  DEFINE DUMPCYCLE(NNNNNNYNNNNNNYNNNNNNYNNNNNNY -
```

```
    CYCLESTARTDATE(95/09/15))

  /* DEFINING THE DUMP CLASSES */
  DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
    DATASETRESTORE FREQUENCY(7) NORESET -
    RETENTIONPERIOD(21) UNIT(ACL) -
    VTOCCOPIES(2))

  DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
    NODATASETRESTORE DISPOSITION('SEND OFFSITE BIWEEKLY') -
    FREQUENCY(14) NORESET RETENTIONPERIOD(42) -
    UNIT(ACL) VTOCCOPIES(2))

  DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
    NODATASETRESTORE DAY(28) -
    DISPOSITION('SEND OFFSITE MONTHLY') NORESET -
    RETENTIONPERIOD(84) UNIT(ACL) -
    VTOCCOPIES(0))

/* SPECIFYING BACKUP CYCLE */
  DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(95/09/15))
```

SETSYS Command

The SETSYS command is used to specify general conditions for DFSMShsm processing.

The consolidated SETSYS command for the example system is:

```
/* SPECIFY THE MVS ENVIRONMENT */
  SETSYS JES2 CSALIMITS(MAXIMUM(100) ACTIVE(90) -
    INACTIVE(30) MWE(4)) NOREQUEST -
    USERDATASETSERIALIZATION NOSWAP

/* SPECIFY THE AGGREGATE BACKUP AND RECOVERY ENVIRONMENT */
  SETSYS MAXABARSADDRESSSPACE(2) ABARSPROCNAME(SAMPLE) -
  NOARECOVERREPLACE -
  ACTLOGTYPE(DASD) ARECOVERUNITNAME(3490)

/* SPECIFY THAT ABARS INSTALLATION-WIDE EXITS BE USED */
  SETSYS EXITON(ARCBEEXT)
  SETSYS EXITON(ARCCREXT)
  SETSYS EXITON(ARCEDEXT)
  SETSYS EXITON(ARCM2EXT)
  SETSYS EXITON(ARCSKEXT)

/* DEFINE DFSMShsm'S SECURITY ENVIRONMENT */
  SETSYS NOACCEPTPSCBUSERID NOERASEONSCRATCH -
    NORACFIND

/* DEFINE DATA FORMATS FOR DFSMShsm'S OWNED DATA SETS */
    SETSYS COMPACT(ALL) COMPACTPERCENT(30) -
    OBJECTNAMES(OBJECT,LINKLIB) -
    SOURCENAMES(ASM,PROJECT) OPTIMUMDASDBLOCKING -
    CONVERSION(REBLOCKTOANY) EXPORTESDS(CIMODE)

/* DEFINE DFSMShsm REPORTING AND MONITORING */
  SETSYS ACTLOGMSGLVL(EXCEPTIONONLY) -
    ACTLOGTYPE(DASD) MONITOR -
    (BACKUPCONTROLDATASET(80) JOURNAL(80) -
    MIGRATIONCONTROLDATASET(80) NOSPACE -
    OFFLINECONTROLDATASET(80) NOSTARTUP -
    NOVOLUME) SYSOUT(A 1) -
    NOSMF

/* DEFINE DFSMShsm'S TAPE ENVIRONMENT */
```

```
  SETSYS INPUTTAPEALLOCATION(NOWAIT) DEFERMOUNT -
    TAPEUTILIZATION(UNITTYPE(3490) PERCENTFULL(90) MOUNTWAITTIME(15) -
    OUTPUTTAPEALLOCATION(NOWAIT) -
    USERUNITTABLE(ACL) SELECTVOLUME(SCRATCH) -
    RECYCLETAPEALLOCATION(NOWAIT) -
    TAPESECURITY(RACF) -
    TAPEDELETION(SCRATCHTAPE) -
    TAPEHARDWARECOMPACT -
    PARTIALTAPE(MARKFUL)

/* SPECIFY THE DFSMShsm INSTALLATION-WIDE EXITS THAT ARE USED */
  SETSYS EXITON(CD)

/* SPECIFY DFSMShsm'S RECOVERABILITY */
  SETSYS JOURNAL(RECOVERY)

/* SPECIFYING WHEN TO START PRIMARY SPACE MANAGEMENT */
/* PROCESSING UNIT 1 */
  SETSYS PRIMARYSPMGMTSTART(0200 0400)

/* SPECIFYING WHEN TO START SECONDARY SPACE MANAGEMENT */
/* PROCESSING UNIT 2 */
  SETSYS SECONDARYSPMGMTSTART(0030 0200)

/* SPECIFYING INTERVAL MIGRATION */
/* PROCESSING UNIT 2 */
  SETSYS INTERVALMIGRATION

/* SPECIFYING NOINTERVAL MIGRATION */
/* PROCESSING UNITS 1 AND 3 */
  SETSYS NOINTERVALMIGRATION
/* SPECIFYING MAXIMUM AUTOMATIC MIGRATION TASKS */
/* PROCESSING UNITS 1 AND 2 */
  SETSYS MAXMIGRATIONTASKS(5)
/* SPECIFYING LEVEL 2 MIGRATION */
  SETSYS TAPEMIGRATION(ML2TAPE(TAPE(ACL)))

/* SPECIFYING EXTENT REDUCTION */
  SETSYS MAXEXTENTS(6)

/* SPECIFYING SCRATCHING EXPIRED DATA SETS */
  SETSYS EXPIREDDATASETS(SCRATCH)

/* SPECIFYING RECALL TASKS*/
  SETSYS MAXRECALLTASKS(15) -
    TAPEMAXRECALLTASKS(6)

/* SPECIFYING SMALL DATA SET PACKING */
  SETSYS SMALLDATASETPACKING(KB(110))

/* SPECIFYING MIGRATED DATA SET NAME PREFIX */
  SETSYS MIGRATEPREFIX(HSM)

/* SPECIFYING MIGRATION CLEANUP */
/* PROCESSING UNIT 1 ONLY */
  SETSYS MIGRATIONCLEANUPDAYS(25 10)

/* SPECIFYING AGE FOR SCRATCHING LIST */
/* DATA SETS */
  SETSYS SCRATCHFREQUENCY(5)

/* SPECIFYING A MINIMUM AGE FOR SPACE MANAGEMENT */
/* PROCESSING (NON-SMS-MANAGED DATA SETS) */
  SETSYS DAYS(0)

/* SPECIFYING INACTIVE AGE FOR ELIGIBILITY FOR */
/* MIGRATION FROM MIGRATION LEVEL 1 */
```

```
  SETSYS MIGRATIONLEVEL1DAYS(60)

/* SPECIFYING KIND OF VOLUMES TO USE FOR RECALL */
  SETSYS RECALL(ANYSTORAGEVOLUME(LIKE))

/* DEFINING WHEN TO START AUTOMATIC DUMP */
  SETSYS AUTODUMPSTART(0300 0400 0500)

/* SPECIFYING THE BUFFERING TECHNIQUE */
  SETSYS DUMPIO(3,1)

/* SPECIFYING THE MAXIMUM NUMBER OF DUMP TASKS */
  SETSYS MAXDUMPTASKS(3)

/* SPECIFYING WHEN AUTOMATIC BACKUP IS TO START (PROCESSING UNIT 1)*/
  SETSYS AUTOBACKUPSTART(0300 0400 0500)

/* SPECIFYING WHEN AUTOMATIC BACKUP IS TO START */
/* (PROCESSING UNITS 2 AND 3) */
  SETSYS AUTOBACKUPSTART(0330 0400 0500)

/* SPECIFYING TO DIRECT BACKUP TO TAPE, AND TO RETRY BACKUP */
/*  WHEN A DATA SET IS IN USE.                              */
 SETSYS BACKUP(TAPE(ACL) INUSE(RETRY(Y) DELAY(10) +
                         SERIALIZATION(PREFERRED)))

/* SPECIFYING FIRST QUALIFIER FOR NAMES OF BACKED UP DATA SETS */
  SETSYS BACKUPPREFIX(HSM)

/* SPECIFYING MAXIMUM NUMBER OF BACKUP TASKS */
  SETSYS MAXBACKUPTASKS(3)

/* SPECIFYING BACKUP OF RACF DISCRETE DATA SET PROFILES */
  SETSYS NOPROFILEBACKUP

/* SPECIFYING SPILL PROCESSING */
  SETSYS NOSPILL

/* SPECIFYING BACKUP OF ONLY CHANGED DATA SETS */
  SETSYS INCREMENTALBACKUP(CHANGEDONLY)

/* SPECIFYING MINIMUM DAYS BETWEEN BACKUPS */
/* NON-SMS-MANAGED DATA SETS */
  SETSYS FREQUENCY(1)

/* SPECIFYING MAXIMUM NUMBER OF BACKUP VERSIONS */
/* TO KEEP (NON-SMS-MANAGED DATA SETS */
  SETSYS VERSIONS(5)

/* SPECIFYING BACKUP OF CONTROL DATA SETS */
  SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(4) -
    BACKUPDEVICECATEGORY(TAPE) BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
    MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
    OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
    JRNLBACKUPDSN(BHSM.JRNL.BACKUP))

/* SPECIFYING RECYCLE PERCENTAGES AND OUTPUT DEVICES */
  SETSYS ML2RECYCLEPERCENT(25) RECYCLEPERCENT(25) -
     RECYCLEOUTPUT(BACKUP(ACL),MIGRATION(ACL)) -
     MAXRECYCLETASKS(3) ML2PARTIALSNOTASSOCIATEDGOAL(8)
```

Storage Groups

Figure 75 on page 618 shows the storage group attributes used to control DFSMShsm management of storage at a volume level.

```
NAME ===> SG1
  MIGRATE SYSTEM/SYSPLEX NAME ===>
  AUTO MIGRATE ===> Y
  ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80

  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> TWOWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG2
  MIGRATE SYSTEM/SYSPLEX NAME ===>
  AUTO MIGRATE ===> Y
  ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80

  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> TWOWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG3
  MIGRATE SYSTEM/SYSPLEX NAME ===>
  AUTO MIGRATE ===> Y
  ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80

  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> FOURWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME ===>

NAME ===> SG4
  MIGRATE SYSTEM/SYSPLEX NAME ===>
  AUTO MIGRATE ===> Y
  ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 95  LOW ===> 80

  AUTO DUMP   ===> Y
  DUMP CLASS  ===> ONEWEEK
  DUMP CLASS  ===> FOURWEEK
  DUMP SYSTEM/SYSPLEX NAME ===>

  AUTO BACKUP ===> Y
  BACKUP SYSTEM/SYSPLEX NAME===>
```

Figure 75. Consolidated Storage Group Attributes

Management Classes

Figure 76 on page 619 shows the management class attributes used to control DFSMShsm management of storage on a data set basis.

```
NAME ===> STANDMC
  EXPIRE AFTER DAYS NON-USAGE ===> 120
  EXPIRE AFTER DATE/DAYS      ===> 360
  PARTIAL RELEASE             ===> Y
  PRIMARY DAYS NON-USAGE      ===> 10
  LEVEL 1 DAYS NON-USAGE      ===> 60
  COMMAND OR AUTO MIGRATE     ===> BOTH

  BACKUP FREQUENCY                   ===> 3
  NUMBER OF BACKUP VERSIONS          ===> 3
    (DATA SET EXISTS)
  NUMBER OF BACKUP VERSIONS          ===> 1
    (DATA SET DELETED)
  RETAIN DAYS ONLY BACKUP VERSION    ===> 60
    (DATA SET DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS  ===> 20
  ADMIN OR USER COMMAND BACKUP       ===> BOTH
  AUTO BACKUP                        ===> Y

NAME ===> GDSMC
  EXPIRE AFTER DAYS NON-USAGE ===> 7
  EXPIRE AFTER DATE/DAYS      ===> 90
  PARTIAL RELEASE             ===> N
  PRIMARY DAYS NON-USAGE      ===> 5
  LEVEL 1 DAYS NON-USAGE      ===> NOLIMIT
  COMMAND OR AUTO MIGRATE     ===> BOTH
  # GDG ELEMENTS ON PRIMARY   ===> 2
  ROLLED-OFF GDS ACTION       ===> EXPIRE

  BACKUP FREQUENCY                   ===> 1
  NUMBER OF BACKUP VERSIONS          ===> 1
    (DATA SET EXISTS)
  NUMBER OF BACKUP VERSIONS          ===> 1
    (DATA SET DELETED)
  RETAIN DAYS ONLY BACKUP VERSION    ===> 14
    (DATA SET DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS  ===>
  ADMIN OR USER COMMAND BACKUP       ===> ADMIN
  AUTO BACKUP                        ===> Y
```

Figure 76. Consolidated Management Classes (Part 1 of 2)

```
NAME ===> LARGEMC
  EXPIRE AFTER DAYS NON-USAGE ===> 90
  EXPIRE AFTER DATE/DAYS      ===> 180
  PARTIAL RELEASE             ===> Y
  PRIMARY DAYS NON-USAGE      ===> 15
  LEVEL 1 DAYS NON-USAGE      ===> 0
  COMMAND OR AUTO MIGRATE     ===> BOTH

  BACKUP FREQUENCY                  ===> 2
  NUMBER OF BACKUP VERSIONS         ===> 4
    (DATA SET EXISTS)
  NUMBER OF BACKUP VERSIONS         ===> 1
    (DATA SET DELETED)
  RETAIN DAYS ONLY BACKUP VERSION   ===> 60
    (DATA SET DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS ===> 20
  ADMIN OR USER COMMAND BACKUP      ===> ADMIN
  AUTO BACKUP                       ===> Y

NAME ===> NOMIGMC
  EXPIRE AFTER DAYS NON-USAGE ===> 20
  EXPIRE AFTER DATE/DAYS      ===> NOLIMIT
  PARTIAL RELEASE             ===> Y
  PRIMARY DAYS NON-USAGE      ===> NOLIMIT
  LEVEL 1 DAYS NON-USAGE      ===> NOLIMIT
  COMMAND OR AUTO MIGRATE     ===> NONE

  BACKUP FREQUENCY                  ===> 1
  NUMBER OF BACKUP VERSIONS         ===> 1
    (DATA SET EXISTS)
  NUMBER OF BACKUP VERSIONS         ===> 1
    (DATA SET DELETED)
  RETAIN DAYS ONLY BACKUP VERSION   ===> 30
    (DATA SET DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS ===>
  ADMIN OR USER COMMAND BACKUP      ===> BOTH
  AUTO BACKUP                       ===> N
```

Figure 76. Consolidated Management Classes (Part 2 of 2)

Appendix B. Preventive Methods for Avoiding Data Loss

This appendix presents methods for avoiding data loss.

Steps for Preventing a Full Journal (Damaged CDS)

To prevent a full journal, perform the following steps:

1. Ensure that no reorganization occurs if DFSMShsm is running on any image in the complex. You must do two things to accomplish this.
   - Make sure that SYSDSN enqueues are in the system's inclusion list for global resource serialization (GRS). This allows the SYSDSN enqueues to be passed to all images in the complex that share the CDSs.
   - Modify the JCL used for reorganization of the CDSs by using a disposition of OLD whenever a DD statement refers to a CDS.
2. Define the DFSMShsm CDSs with VSAM SHAREOPTIONS(2 3). You can find information about defining VSAM SHAREOPTIONS for DFSMShsm in the *z/OS DFSMShsm Implementation and Customization Guide* in "VSAM SHAREOPTIONS for Control Data Sets," and "Control Data Set Considerations."
3. Implement a procedure where the backup of the CDSs is closely monitored.

   One method of monitoring the backups of the CDSs is to review the CDS backup and export messages in the DFSMShsm backup activity log. Specifically, review the number of records processed by the export function in message IDC0005I, for each CDS. Compare the record count of the current CDS backup with the record count from the preceding days. The record count may not be the same, but it should be close.

   Normal update and deletion activity causes some fluctuation in the number of records in the CDSs. However, when a large decrease in records is detected, you should be able to account for the large difference.

   Some large fluctuations in the number of CDS records are normal, and result from processes that reduce the number of records in a CDS. For example, running the EXPIREBV command deletes records in the BCDS. DFSMShsm secondary space management deletes records in the MCDS. However, some large fluctuations in the number of records in a CDS can indicate that the CDS has been damaged. For example, in a multiple-MVS-image environment, the reorganization of a CDS occurring on one image while DFSMShsm is running on another image can damage the structure of the CDS, and cause a large fluctuation in CDS records.

   To eliminate any doubt about the soundness of the CDS structure, run the IDCAMS EXAMINE command against the CDS in question. Running the EXAMINE command against your CDSs every day minimizes the chance of CDS damage going undetected.

Preventing a Full Journal (Undamaged CDS)

To help prevent the problem of a full journal, ensure that your journal is of adequate size to record all the activity between backups of the CDSs. In conjunction with that, schedule CDS backups frequently enough so that the journal can handle the activity. Most installations run automatic backup once a day, seven days a week; the journal data set therefore needs to be large enough to contain data for a 24-hour time period.

© Copyright IBM Corp. 1984, 2004

Also use the SETSYS MONITOR (JOURNAL) command to send your warnings when the journal data set is getting full. When the journal data set is filled to the specified threshold, which you can set to any percentage from 0 to 100 or leave at the DFSMShsm default value of 80%, a warning message is issued to the system operator. Messages continue to be issued until the journal data set is 100% full. Set your threshold at a value that gives you adequate time to back up your CDSs and journal and to allocate a new journal.

Customers that have NetView or another automation product can monitor the ARC0909E and ARC0026E messages, keying in on the 100% and EOV message variables. When 100% and EOV are issued, NetView can automatically issue the BACKVOL CDS command.

Preventing Structural Damage to CDS and Missing Journal Records

Perform the following steps to prevent structural damage to CDS and missing journal records:

1. Schedule reorganizations of CDSs to occur only when necessary. If the CDS is of adequate size, you should not have to reorganize often.
2. Use the proper procedures to reorganize your CDSs.
3. Ensure that no reorganization can occur when DFSMShsm is running on any MVS image in the complex. To do this:
   - Make sure that SYSDSN enqueues are in the system's inclusion list for global resource serialization (GRS); this allows the SYSDSN enqueues to be passed to all images in the complex that share the CDSs.
   - Modify the JCL used for reorganization of the CDSs by using a disposition of OLD whenever a DD statement refers to a CDS.
4. Define the DFSMShsm CDSs with VSAM SHAREOPTIONS(2 3).
5. Use the IDCAMS EXAMINE command to do preventive examines.
6. Reassess the number of backup versions you keep for the CDSs and journal. The more backup versions you have, the greater your chances are of having a good CDS and journal backup version to use should the need arise.

Related reading:

- For additional information about the procedures used to reorganize your CDs, see *z/OS DFSMShsm Implementation and Customization Guide* under the heading "Reorganizing the Control Data Sets."
- For additional information about defining VSAM SHAREOPTIONS for DFSMShsm see the *z/OS DFSMShsm Implementation and Customization Guide* in "VSAM SHAREOPTIONS for Control Data Sets," and "Control Data Set Considerations."

Preventing Time Sequence Gaps in CDS Records (Journal Records Also Missing)

To prevent time sequence gaps in CDS records, perform the following steps:

1. Call on IDCAMS REPRO to create an output data set, check for a successful completion code on the creation of the output data set, and create an input data set.

   If a successful completion code is not received for the creation of the output data set, the job ends, and the input data set is not created.

2. Make CDS data sets large enough that the need for a reorganization is infrequent.

   If your CDS is currently too small and you want to change that, the next time you reorganize the data set, add 10 cylinders. If the data set is still too small, the next time you reorganize add another 10 cylinders. Do this until the data set is large enough that the need for reorganizations is infrequent.
3. Do a backup of the CDS before every reorganization, and keep those backup tapes until it is time for the next reorganization.
4. Reorganize your CDSs by utilizing the proper procedures.
5. Protect DFSMShsm's resources with RACF or a comparable product.

Related reading: For additional information about reorganizing your control data sets, see *z/OS DFSMShsm Implementation and Customization Guide* under the heading "Reorganizing the Control Data Sets."

Preventing Overwritten Migration Tape or Overwritten Backup Tape

Perform the following steps to avoid overwriting migration tape or backup tape:

1. Protect DFSMShsm's resources with RACF or a comparable product.
2. Protect your alternate volumes if your tape management system does not do it for you. DFSMSrmm protects your alternate tape volumes. Some products, however, do not.

Prepare for Recovery of a Head Crash on an ML1 Volume

To prepare for recovery of a head crash on an ML1 volume, perform the following steps:

1. Use the IDCAMS EXPORT TEMPORARY command to regularly export SDSPs with high activity.

   Note: Be sure to use the TEMPORARY parameter with the EXPORT command so the export function does not delete your original data set.
2. Dump ML1 volumes regularly.

Reestablishing Access to Previously Deleted, Migrated Data Sets (No Backup Exists, ML2 Only)

Perform the following steps to reestablish access to previously deleted, migrated data sets:

1. Keep multiple CDS backup versions so you can recover from situations such as this.
2. Create backup copies of data sets that are important. Without a backup copy, the migration copy of a data set is the only copy you have.
3. Use the DCOLLECT function to track what data exists in active, migrated, or backed up format at the time the DCOLLECT function is run. If you run DCOLLECT daily or a couple of times a week, with output going to a generation data group (GDG), the DCOLLECT information could be a valuable asset in data recovery.

Restoring Stacked Output Tapes with a Down-Level MVS Image

The DFSMShsm default is to STACK the ABACKUP output tapes. If you may need to perform ARECOVER processing with an MVS image running at a release prior to DFSMS/MVS Version 1 Release 4, specify the NOSTACK option when you create your ABACKUP output tapes. This can be specified at the system level with a SETSYS ABARSTAPES(NOSTACK) command, or it can be specified with each ABACKUP using the NOSTACK parameter of the ABACKUP command.

Appendix C. Linux-z/OS DFSMShsm Dump or Restore HOW-TO

This appendix describes how to back up your Linux partitions and volumes. z/OS is the IBM operating system that can run on either the IBM S/390™ processor or the IBM eServer zSeries™ processor. The term mainframe refers to either of these two machines. This appendix applies only to Linux for S/390 and Linux for zSeries.

You can include your Linux volumes in an existing z/OS backup solution that uses DFSMShsm. This backup solution works in two environments: (1) a z/OS focused environment with z/OS running on several LPARs and a few dozen Linux servers running under z/VM in an LPAR; (2) a Linux-focused environment with z/VM running in BASIC mode, with hundreds to thousands of Linux guests, and one or more z/OS guests that will perform the DFSMShsm processing.

LINUX is a registered trademark of Linus Torvalds. You can direct your comments to linux390@de.ibm.com.

To get the latest version of this HOW TO and related utilities, check with your Linux provider.

How DFSMShsm Supports Linux

This appendix provides a discussion of only the supported DFSMShsm functions for Linux volumes.

For Linux volumes, DFSMShsm supports only the dump and recover functions. DFSMShsm dump is a physical copy of DASD tracks to tape. An automatic dump process initiates dump processing on a repeating schedule. After reading this HOW TO, you will know what to do to have your Linux volumes dumped by DFSMShsm.

Note: Linux volumes should not be managed by SMS.

Requirements

This section presents a discussion of several requirements related to using Linux.

Hardware Environment

Linux can run on a mainframe under the following three modes:

BASIC A single operating system that owns the entire mainframe.

LPAR A mainframe divided into logical partitions, with each partition running its own operating system image. There can be up to 15 LPARs.

VIRTUAL VM supports hundreds to thousands of guests. Each guest runs its own operating system—for example, one guest using Linux and another using z/OS, all on the same hardware.

© Copyright IBM Corp. 1984, 2004

Tests were run with Linux kernel 2.4 and OS/390 R10 under the same z/VM system.

DASD Background

The mainframe environment can store data on Direct Access Storage Device (DASD) by utilizing an extended count-key-data (ECKD) protocol. DASD is divided into logical units known as volumes. Volumes divide into fixed-size tracks, and the track's geometry determines its size. Linux supports both the 3380 and 3390 track geometries. You can reference a volume by using either a 16-bit device number or a six-character volume serial number (referred to as the *volser*). More than one operating system can access a volume at the same time. A volume can have a different device number for each operating system that has access to it, but only one volser. Linux allows multiple volumes to have the same volser, while z/OS allows only one volume per volser to be accessible at a time.

Linux Requirements

In order for z/OS to process the Linux volumes, they must be formatted in the compatible disk layout (CDL) using **dasdfmt** version 1.0 or later and partitioned using **fdasd** version 1.0 or later.

Linux Disk Utilities

dasdfmt

The default disk layout for **dasdfmt** is CDL. Volumes in the original format, Linux disk layout (LDL), are not compatible with the z/OS system and cannot be accessed and backed up by z/OS. Below is an example of how to format a disk with **dasdfmt** at address 0198, having a byte block size of 4096, and a *volser* of LNX200:

```
dasdfmt -n 198 -b 4096 -l lnx200
```

A screen appears like the one below:

```
Drive Geometry: 3339 Cylinders * 15 Heads =  50085 Tracks
I am going to format the device 198 in the following way:
    Device number of device : 0x198
    Major number of device  : 94
    Minor number of device  : 8
    Labelling device        : yes
    Disk label              : VOL1
    Disk identifier         : LNX200
    Extent start (trk no)   : 0
    Extent end (trk no)     : 50084
    Compatible Disk Layout  : yes
    Blocksize               : 4096

 --->> ATTENTION! <<--- All data in the specified range of that device
will be lost. Type "yes" to continue, no will leave the disk untouched:
```

You can see a 3339 cylinder volume that is attached to Linux at address 198 in the screen capture above. The volume serial number (*volser*) is LNX200, and its block size is 4096 bytes—the optimal and recommended block size. The *volser* must be six characters in length. The disk label, VOL1, indicates that a z/OS system can process this volume.

An LDL volume has a disk label of LNX1. z/OS cannot process any volume with a disk label of LNX1.

fdasd

After you have formatted a volume with **dasdfmt**, you must partition it before Linux can use it. Use the **fdasd** program to partition the volume. This program is similar to the **fdisk** program that comes with the Linux version that runs on personal computers. One difference is that it creates partitions on ECKD DASD instead of hard drives. With **fdasd**, you can create up to three partitions on a volume. Each partition appears to z/OS as a data set. **fdasd** asks you to choose the beginning track and ending track or size of the partition. The naming conventions for the data sets and partitions are listed in "Data Set and Partition Naming Conventions" on page 638.

Follow these rules when you create partitions with **fdasd** that you want to back up with z/OS tools.

- Create partitions by starting from the lowest possible track.
- Do not leave gaps between partitions.
- If you want to restore a partition, ***do not*** delete it with **fdasd** first. When you delete the partition with ***fdasd***, the naming convention between the partition names and data set names is no longer valid. If you then recover a partition on that volume, you could recover over the wrong partition, losing data. When you restore the partition, you replace the existing partition.
- If you follow the previous rules of thumb above, then you can add partitions to a volume at any time up to a maximum of three partitions or until the entire volume is used. If you need to reorganize the partitions, you should back up the data on the partitions and then delete all partitions and create new ones in the desired configuration.

z/OS Requirements

The tools that IBM supports for backing up Linux volumes are DFSMSdss (Data Facilities Storage Management Subsystem data set services) and DFSMShsm. DFSMSdss and DFSMShsm are elements of z/OS and OS/390 for backing up data and managing disk space. The z/OS image must also be able to attach and vary online the volumes that Linux is using that you want backed up.

Authorization Requirements

You need to have root authority for Linux in order to mount and unmount the partitions, as well as to format and partition the volumes using **dasdfmt** and **fdasd**. For z/OS, you need authority to submit DFSMShsm commands and, optionally, to run ADRDSSU, which is the program that is invoked when using DFSMSdss. z/OS sees the Linux partitions as data sets. You can prevent unauthorized access of the Linux partitions by z/OS applications and users by using the IBM RACF, a component of the Security Server for z/OS.

Setting Up Your Dump Environment for Linux Volumes

A Linux volume is formatted with **dasdfmt** in the CDL format and partitions put on it with **fdasd**. It has a valid VTOC on it and a data set for each Linux partition. A Linux volume holds only Linux partitions. No MVS or VM data sets or any non-Linux data should be allowed on a Linux volume. A Linux volume can hold one to three partitions and can be a 3380 or 3390.

The data sets created by **fdasd** are not added to or removed from the catalog by **fdasd** as they are allocated and deleted. If you manually catalog the data sets, you must uncatalog them if you delete them with **fdasd**. It is recommended that you

leave the data sets uncataloged. The partition and data set naming conventions are listed in “Data Set and Partition Naming Conventions” on page 638.

Performing Only Dumps

DFSMShsm can only dump and recover from dump those Linux volumes that are formatted with **dasdfmt** and partitioned with **fdasd**. All other DFSMShsm functions are unsupported for Linux volumes. No data sets should be migrated off or recalled to Linux volumes. Backup is not supported for Linux volumes or partitions. Use Linux volumes only as PRIMARY volumes and use them with the NOAUTOBACKUP, NOAUTOMIGRATION, and NOAUTORECALL options. For the dump class that you use with Linux, use the option VTOCCOPIES(0). VTOCCOPIES is not supported for Linux volumes.

When Do You Want Your Dumps Performed?

If you already use automatic dump, then a time of day and dump cycle is already in place. If not, then typically the time and day of the week is when the system is not very busy. The daily dump cycle consists of a repeating pattern of up to 31 days where dumps are made or not made. Most DFSMShsm administrators use a weekly or daily dump cycle. You can choose any days where dumps are made to have your volumes dumped; however, some dump days might be busier than others, limiting tape hardware availability on those days. You can also specify how frequently you perform dumps by the minimum number of days that must pass from one dump to the next. If you do not already have automatic dump set up, then tape hardware availability and DASD write activity are the only factors limiting when you can dump your Linux volumes. You can also perform a manual dump of your volumes for special situations.

When scheduling your dumps, you must ensure that no writes to the volumes that you are dumping occur while the dumps are being made. The best way to ensure the data integrity of your partitions and their dumps is to unmount them or to have them mounted as read-only. By unmounting a partition, you flush all buffered data that has not yet been written to the partition.

How Long Do You Want to Keep Your Dumps?

You can specify how many days you want to retain your dumps. You may want to specify how many days you retain your dumps in multiples of how frequently in days that you dump. Then you will always have the same number of dump generations at your disposal.

How Many Dumps Are Stacked on Each Dump Tape?

The capacity of current tape technology allows several DASD volumes to be dumped to one dump tape. Only one dump of a specific volume can exist on a dump tape. Once a dump tape has dumps on it, no new dumps can be added to it on any subsequent dump day. Also, a dump tape holds only dumps from one dump class.

Consideration: If you have data that belongs to customers, you may not want to have data from more than one customer on the same tape. If your customer has a right to request all copies of data, and if the data co-exists on a dump tape with another customer’s data, you will have to copy the data to another tape. Assigning different dump classes to different customers ensures that their data is on separate tapes.

How Many Tape Drives Do You Have?

Your mainframe environment probably has many tape drives, but you probably will not be able to use all of them. To limit the number of tape drives that DFSMShsm uses for dump, use the SETSYS command with the MAXDUMPTASKS option. Once you set the maximum number of dump tasks, you prevent DFSMShsm from taking up too many tape drives and delaying other functions that also require tape drives.

Recovering Your Data

When recovering data, unmount any partitions that are being recovered during the recovery process to ensure data integrity. There are several ways you can recover your data from a DFSMShsm dump. You can recover an entire volume with DFSMShsm or DFSMSdss. You can also recover a partition from a dump with DFSMShsm or DFSMSdss. To use DFSMSdss to process Linux volumes, see the HOW TO appendix in the *z/OS DFSMSdss Storage Administration Guide*. When a volume or partition is recovered, the dump overwrites it; the partition or volume then returns to the state at which the dump was made. All changes since the dump was made are lost. See "Data Set and Partition Naming Conventions" on page 638 on the data set and partition naming conventions to make sure that the right partition is recovered.

There are some limitations in how you can recover to a Linux volume. For a volume of more than 65536 tracks, you can recover only the full volume from a dump; no individual data sets can be recovered to it with DFSMShsm. However, you can recover a data set to a volume over 65536 tracks from a DFSMShsm dump with DFSMSdss. To use DFSMSdss on Linux volumes and partitions, see the HOW-TO appendix in the *z/OS DFSMSdss Storage Administration Guide*. Either a full volume or an individual data set can be recovered to a volume of less than 65536 tracks by DFSMShsm. When you recover a data set or volume with DFSMShsm, you must specify the FROMDUMP(DUMPVOLUME(volser)) option. You can list the dump volumes with the LIST DUMPVOLUME command.

Recovering a Volume with DFSMShsm

To recover a volume with DFSMShsm, you need to know only the volser of the volume you want recovered and from how far back you want your volume recovered. Remember, you can go only as far back as the number of days you have been keeping your dumps. You can go back by date or by the generation of dump where generation 0 is the most recent dump, generation 1 is the second most recent dump, and so on.

The following command recovers volume LNX400 from the third most recent dump.

```
RECOVER * TOVOLUME(LNX400) UNIT(3390) -
FROMDUMP(DUMPGENERATION(2))
```

You can also recover a dump to a volume other than the one that was used to make the dump with DFSMShsm. For instance, you can dump volume LNX400 and recover the dump to volume SPARE5; but in the process, volume SPARE5 is renamed with volume LNX400's volser, which is not a problem for Linux as it is for z/OS. The z/OS operating system cannot have two or more volumes with the same volser online at the same time. The original volume must be varied offline before the recovery is started. You need to know the volsers of both the volume you want recovered and the volume you want the dump recovered to.

The following command recovers the most recent dump of volume LNX400 to volume SPARE5 and changes the volser of volume SPARE5 to LNX400. Volume LNX400 must be varied offline in MVS before this recovery command is issued.

```
RECOVER * TOVOLUME(LNX400) UNIT(3390) TARGETVOLUME(SPARE5) -
FROMDUMP
```

If you do not have an extra volume attached to your Linux environment, you need to shut it down and reconfigure your environment with the extra volume and reboot your Linux environment. You do not have to do this in MVS.

Recovering a Volume with DFSMSdss

To recover a volume from a DFSMShsm dump with DFSMSdss, use the LIST command to find the volser of the dump tape volume that contains the dump you want to recover. See "The DFSMShsm LIST Command" on page 637 on how to use the LIST command. You can search the output data set for the volser of the volume you want to recover. Once you find the dump that you want to recover in the output data set, you will find the volser of the dump volume and the data set name of your dump. If you recover the dump to an alternate volume, DFSMSdss does not change the volser to that of the volume that the dump was created from. You need to submit JCL to use DFSMSdss. Once you have the information that you need from the output data set, you can submit a job like the following to recover your volume.

The following JCL recovers the dump data set DFHSM.DMP.DAILY.VLNX300.D01244.T072510 to volume LNX300 from dump tape A02001.

```
//LNXJOB JOB ,'IBMUSER',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS  EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=A02001,DISP=OLD,
               DSN=DFHSM.DMP.DAILY.VLNX300.D01244.T072510
//TARGET   DD UNIT=3390,VOL=SER=LNX300,DISP=OLD
//SYSIN    DD *
 RESTORE FULL INDDNAME(SOURCE) OUTDDNAME(TARGET) PURGE
/*
```

Recovering a Partition with DFSMShsm

When you recover a partition, leave the original damaged partition in place. If you do not still have the original partition or you want to recover the partition to another volume, then you first must preallocate a partition of the exact same number of tracks with **fdasd**. You must know the volser of the volume that contains the partition you want to recover, which partition to recover, and how far back you want to recover from. If you had to preallocate a partition, then you also need to know the volser of the preallocated partition and which partition you preallocated. The preallocated partition does not need a file system.

If a Linux user does not follow the rules of thumb in "fdasd" on page 627, then the data set name of one or more of the partitions could be changed. The **fdasd** utility initially creates the data sets in order where PART0001 is always the data set that has the lowest starting track, PART0002 has the next lowest starting track, and PART0003 has the highest starting track. Once a partition is removed with **fdasd**,

any other changes made after the first removal may cause the order and naming not to be valid. See the naming conventions in “Naming Conventions” on page 638.

You also need to know the volser of the dump tape that contains the dump from which you want to recover. To find the volser of the dump tape, use the LIST command. See “The DFSMShsm LIST Command” on page 637 on how to use the LIST command.

The following command recovers data set LINUX.VLNX300.PART0001.NATIVE on volume LNX300 from the dump of LNX300 on dump tape LX0001. If the dump of volume LNX300 was the only dump on the dump tape, omit the SOURCEVOLUME parameter, but if dumps from two or more volumes are stacked on the dump tape, then you must specify SOURCEVOLUME.

```
RECOVER LINUX.VLNX300.PART0001.NATIVE TOVOLUME(LNX300) -
UNIT(3390) FROMDUMP(DUMPVOLUME(LX0001) SOURCEVOLUME(LNX300)) -
REPLACE
```

To recover a data set to an alternate location, use the RENAME and TOVOLUME options to rename the data set (using the naming convention listed in “Data Set and Partition Naming Conventions” on page 638) and to specify the volume that you want the data set recovered to.

The following command recovers data set LINUX.VLNX300.PART0002.NATIVE from the dump on dump tape LX0002 to volume LNX100 and renames it LINUX.VLNX100.PART0001.NATIVE, replacing the preallocated data set on LNX100.

```
RECOVER LINUX.VLNX300.PART0002.NATIVE -
FROMDUMP(DUMPVOLUME(LX0002)) -
NEWNAME(LINUX.VLNX100.PART0001.NATIVE) REPLACE -
TOVOLUME(LNX100) UNIT(3390)
```

Recovering a Partition with DFSMSdss

To recover a partition from a DFSMShsm dump with DFSMSdss, use the LIST command to get the same information needed to recover a volume with DFSMSdss. See “The DFSMShsm LIST Command” on page 637 on how to use the LIST command. You also need the data set name for the data set that holds the partition you want recovered and the data set name for the partition that you want the dump recovered to.

The following JCL recovers LINUX.VLNX300.PART0001.NATIVE from dump DFHSM.DMP.DAILY.VLNX300.D01244.T072510 on dump tape A02001 to volume LNX200 and renames it LINUX.VLNX200.PART0001.NATIVE, replacing the data set of the same name on LNX200. If you are restoring the data set to its original volume, then you can omit the RENAME portion.

```
//LNXJOB JOB ,'IBMUSER',MSGLEVEL=(1,1),TIME=(5,0),REGION=0M,
//      MSGCLASS=A,CLASS=A
//STEPDSS  EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=A02001,DISP=OLD,
               DSN=DFHSM.DMP.DAILY.VLNX300.D01244.T072510
//TARGET   DD UNIT=3390,VOL=SER=LNX200,DISP=OLD
//SYSIN    DD *
 RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) -
         DATASET(INCLUDE(LINUX.VLNX300.PART0001.NATIVE)) -
         RENAME(LINUX.VLNX200.PART0001.NATIVE) REPLACE
/*
```

Example Dump Environment

This example consists of one 3390-3 volume of about 2.9 gigabytes and one 3390-9 volume of about 8.7 gigabytes and a 65 cylinder 3390 of about 45 megabytes. The volumes will be dumped at 10:00 am to 3480 tapes on two 3480 tape drives. The 3390-3 and 3390-9 will be dumped daily and the 65 cylinder will be dumped weekly. The dump cycle will be a seven day, weekly cycle. The **fdasd** utility will be used to partition each volume as follows.

```
Disk /dev/dasd/0199/disc:
      65 cylinders,
      15 tracks per cylinder,
      12 blocks per track
    4096 bytes  per block
volume label: VOL1, volume identifier: LNX100
maximum partition number: 3

             ----------tracks----------
               Device      start      end   length   Id  System
 /dev/dasd/0199/part1          2      974      973    1  Linux native

Disk /dev/dasd/0198/disc:
    3339 cylinders,
      15 tracks per cylinder,
      12 blocks per track
    4096 bytes  per block
volume label: VOL1, volume identifier: LNX200
maximum partition number: 3

             ----------tracks----------
               Device      start      end   length   Id  System
 /dev/dasd/0198/part1          2     3201     3200    1  Linux native
 /dev/dasd/0198/part2       3202    50084    46883    2  Linux native

Disk /dev/dasd/019a/disc:
   10017 cylinders,
      15 tracks per cylinder,
      12 blocks per track
    4096 bytes  per block
volume label: VOL1, volume identifier: LNX300
maximum partition number: 3

             ----------tracks----------
               Device      start      end   length   Id  System
 /dev/dasd/019a/part1          2     3201     3200    1  Linux native
 /dev/dasd/019a/part2       3202     4267     1066    2  Linux native
 /dev/dasd/019a/part3       4268   150254   145987    3  Linux native
```

The partitions will be mounted as follows.

```
/dev/dasd/0198/part1 on /example type ext2 (rw)
/dev/dasd/0199/part1 on /example/boot type ext2 (rw)
/dev/dasd/0198/part2 on /example/usr type ext2 (rw)
/dev/dasd/019a/part1 on /example/var type ext2 (rw)
/dev/dasd/019a/part2 on /example/tmp type ext2 (rw)
/dev/dasd/019a/part3 on /example/home type ext2 (rw)
```

This example is designed to demonstrate how one might implement a simple DFSMShsm dump environment for a Linux guest on z/VM. It is not intended to demonstrate how to best partition volumes for Linux.

How to Define Your Dump Cycle

Before you start automatic dump, you must define a dump cycle. This environment uses a daily dump cycle. The start date must be today's date or a date that has already passed. You need to use a past date if you are using a dump cycle of more than one day and today is not the first day in the dump cycle.

The command below has the first Y day coincide with the day of 2001/08/26, which is a Sunday. Of the dump classes listed in the next section, the WEEKLY dump class has a DAY value of 7, so it is dumped on Saturdays. The MONTHLY dump class has a DAY value of 1, so it is dumped on Sundays. Because the DAILY dump class has no DAY value specified, it dumps on all Y days. If an N were in the dump cycle, automatic dump would not dump any volumes on that day.

DEFINE DUMPCYCLE(YYYYYYY CYCLESTARTDATE 2001/08/26)

How to Define Your Dump Classes

LNX100 must be backed up only once a week because it will not change very often. LNX200 and LNX300 change more often and must be backed up daily. Because this environment uses a weekly dump cycle, the DAY and FREQUENCY options determine which days each dump class is dumped. The daily and weekly dumps are short-term dumps that are not be kept for long. Another dump class creates dumps monthly and keeps the dumps for three months. All of the dump classes have a STACK value of three to allow dumps from up to three volumes to be put on one dump tape. One is the default STACK value, meaning that no stacking will occur. The following commands define the dump classes.

DEFINE DUMPCLASS(DAILY AUTOREUSE FREQUENCY(0) -
RETENTIONPERIOD(3) STACK(3) UNIT(3480) VTOCCOPIES(0))

DEFINE DUMPCLASS(WEEKLY AUTOREUSE FREQUENCY(0) DAY(7) -
RETENTIONPERIOD(28) STACK(3) UNIT(3480) VTOCCOPIES(0))

DEFINE DUMPCLASS(MONTHLY AUTOREUSE FREQUENCY(28) DAY(1) -
RETENTIONPERIOD(90) STACK(3) UNIT(3480) VTOCCOPIES(0))

If, for example, you have only two tape drives at your disposal, use the SETSYS command to set the maximum number of dumps that may be taken at any one time.

SETSYS MAXDUMPTASKS(2)

How to ADDVOL Your Volumes

Because the volumes will only be dumped, automatic migration and backup features are not allowed. When the dumps expire, the dump tapes are automatically reused for later dumps.

```
ADDVOL LNX100 UNIT(3390) PRIMARY(AUTODUMP(WEEKLY,MONTHLY) -
NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)

ADDVOL LNX200 UNIT(3390) PRIMARY(AUTODUMP(DAILY,MONTHLY) -
NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)

ADDVOL LNX300 UNIT(3390) PRIMARY(AUTODUMP(DAILY,MONTHLY) -
NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)
```

How to Start Automatic Dumping

To automatically dump the volumes, issue the SETSYS AUTODUMPSTART command with the start and finish times.

```
SETSYS AUTODUMPSTART(1100 1200 0000)
```

The times are in a 24-hour format. The first time is when automatic can start, the second time is the latest that an automatic dump can start, and the third time is the latest that a dump of a volume can start. Because the third time is 0000, all volumes are dumped if automatic dump starts. To turn off automatic dump, resubmit the command with all three times equal to 0000.

To manually dump the volumes, issue the BACKVOL command.

```
BACKVOL VOLUMES(LNX100,LNX300,LNX400) -
DUMP(DUMPCLASS(WEEKLY,MONTHLY))
```

If you omit DUMPCLASS(WEEKLY,MONTHLY), the volumes are dumped to the dump classes listed for each volume with the AUTODUMP option when it was added with ADDVOL, but no stacking occurs. You can list up to five dump classes. Dumps made manually are taken into account when recovering from a dump by generation or date. If you are only going to be dumping manually, you can omit the AUTODUMP option when you ADDVOL your volumes. You can use any valid dump class, regardless of the AUTODUMP option.

If you decide to create your dumps manually, you still need to run automatic dump to have old dumps expire. To have automatic dump run without actually taking dumps, you can define a dump cycle of one N day.

Recovering from a Dump

Before recovering any data, you must leave any partitions that you want to recover in place or reallocate them with **fdasd**; otherwise, Linux cannot see the partitions that have been recovered. Listed here are only a few basic examples of recovering data. For a full description on how to recover data, see the *z/OS DFSMShsm Storage Administration Reference* and the *z/OS DFSMSdss Storage Administration Reference*.

Recovering with DFSMShsm

To recover a volume with DFSMShsm, issue the RECOVER command.

The following RECOVER command recovers LNX200 from the most recent dump.

RECOVER * TOVOLUME(LNX200) UNIT(3390) FROMDUMP

The following RECOVER command recovers LNX200 from the most recent dump of the MONTHLY dump class.

RECOVER * TOVOLUME(LNX200) UNIT(3390) -
FROMDUMP(DUMPCLASS(MONTHLY))

The following RECOVER command recovers LNX200 from the third most recent dump. Generation 0 is the most recent dump, 1 is the second most recent, and so on.

RECOVER * TOVOLUME(LNX200) UNIT(3390) -
FROMDUMP(DUMPGENERATION(2))

To recover a partition from a DFSMShsm dump with DFSMShsm, you again use the RECOVER command and the LIST command. See "The DFSMShsm LIST Command" on page 637 on how to use the LIST command. You must use the LIST command to find the dump tape that has the dump from which you want to recover your partition. You also need to know the data set and partition naming conventions so that you recover the correct data set for your partition. The naming conventions are listed in "Naming Conventions" on page 638.

The following RECOVER command recovers data set LINUX.VLNX200.PART0002.NATIVE that holds partition part2 on LNX200. You must specify the SOURCEVOLUME parameter if the dump class has a STACK value greater than one because the dump tape may hold more than one dump data set.

RECOVER LINUX.VLNX200.PART0002.NATIVE -
FROMDUMP(DUMPVOLUME(LX0001) SOURCEVOLUME(LNX200)) REPLACE

Recovering Data from DFSMShsm Dump with DFSMSdss

Before you recover a volume from a DFSMShsm dump with DFSMSdss, you must use the LIST command to see what dumps you have. See"The DFSMShsm LIST Command" on page 637 on how to use the LIST command. Once you find the dump you want to use, you can write and submit JCL to have DFSMSdss recover the volume.

The following JCL recovers dump data set DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610 to volume LNX300. If there is more than one dump stacked on the tape, you must specify the file sequence number of the label keyword. You can get the file sequence number from the information provided by the LIST command in the next section.

```
//LNXJOB JOB ,USER=IBMUSER,PASSWORD=IBM,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS  EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=(A02005,A02008,A02010),DISP=OLD,
             LABEL=(2),DSN=DFHSM,DMP.MONTHLY.VLNX300.D01245.T234610
//TARGET   DD UNIT=3390,VOL=SER=LNX300,DISP=OLD
//SYSIN    DD *
 RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) PURGE
/*
```

To recover a partition from a DFSMShsm dump with DFSMSdss, use the LIST command like before to find the dump tape and dump data set from which you want to recover your partition. You also need the data set name of your partition. See “Data Set and Partition Naming Conventions” on page 638 for the partition and data set naming conventions.

The following JCL recovers data set LINUX.VLNX300.PART0001.NATIVE from dump data set DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610 to volume LNX300.

```
//LNXJOB JOB ,USER=IBMUSER,PASSWORD=IBM,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS  EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=(A02005,A02008,A02010),DISP=OLD,
            LABEL=(2),DSN=DFHSM,DMP.MONTHLY.VLNX300.D01245.T234610
//TARGET   DD UNIT=3390,VOL=SER=LNX300,DISP=OLD
//SYSIN    DD *
 RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) -
         DATASET(INCLUDE(LINUX.VLNX300.PART0001.NATIVE)) REPLACE
/*
```

Submitting DFSMShsm Commands Via JCL and FTP

You can submit DFSMShsm commands by sending JCL files through ftp to z/OS and have output data sets sent back to your Linux system. You must have TCP/IP and ftp running and a userid and password authorized to use ftp. Once you log into the z/OS ftp server you must enter the following command.

site filetype=jes

```
//LNXDUMP JOB ,CLASS=A,USER=IBMUSER,PASSWORD=IBM,
 // MSGCLASS=H,REGION=4096K,MSGLEVEL=(1,1)
 //LISTDUMP EXEC PGM=IKJEFT01
 //SYSTSPRT DD SYSOUT=A
 //SYSTSIN  DD *
 HSEND LIST DUMPVOLUME ODS(LIST.DUMP.VOLUMES)
 //FTPDUMP  EXEC PGM=FTP,COND=(0,NE,LISTDUMP),
 //          PARM='1.2.3.4'
 //SYSPRINT DD SYSOUT=*
 //OUTPUT   DD SYSOUT=*
 //SAVE     DD *
 /*
 //INPUT    DD *
 userid
 password
 ascii
 put 'LIST.DUMP.VOLUMES' dump.list
 quit
 /*
```

The IKJEFT01 program in the above JCL allows you to submit TSO line commands. The HSEND command allows DFSMShsm commands to be submitted from TSO. After HSEND you can use any DFSMShsm command. In the second step, the data set LIST.DUMP.VOLUMES is sent through ftp to IP address 1.2.3.4 and given a file name of dump.list .

The DFSMShsm LIST Command

The LIST command can be used to list several types of information about a DFSMShsm environment. For Linux volumes, we need to list only information for dump tapes. To create a list of dump tapes, use the following command.

LIST DUMPVOLUME OUTPUTDATASET(LIST.DUMP.VOLUMES)

The above command creates data set LIST.DUMP.VOLUMES. You can give the data set your own data set name. You can then browse the data set and search it for the information you need. The list is sorted by the volser of the dump tapes. A sample list is below.

```
-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 11:18:56 ON 01/09/02 FOR SYSTEM=3090

DUMP   VOL    UNIT     FILE  SOURCE             DUMPED     DUMPED                          PCT  SET OF DUMP
VOLSER STATUS TYPE     SEQ   VOLSER SMS CLASS    DATE       TIME     EXP DATE IDRC LIBRARY  FULL VOLSERS
A02001 UNEXP  3480                      DAILY                       2001/09/04 N  *NO LIB*  100
                      01    LNX200 N           2001/09/01 10:00:01                              A02001
                      02    LNX300 N           2001/09/01 10:25:07                              A02001 A02002
                                                                                                A02003
A02002 UNEXP  3480                      DAILY                       2001/09/04 N  *NO LIB*  100
                      **    LNX300 N           2001/09/01 10:25:07                              A02001 A02002
                                                                                                A02003
A02003 UNEXP  3480                      DAILY                       2001/09/04 N  *NO LIB*  255
                      **    LNX300 N           2001/09/01 10:25:07                              A02001 A02002
                                                                                                A02003
A02004 UNEXP  3480                      DAILY                       2001/09/05 N  *NO LIB*  100
                      01    LNX200 N           2001/09/02 10:00:01                              A02004
                      02    LNX300 N           2001/09/02 10:46:23                              A02004 A02007
                                                                                                A02009
A02005 UNEXP  3480                      MONTHLY                     2001/11/25 N  *NO LIB*  100
                      01    LNX200 N           2001/09/02 10:00:01                              A02005
                      02    LNX300 N           2001/09/02 10:46:23                              A02005 A02008
                                                                                                A02010
A02006 UNEXP  3480                      MONTHLY                     2001/11/25 N  *NO LIB*   25
                      01    LNX100 N           2001/09/02 10:00:02                              A02006
A02007 UNEXP  3480                      DAILY                       2001/09/05 N  *NO LIB*  100
                      **    LNX300 N           2001/09/02 10:46:23                              A02004 A02007
                                                                                                A02009
A02008 UNEXP  3480                      MONTHLY                     2001/11/25 N  *NO LIB*  100
                      **    LNX300 N           2001/09/02 10:46:23                              A02005 A02008
                                                                                                A02010
A02009 UNEXP  3480                      DAILY                       2001/09/05 N  *NO LIB*  255
                      **    LNX300 N           2001/09/02 10:46:23                              A02004 A02007
                                                                                                A02009
A02010 UNEXP  3480                      MONTHLY                     2001/11/25 N  *NO LIB*  255
                      **    LNX300 N           2001/09/02 10:46:23                              A02005 A02008
                                                                                                A02010
A02101 UNEXP  3480                      WEEKLY                      2001/09/29 N  *NO LIB*   25
                      01    LNX100 N           2001/09/01 10:00:02                              A02101
----- END OF - DUMP VOLUME - LISTING -----
```

Once you find the dump tape made on the date and of the same dump class as the volume you want to recover from, you must submit another LIST command to list the dump data sets on that dump tape.

LIST DUMPVOLUME(A02001) ODS(LIST.DUMP.A02001)

The above command creates the following list of the dump information about dump tape A02001.

```
-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 11:21:09 ON 01/09/02 FOR SYSTEM=3090

DUMP   VOL    UNIT     FILE  SOURCE              DUMPED     DUMPED                            PCT  SET OF DUMP
VOLSER STATUS TYPE     SEQ   VOLSER SMS CLASS    DATE       TIME     EXP DATE IDRC LIBRARY    FULL VOLSERS
A02001 UNEXP  3480                      DAILY                        2001/09/04 N  *NO LIB*  100
                       01    LNX200 N            2001/09/01 10:00:01                               A02001
DUMP COPY DATA SET NAME = DFHSM.DMP.DAILY.VLNX200.D01244.T010010
                       02    LNX300 N            2001/09/01 10:25:07                               A02001 A02002
                                                                                                    A02003
DUMP COPY DATA SET NAME = DFHSM.DMP.DAILY.VLNX300.D01244.T072510
----- END OF - DUMP VOLUME - LISTING -----
```

From the sample listing above, you can see that dump tape A02001 contains two dump data sets, one of volume LNX200 and the first part of the dump data set for volume LNX300. The name for the dump data set of volume LNX200 is DFHSM.DMP.DAILY.VLNX200.D01244.T010010 and the file sequence number is 1. The dump data set for volume LNX300, DFHSM.DMP.DAILY.VLNX300.D01244.T072510, starts on dump tape A02001 and spans onto dump tapes A02002 and A02003. The file sequence number is 2. You need only the volser of the first dump tape when using the RECOVER command to recover this dump.

Naming Conventions

The following naming conventions apply to data sets, partitions, and volsers.

Data Set and Partition Naming Conventions

The data sets and the partitions they hold have the following naming convention.

```
Dataset Names                   Partition Names

LINUX.Vvolser.PART0001.type     /dev/dasd/yyyy/part1
LINUX.Vvolser.PART0002.type     /dev/dasd/yyyy/part2
LINUX.Vvolser.PART0003.type     /dev/dasd/yyyy/part3
```

Where *volser* is the volume serial number of the volume that the data set is on. *yyyy* is the device number of the volume in the Linux environment. Type can be NATIVE or SWAP. Only NATIVE partitions need to be backed up. SWAP partitions are the Linux equivalent to page packs.

Volser Naming Conventions

The volume serial number of a volume must be 6 characters of upper case letters or numbers. The volser can also have @ , # , and $.

Valid Volsers

```
LINUXA     LNX100    $BILLS     VOL#11
123456     A@FB2#    $@#32A     5#2$XA
```

References

You can search and view mainframe documentation online or download it in the pdf format from the following web pages.

- For os/390 at http://www-1.ibm.com/servers/s390/os390/bkserv/find_books.html
- For z/OS at http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/find_books.html
- http://www.redbooks.ibm.com/

To find DFSMShsm or DFSMSdss documentation, search either of the first two sites for any of the following manuals by title or serial number.

- *z/OS DFSMShsm Storage Administration Guide*
- *z/OS DFSMShsm Storage Administration Reference*
- *z/OS DFSMShsm Implementation and Customization Guide*
- *z/OS DFSMSdss Storage Administration Guide*
- *z/OS DFSMSdss Storage Administration Reference*
- *z/OS MVS JCL Reference*

At the third link, you can search for more DFSMS documentation, and you can find the Linux for the S/390 Redbook, SG24-4987.

Appendix D. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/`

One exception is command syntax that is published in railroad track format; screen-readable copies of z/OS books with that syntax information are separately available in HTML zipped file form upon request to mhvrcfs@us.ibm.com.

© Copyright IBM Corp. 1984, 2004

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 give 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.

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

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

© Copyright IBM Corp. 1984, 2004

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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

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

Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of DFSMShsm.

This document also contains intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSMShsm. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Trademarks

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

AIX	IMS
CICS	MVS
DFSMS/MVS	MVS/DFP
DFSMSdfp	MVS/ESA
DFSMSdss	MVS/XA
DFSMShsm	OS/390
DFSMSrmm	Parallel sysplex
DFSORT	RACF
ESCON	ResourceLink
IBM	z/OS
IBMLink	z/VM

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

NetView is a trademark of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

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

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

Glossary

This glossary defines technical terms and abbreviations used in DFSMShsm documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMShsm manual or view *IBM Glossary of Computing Terms*, located at **http://www.ibm.com/ibm/terminology**

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

Numerics

3480. An IBM 3480 Magnetic Tape Subsystem device that is capable of recording data only in the 3480 format.

3480X. A 3480 XF device or an IBM 3490 Magnetic Tape Subsystem device that is capable of recording data in either the 3480 format or the 3480 XF (IDRC) format. DFSMShsm can, at the user's option, select either IDRC or non-IDRC recording.

3490. An IBM 3490 Magnetic Tape Subsystem device that records data in the 3480-2 XF format. DFSMShsm always requests IDRC recording.

3490E. See *3490*.

3590. An IBM TotalStorage Enterprise Tape Drive 3590 that operates as a 3590-B1x. See *3590-B1x*, *3590-E1x* and *3590-H1x*.

3590B1*x*. An IBM TotalStorage Enterprise Tape Drive 3590 Model B1*x* that uses the 3590 High Performance Cartridge, writes in 128-track format, and can emulate the 3490 Magnetic Tape Subsystem.

3590E1*x*. An IBM TotalStorage Enterprise Tape Drive 3590 Model E1*x* that uses the 3590 High Performance Cartridge, can read 128- or 256-track format tapes, and writes in 256-track format. This drive emulates either the IBM 3490 magnetic tape drive or the IBM TotalStorage Enterprise Tape Drive 3590 Model B1*x*.

3590H1*x*. An IBM TotalStorage Enterprise Tape Drive 3590 Model H1*x* that uses the 3590 High Performance Cartridge, can read 128-, 256-, or 384-track format tapes, and writes in 384-track format. This drive emulates either the IBM 3490 magnetic tape drive or the IBM TotalStorage Enterprise Tape Drive 3590 Model B1*x* or Model E1*x*.

3592J1A. IBM TotalStorage Enterprise Tape Drive 3592 that uses the 3592 Enterprise Tape Cartridge and writes in enterprise format 1 (EFMT1). This drive emulates either the IBM 3490 magnetic tape drive or the IBM Total Storage Enterprise Tape Drive 3590 Model B1*x*.

A

ABARS. Aggregate backup and recovery support.

ABEND. Abnormal end.

ABR. Aggregate backup and recovery record.

ACB. Access control block.

accompany data set. In aggregate backup and recovery processing, a data set that is physically transported from the backup site to the recovery site instead of being copied to the aggregate data tape. It is cataloged during recovery.

ACCOMPANY keyword. The keyword used in the selection data set to create an accompany list.

accompany list. An optional list in the selection data set that identifies the accompany data sets.

ACEE. Access control environment element.

ACS. Automatic class selection.

active data. Data that is frequently accessed by users and that resides on level 0 volumes.

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

© Copyright IBM Corp. 1984, 2004

AG. Aggregate group.

aggregate backup. The process of copying the data sets and control information of a user-defined group of data sets so that they may be recovered later as an entity by an aggregate recovery process.

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

aggregate group. A Storage Management Subsystem class that defines control information and identifies the data sets to be backed up by a specific aggregate backup.

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

AIX. Alternate index.

allocate data set. In aggregate backup and recovery processing, a data set name that is listed in the selection data set. The space for this data set is allocated and the data set is cataloged at the recovery location, but the actual data is not restored.

ALLOCATE keyword. The keyword used in the selection data set to create an allocate list.

allocate list. An optional list in the selection data set that identifies the allocate data sets.

alternate index. In systems with VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the associated base cluster data records. Its function is to provide an alternate directory for locating records in the data component of a base cluster. See also *path.*

alternate index cluster. In VSAM, the data and index components of an alternate index.

alternate tape volume reference. In DFSMShsm, additional fields in the TTOC record that record information about the alternate tape volume. These fields provide DFSMShsm with the necessary information to refer to the alternate tape volume.

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

APAR. Authorized program analysis report.

APF. Authorized program facility.

ASID. Address space identifier.

ATL. Automated tape library.

audit. A DFSMShsm process that detects discrepancies between data set information in the VTOCs, the computing system catalog, the MCDS, BCDS, and OCDS.

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

authorized user. In DFSMShsm, the person or persons who are authorized through the DFSMShsm AUTH command to issue DFSMShsm system programmer, storage administrator, and operator commands.

automated tape library. A device consisting of robotic components, cartridge storage frames, tape subsystems, and controlling hardware and software, together with the set of volumes which reside in the library and may be mounted on the library tape drives.

automatic backup. In DFSMShsm, the process of automatically copying eligible data sets from DFSMShsm-managed volumes or migration volumes to backup volumes during a specified backup cycle.

automatic cartridge loader feature. A feature of the 3480 Magnetic Tape Subsystem providing the operator with the capability of preloading multiple tapes to be used as migration, backup, or dump tapes.

automatic class selection (ACS). A mechanism for assigning SMS classes and storage groups.

automatic dump. In DFSMShsm, the process of using DFSMSdss to automatically do a full volume dump of all allocated space on DFSMShsm-managed volumes to designated tape dump volumes.

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

automatic primary space management. In DFSMShsm, the process of automatically deleting expired data sets, deleting temporary data sets, releasing unused overallocated space, and migrating data sets from DFSMShsm-managed volumes.

automatic secondary space management. In DFSMShsm, the process of automatically deleting expired migrated data sets from the migration volumes, deleting expired records from the migration control data set, and migrating eligible data sets from level 1 volumes to level 2 volumes.

automatic space management. In DFSMShsm, includes automatic volume space management, automatic secondary space management, and automatic recall.

automatic volume space management. In DFSMShsm, includes automatic primary space management and automatic interval migration.

availability management. In DFSMShsm, the process of ensuring that a current version (backup copy) of the installation's data sets resides on tape or DASD.

B

backup. In DFSMShsm, the process of copying a data set residing on a level 0 volume, a level 1 volume, or a volume not managed by DFSMShsm to a backup volume. See *automatic backup, incremental backup.*

backup control data set (BCDS). A VSAM, key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

backup copy. In DFSMShsm, a copy of a data set that is kept for reference in case the original data set is destroyed.

backup cycle. In DFSMShsm, a period of days for which a pattern is used to specify the days in the cycle on which automatic backup is scheduled to take place.

backup frequency. In DFSMShsm, the number of days that must elapse since the last backup version of a data set was made until a changed data set is again eligible for backup.

backup profile. In DFSMShsm, a RACF discrete data set profile associated with the backup version of a cataloged data set that is protected by a RACF discrete data set profile.

backup version. Synonym for backup copy.

backup volume. A volume managed by DFSMShsm to which backup versions of data sets are written.

backup volume cleanup process. A DFSMShsm process that scratches data set backup versions that are no longer needed on DASD.

backup VTOC copy data set. In DFSMShsm, a copy of the VTOC of a volume that was backed up by DFSMShsm. This VTOC data set contains only part of the data set VTOC entry for each data set from the original data set. This data set is written on a migration level 1 volume.

base cluster. In systems with VSAM, a key-sequenced or entry-sequenced file over which one or more alternate indexes are built. See also *cluster.*

base data component. In VSAM, a component of the base cluster containing data of a data set.

base sysplex. A base (or basic) sysplex is the set of one or more MVS systems that is given a cross-system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A base system does not include a coupling facility. See also *parallel sysplex* and *sysplex*.

BCDS. Backup control data set.

BCR. Backup control record.

BDAM. Basic direct access method.

BSAM. Basic sequential access method.

BVR. Backup cycle volume record.

C

catalog. (1) A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (A) (ISO) (2) To enter information about a file or a library into a catalog. (A) (ISO) (3) The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set. (4) To include the volume identification of a data set in the catalog. (5) See *VSAM master catalog, VSAM user catalog.*

CDD. Common data set descriptor record.

CDS. Control data set.

CDSQ serialization. A technique that serializes DFSMShsm CDSs with a global enqueue product (such as GRS).

CDSR serialization. A technique that serializes DFSMShsm CDSs with volume reserves.

CDT. Class descriptor table.

CF. Coupling facility.

changed data set. In DFSMShsm, a data set that has been opened for other than read-only access.

CLIST. Command list.

cluster. In systems with VSAM, a named structure consisting of a group of related components, for example, a data component with its index component. See also *base cluster.*

command list. A command procedure containing executable sequences of TSO commands, subcommands, and command procedure statements.

command procedure. In TSO, a data set or a member of a partitioned data set containing TSO commands to be performed sequentially by the EXEC command. See also *command list.*

common data set descriptor record. The record that precedes a user's data set on a DFSMShsm-owned volume and that is used to return the data set to the user's format.

common filter services. A subcomponent of DFP common services. Common filter services compares data items with filter keys and indicates which data items match the keys and how many matches have been found.

common recall queue (CRQ). A single recall queue shared by multiple DFSMShsm hosts that enables the recall workload to be balanced across each of those hosts.

common service area (CSA). In OS/VS2, a part of the common area that contains data areas addressable by all address spaces, but protected during its use by the key of the requester.

compaction. In DFSMShsm, a method of compressing and encoding data that is migrated or backed up.

comprehensive command authority. In DFSMShsm, allowing a user to issue the ABACKUP command for all aggregate groups.

compress. In DFSMShsm, to release unused space in a partitioned data set during the migrate/recall and backup/recovery processes.

computing system catalog. In DFSMShsm, the master catalog and any associated user catalogs used as sources during the audit process.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

connected set. A group of tapes volumes that are related to each other because one or more data sets span the physical tape volumes. Or, a single tape volume with a data set or data sets that do not span to any other volumes.

contiguous space. An unbroken consecutive series of storage locations.

control data set. In DFSMShsm, one of three data sets (BCDS, MCDS, and OCDS) that contain records used in DFSMShsm processing.

control file. In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the catalog, allocation, volume, and related information necessary to perform aggregate recovery.

converter/interpreter processing. The job segment that converts and interprets JCL for MVS.

copy pool. A named set of storage groups that contains data to be copied and recovered collectively.

copy pool backup storage group. A type of SMS storage group that contains all of the volumes that DFSMShsm might select as the target of a fast replication.

coupling facility. A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

cross-system coupling facility (XCF). A component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

CRQ. Common recall queue.

CRQplex. One or more DFSMShsm hosts that are connected to the same CRQ. There may be multiple CRQplexes within a single HSMplex.

CSA. Common service area.

CSECT. Control section.

CTC. Channel-to-channel.

current backup version. In DFSMShsm, a backup copy of the data set that was created on a date after the data set was last updated.

CVT. Communications vector table.

cycle start date. In DFSMShsm, the date a backup cycle, dump cycle, or migration cleanup cycle is started.

D

DADSM. Direct Access Device Space Management.

DAE. Dump analysis elimination.

daily backup volume. In DFSMShsm, a volume associated with a given day in the backup cycle and assigned to contain backup versions created on that cycle day.

DAIR. Dynamic allocation interface return code.

DASD. Direct access storage device.

DASD calculation services (DCS). A subcomponent of DFP common services. DCS retrieves and calculates data set information for both VSAM and non-VSAM data sets based on the user's input request.

data class. A list of allocation attributes that the system uses for the creation of data sets.

data control block (DCB). A control block used by access method routines in storing and retrieving data.

data file. In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the backup copies of the data sets to be recovered.

data migration. See *migration.*

data set change indicator. A bit in the Format 1 data set VTOC entry that indicates whether the data set has been opened for output.

data set deletion. In DFSMShsm, the space management technique of deleting non-SMS-managed data sets that have not been used for a specified number of days and that do not have expiration date protection.

data set group. Data sets that have the same set of initial characters in their names.

data set organization. The type of arrangement of data in a data set. Examples are sequential organization or partitioned organization.

data set pool. One or more volumes managed by DFSMShsm to which data sets that have migrated can be recalled, depending on the set of initial characters of the data set name.

data set retirement. In DFSMShsm, the space management technique of deleting non-SMS-managed data sets that have not been referred to for a specified number of days, and that have a backup version. See also *delete-if-backed-up*.

date last referred to. In DFSMShsm, the last date when a data set was opened.

DBA. Delete-by-age.

DBU. Delete-if-backed-up.

DCB. Data control block.

DCL. Dump class record.

DCR. Dump control record.

DCS. DASD calculation services.

debug mode. In DFSMShsm, the method of operation that projects the changes that would occur in normal operation but in which no user data moves.

decompaction. In DFSMShsm, the process of decoding and expanding data that was compacted during daily space management or backup.

delete-by-age (DBA). In DFSMShsm, the space management technique of deleting non-SMS-managed data sets that have not been opened for a specified number of days.

delete-if-backed-up (DBU). In DFSMShsm, the space management technique of deleting non-SMS-managed data sets that have not been opened for a specified number of days, and that have a current backup version. See also *data set retirement*.

demotion. The process of one host losing level functions to another. Both original and promoted hosts can go through the demotion process. See also *promotion.*

DFP. Data Facility Product.

DFP common services. A component of DFP that contains three subcomponents: common filter services (CFS), DASD calculation services (DCS), and device information services (DIS).

DFSMS. Data Facility Storage Management Subsystem.

DFSMSdss. A functional component of DFSMS/MVS used to copy, dump, move, and restore data sets and volumes.

DFSMShsm. A functional component of DFSMS/MVS used to manage volumes and data sets.

DFSMShsm-authorized user. In DFSMShsm, the person or persons who are authorized through the DFSMShsm AUTH command to issue system programmer and storage administrator commands.

DFSMShsm log. In DFSMShsm, a pair of sequential data sets, X and Y, containing a chronological list of transactions and statistics occurring in DFSMShsm.

DFSMSdfp. A functional component of DFSMS/MVS used to manage programs, devices, and data.

DFSMShsm-managed volume. A volume managed by DFSMShsm containing data sets that are directly accessible to the user.

DFSMShsm-owned volumes. Storage volumes on which DFSMShsm stores backup versions, dump copies, or migrated data sets.

DFSMShsm secondary address space. A separate address space started and controlled by DFSMShsm to perform aggregate backup or aggregate recovery processing.

DFSORT. An IBM licensed program that serves as a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, copying and reporting operations as well as versatile data manipulation at the record, field and bit level.

DGN. Dump generation record.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of data.

directed recall. Moving a migrated data set from a level 1 or a level 2 volume to a level 0 volume and specifying the target volume and unit name where the data set can be allocated.

disaster. An unplanned occurrence that keeps a company or organization from conducting its normal business for some time period.

disaster backup. A means to protect a computing system complex against data loss in the event of a disaster.

disaster recovery. A means to replace lost data at another location with sufficient resources in order to resume operation.

discrete backup profile. A RACF profile created when DFSMShsm backs up a cataloged, RACF-indicated data set.

discrete profile. A RACF profile that contains security information about a specific data set on a particular volume.

disposition processing. In OS/VS, a function performed by the initiator at the end of a job step to keep, delete, catalog, or uncatalog data sets, or to pass them to a subsequent job step, depending on the data set status of the disposition specified in the DISP parameter of the DD statement.

Distributed System License Option (DSLO). A license option available to IBM customers with a basic license that permits them to copy certain IBM-licensed materials for the purpose of installing multiple systems.

DSLO. Distributed Systems License Option.

DSR. Daily statistics record.

dump. See *full volume dump.*

dump class. A DFSMShsm-named set of characteristics that describe how volume dumps are managed.

dump copy. In DFSMShsm, a copy of the volume image produced by the DFSMSdss full volume dump function.

dump cycle. In DFSMShsm, a period of days for which a pattern is used to specify the days in the cycle on which automatic full volume dump is scheduled to take place.

dump generation. A successful full volume dump of a volume that may contain one to five identical dump copies.

dump VTOC copy data set. In DFSMShsm, a copy of the VTOC of a volume dumped by DFSMShsm. This VTOC data set contains only part of the data set VTOC entry for each data set from the original data set. This data set is written on a migration level 1 volume.

DVL. Dump volume record.

DVT. Device vector table.

E

EA. Extended addressability.

EBCDIC. Extended Binary Coded Decimal Interchange Code.

ECB. Event control block.

eligibility age. The number of days since a data set met its criteria to be migrated.

emergency mode. In DFSMShsm, the method of operation that prevents data set movement and deletion in space management, backup, and recovery processes.

encode. (T) To convert data by the use of a code in such a manner that reconversion to the original form is possible.

erase-on-scratch. A RACF and DFP/XA function that overwrites the space occupied by a data set when the data set is scratched from a DASD device supported by MVS/XA.

ESDS. Entry-sequenced data set.

esoteric unit names. The names a user assigns to I/O devices that have the same device type. When the user specifies the assigned unit name to DFSMShsm, DFSMShsm associates the unit name to its device type.

ESTAE. Extended subtask ABEND exit.

ESTAI. Extended subtask ABEND intercept.

exclude data set. In aggregate backup and recovery processing, a data set in the selection data set exclude list. This data set is to be excluded from being processed by aggregate backup.

EXCLUDE keyword. The keyword used in the selection data set to create an exclude list.

exclude list. An optional list in the selection data set that identifies those data sets that are to be excluded from being processed by aggregate backup.

expiration. The removal of a user data set from either a user (non-DFSMShsm-owned) volume, or from a DFSMShsm-owned volume when the user data set has been migrated. If there is an explicit expiration date, it

is found in the data set VTOC entry for a nonmigrated data set, or in the MCD record for a migrated data set. If there is no explicit expiration date, the management class attributes are checked to determine an implicit expiration date.

extended addressability. A type of extended format VSAM data set that allows greater than 4GB of data storage.

extended format compressed data set. A SAM or VSAM data set whose processing detects errors caused by an incomplete transfer of a physical block of data to a physical record in the data set. Each physical record is ended by a "hidden" suffix. SAM or VSAM processing determines, by examining these suffixes, if a data transfer to a physical record has ended prematurely.

extended remote copy (XRC). A hardware- and software-based remote copy service option that provides an asynchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

extent reduction. In DFSMShsm, the releasing of unused space, reducing the number of extents, and compressing partitioned data sets.

extents. A continuous space on a direct access storage volume, occupied by or reserved for a particular data set, data space, or file.

F

facility class profile. A way that RACF protects all DFSMShsm command access. An active RACF facility class establishes the security environment.

fallback. Pertaining to returning to use of an earlier release of a program after a later release has been installed and used.

fast replication. A general term to describe the functions that create an instant data copy. The Enterprise Storage Server FlashCopy function is an example of fast replication.

fast replication target volume. A DFSMShsm-owned volume that contains the fast replication backup copy of a DFSMShsm-managed volume. Fast replication target volumes are contained within SMS copy pool backup storage groups.

FBA. Fixed-block architecture.

FIFO. First in, first out.

fixed-block architecture. Data stored in blocks of fixed size; these blocks are addressed by block number relative to the beginning of the particular file.

FMID. Function modification identifier.

fragmentation index. The qualitative measure of the scattered free space on a volume.

FSR. Functional statistics record.

full volume dump. In DFSMShsm, the process of using a DFSMSdss function that backs up the entire allocated space on a volume.

full volume restore. In DFSMShsm, the process of using a DFSMSdss function that restores the entire volume image.

functional statistics record. A record that is created each time a DFSMShsm function is processed. It contains a log of system activity and is written to the system management facilities (SMF) data set.

functional verification procedure. A procedure distributed with DFSMShsm that tests to verify that all basic DFSMShsm functions are working correctly.

G

GDG. Generation data group.

GDS. Generation data set.

general pool. In a DFSMShsm environment with JES3, the collection of all DFSMShsm primary volumes added to that processor that have a mount status of permanently-resident or reserved, that have the automatic recall attribute specified, and that have a mount attribute of storage or private.

generation data group. A collection of data sets with the same base name, such as PAYROLL, that are kept in chronological order. Each data set is called a generation data set.

generic profile. A RACF profile that contains security information about multiple data sets, users, or resources that may have similar characteristics and require a similar level of protection.

global scratch pool. A group of empty tapes that do not have unique serial numbers and are not known individually to DFSMShsm. The tapes are not associated with a specific device. Contrast with *specific scratch pools.*

GRS. Global resource serialization.

GRSplex. One or more MVS systems using global serialization to serialize access to shared resources.

GTF. Generalized trace facility.

H

hardware/software compression facility. A function for both SAM and VSAM extended format compressed

data sets that has the ability to utilize either hardware data compression for best performance, or software data compression for DASD space considerations.

high threshold of occupancy. In DFSMShsm, the upper limit of space to be occupied on a volume managed by DFSMShsm. Contrast with *low threshold of occupancy.*

HMT. HSM Monitor/Tuner.

HSMplex. One or more DFSMShsm hosts that share a set of control data sets. The hosts may be on single or multiple z/OS images.

I

ICETOOL. DFSORT's multipurpose data processing and reporting utility.

IDRC. Improved data recording capability.

improved data recording capability. An improved data recording mode that, depending on data set characteristics and machine model, may allow a significant increase in effective cartridge data capacity and performance.

inactive age. In DFSMShsm, the number of days since the data set was last referred to.

inactive data. Copies of active or low-activity data that reside on DFSMShsm-owned dump and incremental backup volumes. See also *low-activity data.*

include data set. In aggregate backup and recovery processing, a data set in the selection data set include list. This data set is processed by aggregate backup.

INCLUDE keyword. The keyword used in the selection data set to create an include list.

include list. A required list in the selection data set that identifies the include data sets that are to be processed by aggregate backup.

incremental backup. In DFSMShsm, the process of copying a data set that has been opened for other than read-only access since the last backup version was created, and that has met the backup frequency criteria.

incremental recovery. Recovery of the latest backup copy of a data set or data sets made by incremental backup.

inline backup. The process of copying a specific data set to a migration level 1 volume from a batch environment. This process allows you to back up data sets in the middle of a job.

installation verification procedure (IVP). A procedure distributed with DFSMShsm that tests to verify that the basic facilities of DFSMShsm are functioning correctly.

installation exit. A means specified by an IBM software product by which a customer's system programmers may modify or extend the functions of the IBM software product. Such modifications or extensions consist of exit routines written to replace an existing module of an IBM software product, or to add one or more modules or subroutines to an IBM software product for the purpose of modifying the functions of the IBM software product.

instruction data set. In aggregate backup and recovery processing, a data set that contains instructions, commands, or any data the aggregate backup site defines as needed for aggregate recovery at the recovery site.

instruction file. In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the instruction data set.

Interactive Storage Management Facility (ISMF). The interactive panels of DFSMSdfp that allow users and storage administrators access to the storage management functions of DFSMSdss and DFSMShsm.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive access method for all storage management functions.

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

IPCS. Interactive problem control system.

ISMF. Interactive Storage Management Facility.

ISPF. Interactive System Productivity Facility.

ISPF/PDF. Interactive System Productivity Facility/Program Development Facility.

IVP. Installation verification procedure.

J

JCL. Job control language.

JES. Job entry subsystem.

JES2. Job entry subsystem 2.

JES3. Job entry subsystem 3.

JFCB. Job file control block.

journal data set. In DFSMShsm, a sequential data set used by DFSMShsm for recovery of the MCDS, BCDS, and OCDS. The journal contains a duplicate of each record in the control data sets that has changed since the MCDS, BCDS, and OCDS were last backed up.

K

KB. Kilobyte; 1024 bytes.

key-range data set. A key-sequenced data set that is defined with one or more KEYRANGE groups.

KSDS. Key-sequenced data set.

L

L2CR. Migration level-2 control record.

LE. Logical end.

level functions. (1) Functions that must be performed but need be performed by only one processing unit. Level functions include backing up the control data sets, backing up migrated data sets, deleting expired dump copies, moving backup versions from level 1 volumes to backup volumes, migration cleanup and level-1-to-level-2 migration. (2) See *primary-processing-unit functions.*

level 0 volume. A volume that contains data sets directly accessible by the user. The volume may be either DFSMShsm-managed or non-DFSMShsm-managed.

level 1 volume. A volume owned by DFSMShsm containing data sets that migrated from a level 0 volume.

level 2 volume. A volume under control of DFSMShsm containing data sets that migrated from a level 0 volume, from a level 1 volume, or from a volume not managed by DFSMShsm.

like device. Pertaining to (DASD) devices with identical geometry: that is, the same number of bytes per track, the same number of tracks per cylinder, and the same number of cylinders per actuator.

linear data set. In VSAM, a named linear string of data, stored in such a way that it can be retrieved or updated in 4096-byte units.

logical end (LE). Signifies that a concurrent copy environment has been successfully initialized. After logical end, the data is again available for unrestricted application access. Actual movement of data begins after logical end. Contrast with*physical end*.

low-activity data. Data that is infrequently accessed by users and is eligible to be moved or has been moved to DFSMShsm-owned migration volumes.

low threshold of occupancy. The lower limit of space to be occupied on a volume managed by DFSMShsm. Contrast with *high threshold of occupancy.*

M

managed volume. See *DFSMShsm-managed volume* and *primary volume.*

management class. A list of data set migration, backup, and retention attributes that DFSMShsm uses to manage storage at the data set level.

management work element (MWE). A control block containing the necessary information to direct DFSMShsm functions.

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which may be mounted on those drives.

master catalog. A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

MB. Megabyte; 1 048 576 bytes.

MC1. Migration level-1 free space record.

MCA. Migration control data set alias entry record.

MCB. BCDS data set record.

MCC. Backup version record.

MCD. MCDS data set record.

MCDS. Migration control data set.

MCL. Backup CDS backup changed migrated data set record.

MCM. Backup CDS move backup version record.

MCO. Migrated CDS VSAM associations record.

MCP. Eligible volume record.

MCR. Management control record.

MCT. Backup volume record.

MCU. Migration CDS user record.

MCV. Primary and migration volume record.

MCVT. Management communication vector table.

MHCR. Multiple-processing-unit control record.

migration. In DFSMShsm, the process of moving a cataloged data set from a DFSMShsm-managed volume to a migration level 1 or migration level 2 volume, from a migration level 1 volume to a migration level 2 volume, or from a volume not managed by DFSMShsm to a migration level 1 or migration level 2 volume.

migration cleanup. In DFSMShsm, a process of automatic secondary space management that deletes unnecessary records or migration copies.

migration control data set (MCDS). A VSAM, key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMShsm.

migration level 1 volume. Level 1 volume.

migration level 2 volume. Level 2 volume.

migration volume. A volume, under control of DFSMShsm, that contains migrated data sets.

minimal discrete profile. A profile with no access list or model profile. The minimal discrete profile is used when recovering a RACF-indicated data set whose original profile or backup profile no longer exists.

minimum migration age. In DFSMShsm, the number of days a data set must remain unopened before DFSMShsm can select it to migrate from a volume.

MIPS. Million instructions per second.

ML1. Migration level 1. Synonym for level 1 volume.

ML2. Migration level 2. Synonym for level 2 volume.

model entity. A model data set name that defines a discrete data set profile for RACF protection.

MSS. Mass Storage System.

MTL. Manual tape library.

multicluster, key-range control data set. A control data set (MCDS or BCDS) that is represented by two or more key-sequenced data set (KSDS) clusters. Each cluster is a single key range. Contrast with *single-cluster, key range control data set.*

multiple-file format. In DFSMShsm, a 3480 tape format, or the equivalent, that requires a unique standard label data set for each user data set written. When DFSMShsm writes in multiple-file format, it writes one tape data set for every user data set to all 3480 migration and backup volumes.

multiple DFSMShsm-host environment. Any environment in which two or more DFSMShsm hosts share a common set of control data sets.

mutually exclusive parameters. A set of parameters of which only one can be used. If more than one parameter is specified, only the last parameter specified is used.

MVS/Enterprise Systems Architecture (MVS/ESA). An MVS operating system environment that supports accessing of virtual storage in multiple address spaces and data spaces.

MVS/ESA. MVS/Enterprise Systems Architecture.

MVS/Extended Architecture (MVS/XA). An MVS operating system environment that supports 31-bit real and virtual storage addressing, increasing the size of addressable real and virtual storage from 16 megabytes to 2 gigabytes.

MVS/SP™. An IBM licensed program used to control the MVS operating system and to establish a base for an MVS/XA or MVS/370 environment.

MVT. Mounted volume table.

MWE. Management work element.

N

non-DFSMShsm-managed volume. A volume not defined to DFSMShsm containing data sets that are directly accessible to users.

O

OCDS. Offline control data set.

O/C/EOV. Open/close/end-of-volume.

OCO. Object code only.

offline control data set (OCDS). In DFSMShsm, a VSAM, key-sequenced data set that contains information about tape backup volumes and tape migration level 2 volumes.

online. (1) Pertaining to the operation of a functional unit when under the direct control of a computer. (A) (ISO) (2) Pertaining to a user's ability to interact with a computer. (A) (3) Pertaining to a user's access to a computer via a terminal. (A) (4) Controlled by, or communicating with, a computer.

original tape volume. In DFSMShsm, a 3480 or 3490 single-file tape volume, or the equivalent, used to store data during migration or backup processing, and from which a copy (called the alternate volume) is made for disaster recovery.

OS/VS2. A virtual storage operating system that is an extension of OS/MVT.

owned space. The storage space on a set of volumes to which DFSMShsm allocates migrated data sets and backup versions, but to which user jobs should not allocate. Included in this set are migration level 1, migration level 2, and backup volumes.

owned volume. A volume on which DFSMShsm writes dump, migration, or backup data sets.

P

parallel sysplex. A sysplex with one or more coupling facilities. See also *base sysplex* and *sysplex*.

partitioned data set (PDS). A data set in DASD that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A DFP library structure that is an enhanced replacement for a partitioned data set.

path. (1) (T) In a network, any route between any two nodes. (2) In a data base, a sequence of segment occurrences from the root segment to an individual segment. (3) In VSAM, a named, logical entity providing access to the records of a base cluster either directly or through an alternate index. (4) In an online IMS/VS system, the route a message takes from the time it is originated through processing; in a multisystem environment, the route can include more than one IMS/VS system.

PCDD. Pseudo common data set descriptor (CDD) record.

PDA. Problem Determination Aid.

PDF. Program Development Facility.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

PE. Physical end.

peer-to-peer remote copy (PPRC). A hardware-based remote copy option that provides a synchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

performance scaling. Formatting a tape so that less than its full capacity is used. Formatting a tape to its optimal performance capacity decreases the amount of time required to access the data on the tape. A tape that is formatted to its optimal performance capacity can later be reused and formatted to its full capacity.

physical data set restore. In DFSMShsm, the process of using a DFSMSdss function to restore one data set from a dump copy created by using the DFSMShsm full volume dump function.

physical end (PE). Signifies that the concurrent copy process has finished copying the data to the output device. Contrast with*logical end*.

physical sequential. See *sequential data set.*

pool of volumes. See *data set pool, general pool,* and *volume pool.*

PPRC. Peer-to-peer remote copy.

primary processing unit. In a multiple processing-unit-environment, the processing unit assigned to do level functions for backup and dump. Level functions for backup are: backup of CDS, move backup versions, and backup of migrated data sets. Level functions for dump are: delete expired dump copies, and delete excess dump VTOC copy data sets.

primary-processing-unit functions. (1) The level functions (backing up migrated data sets, deleting expired dump copies, and moving backup versions from level 1 volumes backup volumes) that must be performed by the primary processing unit. (2) See *level functions.*

primary volume. A non-SMS volume managed by DFSMShsm containing data sets that are directly accessible to the user.

problem determination aid trace. This data set is used to gather information about DFHSM processing.

promotion. The process of one host taking over level functions for another. See also *demotion.*

PSCB. Protected step control block.

PSP. Preventive service planning.

PSW. Program status word.

PTF. Program temporary fix.

Q

quiesce time. A time of day after which an automatic function does not start processing any more volumes.

R

RACF. Resource Access Control Facility.

recall. The process of moving a migrated data set from a level 1 or level 2 volume to a DFSMShsm-managed volume or to a volume not managed by DFSMShsm.

reconnection. In DFSMShsm, the process by which an unchanged data set recalled from a migration Level 2 tape is space managed by reestablishing access to the

migration copy from which it was recalled, rather than by moving the data set and creating a new migration copy.

record-level sharing (RLS). An extension to VSAM that provides direct shared access to a VSAM data set from multiple systems using cross-system locking.

recovery. In DFSMShsm, the process of copying a backup version of a data set from a backup volume to a specified volume or to the volume from which the backup version was created.

recycle process. A DFSMShsm process that, based on the percentage of valid data on a tape backup or migration level 2 volume, copies all valid data on the tape to a tape spill backup or migration level 2 volume.

reentrant. The attribute of a program or routine that allows the same copy of that program or routine to be used concurrently by two or more tasks.

Resource Access Control Facility (RACF). An IBM licensed program that provides access control by identifying and by verifying users to the system. RACF authorizes access to protected resources, logs unauthorized access attempts, and logs accesses to protected data sets.

restart data set. A data set created by DFSMShsm if aggregate recovery fails. It contains a list of all the data sets successfully restored during the aggregate recovery and allows the user to restart the aggregate recovery after the cause of the failure has been resolved.

restore. In DFSMShsm, the process of invoking DFSMSdss to perform the program's recover function. In general, it is to return to an original value or image, for example, to restore data in main storage from auxiliary storage.

restricted command authority. In DFSMShsm, allowing a user having at least read authority to each data set within an aggregate group to process ABACKUP or ARECOVER.

retired version. In DFSMShsm, a specially marked backup version that DFSMShsm created before it deleted the non-SMS-managed original data set during data set retirement.

retirement. See *data set retirement.*

REXX. Restructured extended executor.

RLS. Record level sharing.

roll off. When a new backup, dump, or generation data set copy is created, the oldest copy becomes a candidate for deletion, to maintain a customer-specified limit.

RPL. Request parameter list. Part of VSAM control block.

RRDS. Relative record data set.

S

SAF. System authorization facility.

SAQ. Supplement for additional quantities.

SCP. System control programming.

SDSP. Small data set packing.

secondary address space. Synonym for DFSMShsm secondary address space.

selection data set. In aggregate backup and recovery processing, a sequential data set or a member of a partitioned data set used to define the data sets that compose the input to the aggregate backup function. It contains any include, exclude, accompany, or allocate lists.

sequential data set. A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

similar device. A (DASD) device with the same number of bytes per track and tracks per cylinder.

single-cluster control data set. A control data set (MCDS or BCDS) that is represented by a single VSAM key-sequenced data set (KSDS) cluster. This can be a key-range or non-key-range cluster.

single-file format. In DFSMShsm, a 3480 or 3490 format, or the equivalent, consisting of one standard-label data set that spans up to 255 tape volumes.

single DFSMShsm-host environment. Any environment in which a single DFSMShsm host has exclusive use of a set of control data sets.

small data set packing (SDSP). In DFSMShsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

small-data-set-packing data set. In DFSMShsm, a VSAM key-sequenced data set allocated on a migration level 1 volume and containing small data sets that have migrated.

SMF. System Management Facilities.

SMP. System Modification Program.

SMP/E. System Modification Program Extended.

SMS. Storage Management Subsystem.

SMS class. A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or availability (management class) needs.

SMS-managed data set. A data set that has been assigned a storage class.

space manager. Synonym for storage administrator.

space management. In DFSMShsm, the process of managing aged data sets on DFSMShsm-managed and migration volumes. The three types of space management are: migration, deletion, and retirement.

spanning data sets. Data sets that span one or more tape volumes. The tape volumes that are spanned are part of a connected set. See also *connected set*.

specific scratch pool. A group of empty tapes with unique serial numbers that are known to DFSMShsm as a result of being defined to DFSMShsm with the ADDVOL command.

spill backup volume. A volume owned by DFSMShsm to which all but the latest backup version of a data set are moved when more space is needed on a DASD daily backup volume or all valid versions are moved when a tape backup volume is recycled.

spill process. A DFSMShsm process that moves all but the latest backup version of a data set from a DASD daily backup volume to a spill backup volume.

SSF. Software Support Facility.

SSI. Subsystem interface.

SSM. Secondary space management.

SSSA. Subsystem option block extension for SMS.

storage administrator. In DFSMShsm, the person who is authorized through the DFSMShsm AUTH command to issue DFSMShsm system programmer and storage administrator commands, who can affect the authority of other DFSMShsm userq, and who contrmls the ways DFS Shsm manages DASD space.

storage class. A named list of data set storage service attributes that identifhes perfmrmance and availability requirements. SMS uses these attributes to control data placement.

storage group. (1) A named list of DASD volumes used for allocation of new SMS-managed data sets, or a dummy storage group, or a VIO storage group. (2) A list of real DASD volumes, or a list of serial numbers of volumes that no longer reside on a system but that end users continue to refer to in their JCL.

storage hierarchy. An arrangement in which data may be stored in several types of storage devices that have different characteristics such as capacity and speed of access.

Storage Management Subsystem (SMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

suballocated file. A VSAM file that occupies a portion of an already defined data space. The data space may contain other files. Contrast with *unique file.*

subsystem interface (SSI). The means by which system routines request services of the master subsystem, a job entry subsystem, or other subsystems defined to the subsystem interface.

SVC. Supervisor call instruction.

sysplex. A set of MVS or z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. This term is derived from *system complex*. See also *base sysplex* and *parallel sysplex.*

sysplex timer. An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides.

system-managed storage. An approach to storage management in which the system determines data placement and an automatic data manager handles data backup, movement, space, and security.

T

take back. Process performed by a demoted host to take back the level functions that have been taken over by a promoted host.

TCB. Task control block.

threshold of occupancy. A limit of occupied space on a volume managed by DFSMShsm.

time sharing option (TSO). An option on the operating system for a System/370 that provides interactive time sharing from remote terminals.

TIOT. Task input/output table.

TMP. Terminal monitoring program.

TOD. Time of day.

trace. (1) A record of the execution of a computer program that exhibits the sequence in which the instructions were executed. (2) To record a series of events as they occur.

TSN. Time sharing option.

TSO/E. Tile sharing option/extended.

TTOC. Tape table of contents record.

TVT. Tape volume table.

U

UCB. Unit control block.

UID. The DFSMShsm authorized-user identification in 1 to 7 characters.

undirected recall. In DFSMShsm, moving a migrated data set from a level 1 or level 2 volume to a level 0 volume without specifying the target volume or unit where the volume can be allocated. Undirected recall can be automatic or by command.

unique file. A VSAM file that occupies a data space of its own. The data space is defined at the same time as the file and cannot contain any other file. Contrast with *suballocated file.*

unlike device. A DASD device with a different number of bytes per track and tracks per cylinder, or both.

user catalog. An optional catalog used in the same way as the master catalog and pointed to by the master catalog. Use of user catalogs lessens the contention for the master catalog and facilitates volume portability.

user exit. A programming service provided by an IBM software product that may be requested during the execution of an application program that transfers control back to the application program upon the later occurrence of a user-specified event.

V

VAC. Volume activity count record.

virtual DASD. In DFSMShsm, this refers to the 3850 Mass Storage System (MSS).

virtual storage access method (VSAM). An access method for indexed or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

virtual storage constraint relief (VSCR). A function that increases the amount of storage available for the user's application program.

volume. (1) A certain portion of data, together with its data carrier, that can be handled conveniently as a unit. (ISO) (2) A data carrier that is mounted and demounted as a unit, for example, a reel of magnetic tape, a disk pack. (ISO) (3) That portion of a single unit of storage that is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (4) A storage medium that is mounted and demounted as a unit; for example, magnetic tape or diskette.

volume pool. In DFSMShsm, a set of related primary volumes. When a data set is recalled, if the original volume that it was on is in a defined volume pool, the data set can be recalled to one of the volumes in the pool.

volume serial number. An identification number in a volume label that is assigned when a volume is prepared for use in the system.

volume table of contents (VTOC). (1) A table on a direct access volume that describes each data set on the volume. (2) An area on a disk or diskette that describes the location, size, and other characteristics of each file and library on the disk or diskette.

VSA. VSAM data set allocation control block.

VSAM. Virtual storage access method.

VSAM sphere. A sphere containing the following eight components: base cluster, base data object, base index object, base path, alternate index, alternate index data object, alternate index index object, and alternate index path.

VSCR. Virtual storage constraint relief.

VSR. Volume statistics record.

VTOC. Volume table of contents.

VTOC copy data set. A data set that contains a copy of the VTOC entry for each data set that DFSMShsm backs up or dumps.

VTS. Virtual tape server.

VVDS. VSAM volume data set.

W

WPCDD. ABACKUP/ARECOVER pseudo CDD parameter list.

WWFSR. ABACKUP/ARECOVER backup and recovery record.

X

XCF. Cross-system coupling facility.

XRC. Extended remote copy.

Index

Special characters

GDG ELEMENTS ON PRIMARY
  attribute 27

A

ABA (aggregate backup assist) 324
ABACKUP
  command 287
  examples 314
  HOLD command, parameter of 543
  invoking 311
  issuing with the EXECUTE
    parameter 315
  optional parameters 312
  parameters
    EXECUTE 311
    FILTEROUTPUTDATASET 312
    MOVE 312
    NOSTACK 308
    OPTIMIZE 308, 311, 312
    STACK 308
    UNIT 312
    VERIFY 311, 312
  RELEASE command, parameter
    of 547
  required parameters 311
  syntax 311
  using with VERIFY parameter 314
ABARS (aggregate backup and recovery
  support)
  account field for aggregate
    groups 319, 376
  activity log, backing up and
    recovering 360
  activity record
    creating 338
    expiring 320
  backup error exit, ARCBEEXT 306
  CPU processing time, tracking 319,
    376
  defining aggregate group and
    management class to DFSMS 283
  DFSMSdss 283
  expiration date exit, ARCEDEXT 306
  functions
    held, querying 370
    held, releasing 370
    holding 369
  general description 9
  ICF catalog and catalog alias
    processing 323, 371
  instruction data set, recovering 359
  interaction with other products 281
  ML2 data set exit, ARCM2EXT 306
  overview 281
  RACF 283
ABARSACTLOGMSGLVL parameter,
  SETSYS command 308
ABARSBUFFERS parameter, SETSYS
  command 307
ABARSDELETEACTIVITY parameter,
  SETSYS command 318
ABARSOPTIMIZE parameter, SETSYS
  command 308
ABARSPROCNAME parameter, SETSYS
  command 307, 341
ABARSTAPES parameter, SETSYS
  command 307, 342
ABEND (abnormal end)
  restarting DFSMShsm after 562
Access Method Services
  considerations 135
accessibility 641
accessing a data set when profile cannot
  be recovered 432
ACCOMPANY
  data set 285, 295
  keyword 295
ACCOMPANYEXCLUDE
  data set 285
  keyword 295
accounting information, ABARS CPU
  time 319, 376
activity log
  aggregate backup output 287, 317
  aggregate recovery output 373
  SETSYS command
    ACTLOGTYPE parameter 307
ACTIVITY parameter, ARECOVER
  command 352
additional considerations for
  availability management 413
  backup 413
  dump 424
  recall 68
ADDVOL
  adding aggregate recovery volumes to
    DFSMShsm 340
  defining dump volumes to
    DFSMShsm 162
  level 0 volumes for automatic backup,
    non-SMS 262
  level 0 volumes for space
    management, non-SMS 82
  level 0 volumes to be dumped,
    specifying 263
  migration level 2 volumes 39
  optional parameters
    AUTOBACKUP 262
    AUTODUMP 264
    AUTOMIGRATION 83
    AUTORECALL 84
    BACKUPDEVICECATEGORY 84
    DAILY 177
    DELETEBYAGE 84
    DELETEIFBACKEDUP 84
    DRAIN 493
    DUMPCLASS 162
    MIGRATE 84
ADDVOL (continued)
  optional parameters (continued)
    THRESHOLD 84
  PRIMARY, pool configuration in
    JES3 88
  required parameters
    BACKUP 177
    DUMP 162
    PRIMARY 88, 262
ADMIN OR USER COMMAND
  BACKUP attribute 168
age of records for migration cleanup,
  specifying 38
aggregate backup
  aggregate group, example 287
  assist 324
  checklist 289
  comprehensive ABACKUP command
    authority 304
  creating selection data sets 291
  data file, description 287
  data sets dynamically allocated
    during 310
  defining SETSYS parameters 306
  determining unit type for control and
    data files 309
  ensuring that the volumes are
    online 306
  identifying applications to be backed
    up 284
  input for recovery 287
  installation-wide exits 306
  instruction/activity log file,
    description 287
  minimizing data set contention 309
  multiple ML2 data sets with single
    tape mount 316
  of data sets 311
  output 316
  overview 284
  performing 310
  performing actual backup of data
    sets 315
  preparing for 289
  RACF checking 304
  restricted ABACKUP command
    authority 305
  selection data set 324
  sequential data striping 396
  SMS requirements 290
  specifying applications to be backed
    up 284
  tape files created 316
  tape volumes, RACF 303
  using NetView file transfer
    program 391
  verification 311
aggregate backup and recovery record
  creating 338
  description of 338
  expiring 320

© Copyright IBM Corp. 1984, 2004

aggregate backup and recovery record *(continued)*
recording CPU time and accounting codes 376
aggregate group
AGGREGATE GROUP NAME parameter 311
Aggregate Group Name, ISMF panel parameter 299
backup, using ISMF panels 376
defining a group 298
example 300
overview 284
panel parameter 299
AGGREGATE parameter, ARECOVER command 351
aggregate recovery
ABARSPROCNAME parameter, SETSYS command 341
ARECOVERUNITNAME parameter, SETSYS command 342
checklist 332
comprehensive ARECOVER command authority 333
data set name conflict resolution 361
data sets dynamically allocated during 372
defining SETSYS parameters 341
determining site constraints 336
example
instruction data set 337
recovered data sets 375
HOLD command, parameter of 544
installation-wide exits 344, 358
MAXABARSADDRESSSPACE parameter, SETSYS command 341
model entity 344
model entity (MENTITY) parameter, ARECOVER command 353
output 329, 372
overview 328
performing 350
recovery of data sets 358
verification 358
preparing for 331
RACF checking during 333
recovered data sets 330
recovery
of data sets 350
site, example 330
RELEASE command, parameter of 547
restart data set 359
restricted ARECOVER command authority 334
restrictions to adding volumes to DFSMShsm 340
retrieving the instruction data set 336
secondary address space 341
sequential data striping 397
SETSYS command examples 343
SMS requirements 334
ALL parameter
DEFINE command 339
HOLD command 543
RECYCLE command 497
ALL parameter *(continued)*
RELEASE command 547
TAPECOPY command 404
UPDATEC command 471
ALLOCATE
data set 285, 296
keyword 296
list 285
processing VSAM data sets in list 348
ALLOCATEEXCLUDE
data set 285
keyword 296
allocation
ALLOCATE keyword 296
dynamic data set, during aggregate backup processing 310
dynamic data set, during aggregate recovery processing 372
recalled data sets to volumes, non-SMS-managed 103
ALLOCATION/MIGRATION THRESHOLD attribute 22
ALTERDS 267
alternate tape volume, replacing an original with 407
alternate tapes, sending off site 408
ALTERNATEVOLUMES parameter, TAPECOPY command 406
always-call
environment, definition 575
RACF indicator with 575
support 575
support for migrated data sets and backup copies 579
ANYSTORAGEVOLUME subparameter, RECALL parameter, SETSYS command
defining volumes eligible for recall selection 82
general pool 87
pool configuration 87
pool configuration in JES3 88
application programs, requesting DFSMShsm service 515
ARCADEXT, data set deletion exit 96
ARCBDEXT, data set backup exit 189
ARCBEEXT, ABARS backup error exit 306
ARCBJRNL
journal copy utility 463
return codes 468
ARCCATGP, RACF logon group 68
ARCCBEXT, CDS backup exit 453
ARCCREXT
in ABARS ARECOVER support 371
in aggregate recovery 330, 344
in RACF facility class authorization checking 367
installation exit 366
ARCEDEXT, ABARS expiration date exit 306
ARCIMPRT
import utility 463
return codes 468
ARCINBAK program, inline backup 415
ARCM2EXT, ABARS ML2 data set exit 306
ARCMDEXT, data set migration exit 78, 100
ARCMMEXT, second-level migration data set exit 48
ARCPEDIT program
printing DFSMShsm log information 553
ARCPRLOG program
printing DFSMShsm log information 553
sample list 559
ARCSAEXT, space management and backup exit 51
ARCTDEXT, tape data set exit 48
ARCTVEXT tape volume exit 403, 408
ARECOVER
and the duplex tape option 403
catalog considerations 349
conflict resolution, order of processing 367
examples 356
examples with
EXECUTE parameter 358
PREPARE parameter 357
VERIFY parameter 358
invoking command 350
optional parameters
BYPASS 353
DATASETCONFLICT 353
DATE 352
MENTITY 344, 353
MIGRATEDDATA 353
ML1 353
ML2 353
NOBACKUPMIGRATED 354
NOSTACK 313, 352
PERCENTUTILIZED 354
RECOVERNEWNAMEALL 354
RECOVERNEWNAMELEVEL 354
RENAMESOURCE 353
RENAMETARGET 353
REPLACE 353
SOURCELEVEL 354
STACK 313, 352
TARGETUNIT 342, 355
TGTGDS 343, 350, 355
UNIT 352
VERSION 353
performing verification 350
required parameters
AGGREGATE 351
DATASETNAME 351
EXECUTE 350
PREPARE 357
VERIFY 350
VOLUMES 351
XMIT 351
syntax 350
ARECOVERPERCENTUTILIZED parameter, SETSYS command 342
ARECOVERTGTGDS parameter, SETSYS command 343
ARECOVERUNITNAME parameter, SETSYS command 342
ARPOOL parameter, DEFINE command 338
assumptions 16

attributes
GDG ELEMENTS ON PRIMARY 27
ADMIN OR USER COMMAND BACKUP 168
ALLOCATION/MIGRATION THRESHOLD 22
AUTO BACKUP
management class 169
storage group 163
AUTO DUMP 152
AUTO MIGRATE 22
AUTOBACKUP 84
AUTODUMP 84
AUTOMIGRATION 84
AUTORECALL 84
BACKUP FREQUENCY 165
BACKUP SYSTEM/SYSPLEX NAME 163
BACKUPDEVICECATEGORY 84
COMMAND OR AUTO MIGRATE 27
DUMP CLASS 152
DUMP SYSTEM/SYSPLEX NAME 153
EXPIRE AFTER DATE/DAYS 25
EXPIRE AFTER DAYS NON-USAGE 25
GDG management, specifying 27
GUARANTEED BACKUP FREQUENCY 165
LEVEL 1 DAYS NON-USAGE 27
MIGRATE SYSTEM/SYSPLEX NAME 25
migration, specifying 27
NUMBER OF BACKUP VERSIONS (DATA SET DELETED) 166
NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) 166
of primary volumes 84
PARTIAL RELEASE 26
PRIMARY DAYS NON-USAGE 27
recall 85
RETAIN DAYS EXTRA BACKUP VERSION 167
RETAIN DAYS ONLY BACKUP VERSION 167
ROLLED-OFF GDS ACTION 28
AUDIT
ensuring that CDSs are accurate 475
parameters
COMMONQUEUE 115
DATASETCONTROLS 472
HOLD 544
NOFIX 472
RELEASE 548
REPORT 472
SYSOUT 472
authority
comprehensive ABACKUP command 304
comprehensive ARECOVER command 333
needed to perform DFSMShsm RACF functions 577
restricted ABACKUP command 305
restricted ARECOVER command 334
authorization
checking
during HBDELETE processing 223
during profile recovery 431
recovery of RACF-indicated data sets 429
AUTO BACKUP attribute
management class 169
storage group 163
AUTO DUMP attribute 152
AUTO MIGRATE attribute 22
AUTO subparameter
BACKUP parameter
HOLD command 544
RELEASE command 548
DUMP parameter
HOLD command 544
RELEASE command 548
MIGRATION parameter
HOLD command 546
RELEASE command 551
AUTOBACKUP subparameter, PRIMARY parameter, ADDVOL command 262
AUTOBACKUPSTART parameter, SETSYS command 172
AUTODUMP subparameter, PRIMARY parameter, ADDVOL command 264
automatic
availability management of non-SMS-managed volumes 265
availability management operation 178
backup
example 206
preventing 419
specifying when to start 172
specifying whether to perform 169
class selection, function of 8
dump processing
specifying when to start 153
full-volume dump, example system 151
functions, running concurrently 563
hold, release 551
incremental backup, example system 151
interval migration, example system 22
primary space management, example system 21
recall
data sets migrated for extent reduction 67
during data set allocation 67
during non-VSAM data set opening, non-SMS-managed 103
during VSAM data set opening, non-SMS-managed 103
example system 22
for command data set scratching, non-SMS-managed 104
for command deletion if VSAM AIX clusters 67
for data set renaming 67
SMS-managed 65
automatic *(continued)*
recall *(continued)*
with DFSMSdss 67
restart of primary space management 142
restart of secondary space management 142
reuse of dump tapes, controlling 156
SMS-managed volumes 151
tasks for availability management 152
automatic space management
candidate data sets for backup, determining, non-SMS-managed volume 93
candidate data sets for backup, determining, SMS-managed storage 53
candidate data sets for extent reduction, determining, non-SMS-managed volume 94
candidate data sets for extent reduction, determining, SMS-managed storage 53
candidate data sets for migration, determining, non-SMS-managed volume 93
candidate data sets for migration, determining, SMS-managed storage 52
candidates for both migration and extent reduction, migrate, SMS-managed storage 55
candidates for extent reduction, migrate, SMS-managed storage 56
compaction during 58
defining a new start window 144
deleting expired data sets, SMS-managed storage 51
deleting temporary data sets, SMS-managed storage 51
determining candidate data sets for backup, non-SMS-managed volume 93
determining candidate data sets for backup, SMS-managed storage 53
determining candidate data sets for extent reduction, non-SMS-managed volume 94
determining candidate data sets for extent reduction, SMS-managed storage 53
determining candidate data sets for migration, non-SMS-managed volume 93
determining candidate data sets for migration, SMS-managed storage 52
DFSMShsm-managed volumes, SMS-managed storage 50
example management classes
GDS management class 29
large management class 30
no-migration management class 30
standard management class 29
functions 3

automatic space management *(continued)*
general description 41
migrate candidates for both migration and extent reduction, SMS-managed storage 55
migrate candidates for extent reduction, SMS-managed storage 56
migrate data sets, SMS-managed storage 56
operation 41
out of space condition 60
Phase 1
processing example, SMS-managed storage 54
SMS-managed storage 50
Phase 2
SMS-managed storage 55
priority order for migration 56
processing for non-SMS-managed storage 90
processing for SMS-managed storage 41
purpose 3
releasing unused space, SMS-managed storage 54
sort data sets in priority order for migration, SMS-managed storage 56
specifying the number of secondary space management tasks 35
specifying when to start primary space management 31
specifying when to start secondary space management 31
tasks 22
VSAM data sets 58
AUTOMIGRATION
parameter, HOLD command, preventing processing 131
subparameter, ADDVOL command 83
AUTORECALL subparameter, ADDVOL command 84
AUTOREUSE subparameter, DUMPCLASS parameter, DEFINE command 156
auxiliary host 539
availability management
automatic operation 178
automatic, non-SMS-managed volumes 265
automatic, SMS-managed volumes 151
backup 147
command 207
command, example system 151
command, for non-SMS-managed storage 267
considerations
extended and compressed VSAM data sets 434
extended SAM data sets 434
sequential data striping 434
defining a new start window 439
dump 148
dump tasks 152
availability management *(continued)*
functions 4, 147
how to do 147
non-SMS-managed storage 261
other considerations 413
procedures 437
purpose 4
security considerations 429
SMS-managed storage 151
tasks for automatic 152

B

BACKDS command
UNIT parameter 268
VOLUME parameter 268
backing up
and recovering the ABARS activity log 360
applications 286
backup
additional considerations 413
all changed data sets on a volume, non-SMS-managed 268
all data sets on a volume, SMS-managed 215
all eligible changed data sets
in one or more storage groups 216
on a volume 216
all primary volumes, non-SMS-managed 268
automatic
example 206
example system 151
specifying storage for 163
specifying whether to perform 169
when to start, specifying time 172
cleanup process 205
conditions for a data set, changing 267
control data set (CDS)
examples 475
how to 448
manual 460
processing 183
protecting tapes 453
recovery and multiple backup data sets, using UPDATEC 469
specifying DASD for 452
specifying device type 451
specifying how many backup versions to keep 450
specifying tape for 452
when accessing in RLS mode 184, 448, 452, 459
copies
always-call support 579
of DFSMShsm-owned tape volumes 401
of SMS-managed data sets at recovery site 399
current, definition 95
cycle
defining when to start 176
specifying 176
backup *(continued)*
daily volume selection
DASD 193
DASD and tape 197
initial DASD 194
initial DASD or tape 198
subsequent DASD 195
subsequent DASD or tape 199
daily, selecting 191
DASD spill volume 203
data sets
all changed, in one or more storage groups 215
all changed, on a volume 216
by command, specifying who can 168
data sets for CDSs
changing names 457
specifying names 451
determining candidate data sets, non-SMS-managed volumes 93
DFSMShsm-managed volumes from storage groups 186
disaster, facilities 402
eligible data sets 187
expired versions of non-SMS-managed data sets, deleting 269
frequency
non-SMS-managed, specifying 261
relationship to maximum number of backup versions 167
SMS-managed, guaranteed backup 165
SMS-managed, specifying 165
from DASD, RACF considerations 580
function 147
ICF catalog, procedure 440
individual data set 208
individual data set, non-SMS-managed 268
initial DASD daily volume selection 194
initial selection, DASD or tape daily 198
inline backup
about 415
invoking 416
non-SMS-managed 268
SMS-managed 215
interrupting and restarting 437
management classes, example 169
migrated data sets 185
migrated data sets, specifying first qualifier 37
name of
cartridge-type single-file data sets 190
dump copy data sets 181
versions of data sets 189
VTOC copy data sets 187
obtaining information 518
overview
non-SMS-managed 12
SMS-managed 8
preventing 419

backup *(continued)*
procedures 437
processing example 182
profile
authorization checking during recovery 429
deleting 191
parameters used in creating 191
RACF discrete, for cataloged data sets 190
transferring from one backup version to another 223
RACF-indicated data sets 581
recovering from a damaged migration level 1 volume 510
redefining the cycle 415
relationship of maximum number of versions to frequency 167
releasing 206
SDSP 512
selecting volumes 201
specified data sets, delete 222
specifying
first qualifier for names of versions 173
how groups of data sets are to be managed 164
how long to keep versions, SMS-managed 167
kind of device to use 173
maximum number of tasks in each processor 174
maximum number of versions to keep, SMS-managed 166
number of versions to keep, non-SMS-managed 262
only changed data sets 175
volumes 262
whether to perform 173
spill 200
spill volume selection
initial 202
subsequent 203
storage groups with automatic backup attribute 186
subsequent selection, DASD or tape daily 199
switching backup tapes 212
tape daily volume selection 196
initial 196
subsequent 197
tape spill volume 203
tasks 163
types 148
uncataloged data sets 265
versions
moving 184
of specific data sets, non-SMS-managed, deleting 270
volume
aggregate backup, RACF protection 303
contention 413
defining to DFSMShsm 177
deleting expired backup versions from 216
initial selection, spill 202
backup *(continued)*
volume *(continued)*
not managed by DFSMShsm 268
recovery combined with full-volume restore 251
releasing a backup 206
restore directly with DFSMSdss 427
selection 201
spill 249
subsequent selection, spill 203
types 148
wait for spill backup volume 203
VSAM data sets 188
VTOC copy data set, create 186
wait for spill volume 203
backup data sets, changing names 457
BACKUP FREQUENCY attribute 165
BACKUP parameter
ADDVOL command 177
DEFINE command 176
INUSE subparameter 176
QUERY command 527
RECYCLE command 498
RELEASE command 548
SETSYS command 173
TAPECOPY command 404
BACKUP SYSTEM/SYSPLEX NAME attribute 163
backup tape, missing 608
backup tape, overwritten 594
backup version
creating while data set open for update 175
maximum number per data set 262
BACKUPCONTROLDATASET parameter, UPDATEC command 471
BACKUPCOPIES subparameter, CDSVERSIONBACKUP parameter, SETSYS command 450
BACKUPDEVICECATEGORY parameter, ADDVOL command 84
BACKUPDEVICECATEGORY subparameter, CDSVERSIONBACKUP parameter, SETSYS command 451
BACKUPPREFIX parameter, SETSYS command 173
BACKUPVOLUME parameter, LIST command 518
BACKVOL
optional parameters
DUMP 207
FREQUENCY 269
INCREMENTAL 216
RETENTIONPERIOD 207
TOTAL 215
VERSIONS 269
PRIMARY parameter
interrupting and restarting backup processing 438
interrupting and restarting dump processing 443
redefining the backup cycle 415
required parameters
CONTROLDATASETS 460
PRIMARY 268, 269
VOLUME 207
base cluster 237
base data component 237
base index component 237
BCDS scan, EXPIREBV processing
determining starting and ending point 218
BCDSBACKUPDSN subparameter, CDSVERSIONBACKUP parameter, SETSYS command 451
BDELETE command
backup volume cleanup process 204
FROMVOLUME parameter 270
VERSIONS parameter 270
bypass enqueue processing between dump and DFP functions 426
BYPASS subparameter, DATASETCONFLICT parameter, ARECOVER command 353, 362

C

CANCEL
DFSMShsm queued requests 552
candidate data sets
extent reduction, migrate, SMS-managed storage 56
for backup, determining, SMS-managed storage 53
for both migration and extent reduction, migrate, SMS-managed storage 55
for extent reduction, determining, SMS-managed storage 53
for migration, determining, SMS-managed storage 52
capacity utilization
specifying 502
cartridge-type tape volumes
making from reel-type or multiple-file-format tapes 501
case 1: damaged CDS, full journal 585
case 10: restoring stacked output tapes with a down-level MVS image 606
case 11: correcting errors within the common recall queue 607
case 12: recovering a deleted ML1 data set without a backup 608
case 2: damaged journal, undamaged CDS 588
case 3: full journal, undamaged CDS 589
case 4: structurally damaged CDS, missing journal records 589
case 5: time sequence gaps in CDS records, journal records also missing 589
case 6: overwritten migration tape 590
case 7: overwritten backup tape 594
case 8: Damaged ML1 volumes 598
case 9: reestablish access to previously deleted migrated data sets (no backup exists, ml2 only) 604
catalog
considerations
ARECOVER processing 349
migration processing 61
inline backup of data sets 415

catalog *(continued)*
RACF maintenance group 68
cataloged data sets, backing up RACF discreet profiles for 190
CATALOGEDDATA subparameter, NONSMSVERSIONS parameter, EXPIREBV command 269
categories of DFSMShsm-owned tape volumes, copying 404
CDS (control data set)
accuracy, ensuring 475
backing up 183, 450
determining data mover being used 459
manually 460
specifying DASD 452
specifying tape 452
specifying the data mover 448
using concurrent copy 449
when accessing in RLS mode 184, 448, 452, 459
backup exit ARCCBEXT 453
backup tapes, protecting 453
determining if accurate 472
device type for backup, specifying 451
displaying the CDS serialization technique 459
displaying the key boundaries of non-key-range multicluster CDSs 459
dumping 424
maintaining 447
management class support 449
migration
recovering using multiple DASD backup data sets 475
recovering using multiple tape backup data sets 479
multicluster considerations 456
multiple backup data sets
DASD 453
description 450
tape 455
names of backup data sets
changing 457
specifying 451
obtaining information from 517
protecting backup tapes 453
recovered, updating 471
recovery
description 462
examples 475
multiple backup data sets, using UPDATEC 469
recovery functions 463
enhanced, implementing using JCL 464
enhanced, requirements for using 463
repairing errors 473
specifying how to back up 448
tasks for maintaining 447
VSAM considerations 472
CDS damaged, full journal 585
CDS damaged, missing journal records 589
CDS records with time sequence gaps, journal records missing 589
CDS undamaged, full journal 589
CDS undamaged, journal damaged 588
CDSVERSIONBACKUP parameter, SETSYS command 448
recovering the CDSs using UPDATEC 469
changed data set on volumes, non-SMS-managed, backing up all 268
CHANGEDONLY subparameter, INCREMENTALBACKUP parameter, SETSYS command 175
changes
for SMS, command function 569
in storage group, recall problems with GUARANTEED SPACE VSAM data sets 136
in storage group, recover problems with GUARANTEED SPACE VSAM data sets 421
in storage-class-to-data set mapping, recall problems with GUARANTEED SPACE VSAM data sets 137
in storage-class-to-data set mapping, recover problems with GUARANTEED SPACE VSAM data sets 422
changing
backup conditions for a data set 267
names of backup data sets 457
characteristics of dump classes, specifying 156
characteristics of recovered profile 431
checking integrity age, non-SMS-managed data sets 92
checking, authorization during profile recovery 431
checklists, aggregate backup 289
checkpointed data sets
and extent reduction 53, 164
and migration eligibility 52, 164
DFSMShsm patch to force migration 51, 52, 164
classes
dump, specifying characteristics 156
cleanup
backup volume 204
processing 204
closing and processing the activity logs 550
coexistence considerations
DFSMShsm ABACKUP output data sets 320
DFSMShsm aggregate groups and management classes 302
command
ABACKUP authority
comprehensive 304
restricted 305
access with RACF FACILITY class 582
ARECOVER authority
comprehensive 333
restricted 334
command *(continued)*
availability management
example system 151
non-SMS-managed storage 267
SMS-managed storage 207
function changes for SMS 569
migrating
individual data sets 73
SMS-managed volume 74
recalling
direct access data sets 75
SMS-managed data sets 75
recovery and restore 224
space management
all primary volumes, non-SMS-managed 107
example system 22
non-SMS-managed data 106
SMS-managed data 71
volume space management, non-SMS-managed 107
COMMAND OR AUTO MIGRATE attribute 27
commands
ABACKUP 287
ADDVOL
for backup volumes 177
for dump volumes 162
for ML2 volumes 39
for primary volumes 82
ALTERDS 267
AUDIT 472
BACKDS 208, 268
backup
preventing processing 419
BACKVOL 215, 268
COPYPOOL 519
DELETE 72
deleting migrated data sets 71
FRBACKUP 240, 244
FRDELETE 244
FRRECOV 242, 244
HBACKDS 208, 415
HDELETE 72
HOLD 244, 419, 543
HRECALL 75
LIST 517
long-running 570
MIGRATE 73, 489
purpose of DFSMShsm 567
QUERY 526
RECALL 75
RECYCLE 497
TAPECOPY 402
TAPEREPL 407
to directly manage data sets or volumes 568
to operate DFSMShsm 569
UPDATEC 471
common recall queue 109
connecting to list structure 111
deleting the structure 114
disconnecting from 113
failures
coupling facility link 115
coupling facility structure 114
full list structure 115

common recall queue *(continued)*
failures *(continued)*
primary address space 115
z/OS image 115
populating 111
processing
recall requests 113
shutdown 113, 543
selecting requests on 112
common recall queue (CRQ) 607
common volumes, example 15
COMMONQUEUE 111, 115
compaction
during migration 58
specifying utilization on a 3480X 501
complete volume, restoring from a dump copy 254
comprehensive ABACKUP command authority 304
comprehensive ARECOVER command authority 333
concurrent copy
specifying for volume dumps 154
specifying whether to use 164
using to back up the CDSs 449
with data set backup 423
with RACF 423
conditions for a data set, backup, changing 267
configurations
general pool 87
pool, commands that affect in JES3 87
conflict resolution data set 330, 362
connected set processing 496
considerations
additional, for recall 68
availability management
extended and compressed VSAM data sets 434
extended SAM data sets 434
sequential data striping 434
catalog, during ARECOVER processing 349
copy pool 247
DFSMShsm commands that affect pool configuration in JES3 87
dump 424
FRBACKUP 241
FRRECOV 243
multicluster CDS 456
multitasking, for SDSP data sets 128
sequential data striping
aggregate backup processing 396
aggregate recovery processing 397
volume pooling 89
VSAM, for control data sets 472
consolidating valid data on tape volumes 495
contention
backup volume 413
migration needs volume 117
multiple-processing-unit environment 121
recycle needs volume 119
SDSP data sets 125
contention *(continued)*
single-processing-unit environment 117
control file
aggregate backup output 316
input for recovery 329
conversion of data sets to GUARANTEED SPACE
recall problems 138
recover problems with VSAM data sets 422
conversion of non-SMS-managed volume to SMS-managed 7
CONVERT parameter
in direct-to-tape environment 485
MIGRATE command
non-SMS-managed data sets 484
SMS-managed data sets 484
with BDAM data sets 485
converting partial tapes to full tapes 407
copies
of DFSMShsm-owned tape volumes 402
Copies, ISMF panel parameter 299
copy pool
considerations 243, 247
decreasing volumes 248
deleting unneeded volumes 244
increasing volumes 247
information, obtaining 245, 519
recovery of 242, 249
setup 238
using 238
copy pool backup storage group type 239
copy pool construct 238
copying
categories of DFSMShsm-owned tape volumes 404
individual volumes, disaster backup 406
CPU processing time, tracking for ABARS 319, 376
create a VTOC copy data set for the volume 186
creating
ABARS activity record 338
backup profiles, parameters used 191
selection data sets 291
current backup version, definition 95
cycle
backup 176
dump 155
redefining backup 415
start date
backup 176
dump 155
CYCLESTARTDATE subparameter
BACKUP parameter, DEFINE command 176
DUMPCYCLE parameter, DEFINE command 155

D

daily
backup volume selection
DASD 193
DASD, initial 194
DASD, subsequent 195
tape 196
tape, subsequent 197
initial DASD or tape backup volume selection 198
limit
not reached, DASD or tape daily backup volume selection 199
reached or exceeded, DASD or tape daily backup volume selection 198
subsequent DASD or tape backup volume selection 199
DAILY parameter, RECYCLE command 497
damaged CDS, full journal 585
damaged CDS, missing journal records 589
damaged journal, undamaged CDS 588
damaged ML1 volume 598
damaged ML1 volume, recovering backup versions from 510
DAOPTION parameter, RECALL command 75, 108
DASD (direct access storage device)
and tape daily backup volume, selecting 197
backing up control data sets to multiple backup data sets 453
changing from DASD to tape 459
changing from multiple tape backup data sets 458
choosing for multiple backup data sets 451
daily backup volume selection 193
daily backup volume selection, initial 194
daily backup volume selection, subsequent 195
I/O buffering technique to use for dump by DFSMSdss 154
migration level 2 volume, moving data sets to other migration level 2 volumes 492
migration volume, preventing further migration to 493
or tape, initial daily backup volume selection 198
or tape, subsequent daily backup volume selection 199
specifying for CDS backup 452
spill backup volume, subsequent selection 203
data class, use of 8
data file
aggregate backup output 316
description 287
input for recovery 329
data flow in DFSMShsm 5
data mover
DFSMSdss, specifying for CDS backup 448

data mover *(continued)*
DFSMShsm, specifying for CDS backup 450
data on dump volumes becomes invalid, specifying when 159
data set backup
inline backup 215
specifying tasks 209
switching tapes 212
taking a tape away 213
targeting to device types 213
using concurrent copy 214, 423
using the data set backup function 209
data sets
accessing when profile cannot be recovered 432
allocating recalled to volumes, non-SMS-managed 103
always open, backing up 176
automatic recall during opening VSAM, non-SMS-managed 103
automatic recall for command scratching, non-SMS-managed 104
backing up
individual, non-SMS-managed 268
migrated 185
serialization during 175
SMS-managed 208
backup
conditions for, changing 267
uncataloged 265
versions of specific data sets 222
VSAM 188
VTOC copy, create 186
changed
backing up all eligible in one or more storage groups 216
backing up all eligible on a volume, SMS-managed 216
backing up all, non-SMS managed 268
only, specifying whether to back up 175
command
migration 73
recall of direct access 75
recall of SMS-managed 75
commands to directly manage 568
contention, SDSP 125
control
backing up 450
DASD backup data sets 453
multiple backup data sets 450
multiple tape backup data sets 455
recovering, examples 475
recovery 462
recovery and multiple backup data sets, using UPDATEC 469
controlling migration, non-SMS-managed 107
deleting
expired dump copies 179, 222
from migration volumes with the MIGRATE command 490
data sets *(continued)*
deleting *(continued)*
migrated 71
migrated data set fails, JES3 105
deletion
ADDVOL command 84
description 73
exit ARCADEXT 95
non-SMS-managed volumes 95
partitioned data set 73
priority 116
determining
candidates for backup, non-SMS-managed volumes 93
candidates for extent reduction, non-SMS-managed volumes 94
candidates for migration, non-SMS-managed volumes 93
status 567
dump VTOC copy, write 181
dynamically allocated during aggregate backup processing 310
dynamically allocated during aggregate recovery processing 372
excess dump VTOC copy
deleting 182
uncataloging 182
expired backup versions
deleting from backup volumes 216
non-SMS-managed, deleting 269
non-SMS-managed, processing 269
from dump tapes, controlling restoring 157
generation, deferred 131, 428
inline backup 215
journal
DASD backup data sets 453
multiple backup data sets 450
multiple tape backup data sets 455
recovering, examples 475
replacing using multiple DASD backup data sets 478
migrated
deleting 71
profile backup for 190
recovering 506
uncataloging 582
migrating 489
migration exit ARCMDEXT 100
migration of VSAM, non-SMS-managed volumes 97
moving
between level 0 volumes, example 485
from a DASD ML2 volume to other ML2 volumes 492
from DASD backup volumes 493
from migration level 1 volumes 488, 490
from ML1 volumes 488
from one level 0 volume to other level 0 volumes 483
journal 474
with FREEVOL 488
data sets *(continued)*
name
backup versions 189
backup VTOC copy 187
conflict resolution, aggregate recovery 361
dump copy 181
dump VTOC copy 181
format of SDSP 512
migrated data sets 57
name of backup copy data sets 190
name of dump copy data sets 181
non-SMS-managed volumes
retirement 84, 95
obtaining information 518
on a volume, backing up 215
password protection on DASD 134
pools, definition 15
preventing processing during volume space management 131
priority order
non-SMS-managed 94
SMS-managed 56
RACF
discrete profile 574
discrete profiles for, backing up 190
generic profile 573
protection 573
RACF-indicated
backing up 581
deleting 133
recalling 134
recovering or restoring 430
recall
during opening non-VSAM, non-SMS-managed 103
non-SMS-managed 101
recalling to a specific volume, non-SMS-managed 108
recovered aggregate 330
recovered, maintaining password protection for 432
recovery
description 224
of direct access 232
recovery of extended format 232
VSAM 273
recovery and restore 149
restore
directly with DFSMSdss 427
from a DFSMShsm dump copy, non-SMS-managed 274
from a DFSMShsm dump copy, SMS-managed 235
individual from dump tapes, controlling 157
of a retired data set 276
SDSP
managing 511
removing 513
security 582
sometimes open, backing up 176
space management with discrete data set profiles 133

data sets *(continued)*
specific
delete backup versions 222
recovering, non-SMS-managed 271
specifying whether to back up 175
temporary, deleting 186
unmatched
backup 186
space management 51
VSAM, restore of 237
with discrete data set profiles, space management of 133
data sets, reestablishing access to deleted migrated 604
data-set-changed indicator
resetting 188
setting only when backup serialized 188
specifying whether to reset 158
DATASET parameter
ARECOVER command 351
LIST command 518
DATASETCONFLICT parameter, ARECOVER command 344, 353
DATASETCONTROL parameter, AUDIT command 472
DATASETNAME parameter
ARECOVER command 352
MIGRATE command 484
DATASETRESTORE subparameter, DUMPCLASS parameter, DEFINE command 157
date
hold volumes 104
prevent migration 98
DATE parameter, ARECOVER command 352
DATE parameter, RECOVER command 225
date-last-referenced field for recovery 428, 451
date-last-referenced field, maintenance and use for space management 61
date-of-scratch field, resetting, recovery processing 234
DAY subparameter, DUMPCLASS parameter, DEFINE command 157
day to dump data to a dump class, specifying 157
days on which dump is to occur, specifying 155
DAYS parameter, MIGRATE command 489
DAYS parameter, SETSYS command 81
DEBUG parameter, SETSYS command 131, 533
default management class 28
deferred generation data sets 131, 428
DEFINE
example 340
issuing the command 338
optional parameters
ALL 339
ARPOOL 338
AUTOREUSE 156
BACKUP 176
DEFINE *(continued)*
optional parameters *(continued)*
CYCLESTARTDATE 155
DATASETRESTORE 157
DAY 157
DISPOSITION 158
DUMPCLASS 156
DUMPCYCLE 155
FREQUENCY 158
L0VOLS 339
ML1VOLS 339
POOL 85
PRIMARYSPMGMTCYCLE 31
RESET 158
RETENTIONPERIOD 159
SECONDARYSPMGMTCYCLE 31
STACK 155, 160
TAPEEXPIRATIONDATE 159
UNIT 160
VOLUMEPOOL 88
VTOCCOPIES 161
pool configuration in JES3 88
defining
aggregate group 298
automatic backup window 439
automatic dump start window 443
automatic primary space management window 144
automatic secondary space management window 144
backup volumes to DFSMShsm 177
dump volumes to DFSMShsm 162
new processing window 144
parameters of the recycle process 496
pools of volumes 85
primary volumes to DFSMShsm 82
volume pool configuration in JES3 88
volumes for DFSMShsm 567
when backup cycle starts 176
definition
current backup version 95
data set pool 15
DELETE 73
DELETEBYAGE subparameter, ADDVOL command 84
DELETEBYAGE subparameter, VOLUME parameter, MIGRATE command 490
deleted migrated data sets 604, 608
DELETEIFBACKEDUP subparameter
ADDVOL command 84
NONSMSVERSIONS parameter, EXPIREBV command 269
VOLUME parameter, MIGRATE command 490
deleting
backup profiles 191
backup versions of specific data sets, non-SMS-managed 270
data set 73
excess dump VTOC copy data sets 182
expired backup versions from backup volumes, SMS-managed 216
expired backup versions of non-SMS-managed data sets 269
deleting *(continued)*
expired data sets, non-SMS-managed volumes 93
expired data sets, SMS-managed storage 51
list and utility data sets, non-SMS-managed volumes 93
list data sets, specifying age 80
migrated data sets 71
non-SMS-managed data sets from migration volumes with the MIGRATE command 490
RACF-indicated data sets 133
tape volumes from the HSMABR RACF tape volume set 304
temporary data sets 186
temporary data sets, SMS-managed storage 51
utility data sets, specifying age 80
DELVOL
JES3 considerations 495
optional parameters
PURGE 494
REASSIGN 494
UNASSIGN 494
pool configuration in JES3 495
PURGE parameter, redefining the backup cycle 415
description, ISMF panel parameter 299
device used for backup, specifying 173
devices
supported by DFSMShsm 12
used for backup 12
used for migration level 1 12
used for migration level 2 12
DFSMSdfp
family membership 3
DFSMSdss
automatic recall with 67
DASD I/O buffering technique to use for dump 154
invoking RESTORE function 427
restore data sets or volumes 427
specifying as data mover for CDS backup 449
DFSMShsm
address space, obtaining information about 526
allowing all functions to run 547
cancel queued request 552
coexistence considerations
ABACKUP output data sets 320
aggregate groups and management classes 302
data flow 5
devices supported 12
functions
for non-SMS-managed volumes 11
for SMS-managed volumes 7
overview 5
partitioning 539
RACF authority needed to perform 577
log information, printing 553
media, managing 483
member of DFSMSdfp family 3

DFSMShsm *(continued)*
objective in using 6
operations, obtaining statistical information 530
partitioning functions 539
preventing functions from running 543
previously prevented functions, allowing to run 547
processing, determining results without moving data sets 533
RACF concepts 576
service request from application programs 515
starter set 4
stopping support activities 445, 542
tape security when using with DFSMSrmm 538
using with DFSMSrmm 537
DFSMShsm-managed
automatic space management, SMS-managed storage 50
volumes
dumping 180
obtaining information 522, 523
DFSMShsm-owned
tape volumes
copying categories 404
data set name of backup tape 58
data set name of migration tape 57
data set name of TAPECOPY of backup tape 190
data set name of TAPECOPY of migration tape 190
making backup copies 401
making copies 402
protecting 514
DFSMSrmm
invocation by tape volume exit ARCTVEXT 406
retaining DFSMShsm tape volumes
by expiration date 537
by vital record processing 538
tape security when using with DFSMShsm 538
using with DFSMShsm 537
disability 641
disaster
alternate volumes 409
backup 401
backup copies of individual volumes, making 406
backup facilities 402
recovery 350
recovery, planning 401
disaster backup
making copies of individual volumes 406
discrete
profile data set 574
profiles, RACF, recovering 430
DISPLAY
EXPIREBV command 216
displaying SMS class definitions during ABACKUP and ARECOVER 370
DISPOSITION subparameter, DUMPCLASS parameter, DEFINE command 158
documents, licensed xviii
down-level MVS image, restoring stacked output tapes with 606
DRAIN parameter, ADDVOL command 493
DSCOMMAND parameter 419
dump
additional considerations 424
automatic full-volume, example system 151
class
day to dump data to, specifying 157
how often to dump to, specifying 158
obtaining information 520
specifying characteristics 156
specifying, SMS-managed 152
command 207
compress parameter 426
control data sets 424
controlling automatic reuse of tapes 156
copy
data set restore from 235
list dump volumes for 427
restoring a complete volume from 254
stacking 155, 160
cycle 155
data on volumes becomes invalid, specifying when 159
data-set-changed indicator, specifying whether to reset 158
day to dump data to a dump class, specifying 157
days on which to dump, specifying 155
DFSMShsm-managed volumes 180
DUMP parameter
ADDVOL command 162
BACKVOL command 207
HOLD command 544
RELEASE command 548
excess VTOC copy data sets, deleting 182
excess VTOC copy data sets, uncataloging 182
expired copies, deleting 179
function 148
overview, non-SMS-managed 11
overview, SMS-managed 8
how often to dump to dump classes, specifying 158
kind of tape units to use, specifying 160
maximum number of tasks 155
name
data sets 181
dump copy data sets 181
VTOC copy data set 181
number of VTOC copies to keep, specifying 160
dump *(continued)*
of migrated data sets, specifying first qualifier 37
owned DFSMShsm volumes 424
preventing 425
processing
determining if volumes can be stacked 178
determining tape dump volumes available for output 526
example 178
restoring individual data sets from dump tapes, controlling 157
selecting tape volumes 182
specifying volumes, non-SMS-managed 263
stacking multiple copies on tape 155, 160
start window, specifying 153
tapes, returning to use 511
tasks 152
varying invocation 425
volume
defining to DFSMShsm 162
in one or more storage groups 208
list for a dump copy 427
non-SMS-managed 267
obtaining information 520
serialization 181
specifying concurrent copy with 154
VTOC copies, example 160
VTOC copy data set, write 181
what to do with volumes, specifying 158
DUMP CLASS attribute 152
DUMP SYSTEM/SYSPLEX NAME attribute 153
DUMPCLASS parameter, DEFINE command 156
DUMPCLASS parameter, LIST command 520
DUMPCLASS subparameter, DUMP parameter, ADDVOL command 162
DUMPCLASS subparameter, FROMDUMP parameter, RECOVER command 235
DUMPCYCLE parameter, DEFINE command 155
dumping
a volume by command 207
volumes in one or more storage groups 208
DUMPIO parameter, SETSYS command 154
DUMPVOLUME parameter, LIST command 520
DUMPVOLUME subparameter, FROMDUMP parameter, RECOVER command 235
DUPLEX parameter, SETSYS command 402
duplex tapes
and ARECOVER processing 403
creating off site 408
discussion of function 402

duplex tapes *(continued)*
specifying with SETSYS command 402
dynamic data set allocation
during aggregate backup processing 310
during aggregate recovery processing 372

E

edit-log data set, printing DFSMShsm log information 555
effects of profile modeling on profile recovery 431
eligibility for level 1 to level 2 migration, specifying for non-SMS-managed data sets 81
eligible data sets, back up 187
eligible dump copies, recovery processing 229
EMERGENCY parameter, SETSYS command, preventing processing of data sets 131
end BCDS scan, EXPIREBV processing 222
end of level 0 volume processing 62
ENDKEY parameter, EXPIREBV command 216
enhanced CDS recovery functions
implementing using JCL 464
requirements for using 463
EOD parameter, HOLD command 369
error
in control data sets, repairing 473
obtaining system condition information when one occurs 532
recovering from journal I/O 473
spill backup volume selection 203
spill backup volume selection, subsequent 203
errors within the common recall queue 607
examples
automatic backup 206
backing up cataloged and uncataloged data sets with the same name 266
backup processing 182
CDS backup and recovery 475
common volumes 15
DASD storage 14
dump processing 178
dump VTOC copies 160
JES3 pooling 89
management classes, backup 169
management classes, space management 29
migration level 1 volumes 15
moving data sets between level 0 volumes 485
non-SMS-managed 487
SMS-managed 486
processors 14
recovering the control and journal data sets
journal from DASD 478
MCDS from DASD 475
examples *(continued)*
recovering the control and journal data sets *(continued)*
MCDS when latest MCDS backup version is unusable 479
recovering the control data sets 475
storage groups 14
system 13
system, space management 21
tape devices 15
unmanaged volumes 15
volume pool 15
excess dump VTOC copy data sets, deleting 182
EXCLUDE
data set 285
keyword 295
EXECUTE parameter
ABACKUP command 311
ARECOVER command 351
EXPIREBV command 216
RECYCLE command 498
exit
ABARS backup error, ARCBEEXT 306
ABARS expiration date, ARCEDEXT 306
ABARS migration level 2 data set, ARCM2EXT 306
called during level 1 to level 2 migration 48
CDS backup, ARCCBEXT 453
data set deletion, ARCADEXT 96
data set migration, ARCMDEXT 100
second-level migration data set, ARCMMEXT 48, 74
space management and backup, ARCSAEXT 51
tape data set, ARCTDEXT 48, 74
tape volume, ARCTVEXT 408
EXITOFF parameter, SETSYS command, inactivate DFSMShsm exits 131
EXITON parameter, SETSYS command, activate DFSMShsm exits 131
EXITS parameter, SETSYS command, preventing processing of data sets 131
expiration attributes, specifying 25
EXPIRE AFTER DATE/DAYS attribute 25
EXPIRE AFTER DAYS NON-USAGE attribute 25
EXPIREBV
BCDS scan 218
ending message 222
optional parameters
CATALOGEDDATA 269
DELETEIFBACKEDUP 269
ENDKEY 216
EXECUTE 216
NONSMSVERSIONS 269
RESUME 216
STARTKEY 216
UNCATALOGEDDATA 269
parameter
HOLD command 545
RELEASE command 550
processing 217
EXPIREBV *(continued)*
required parameters
DISPLAY 216
set up output destination 218
starting message 218
expired
backup versions
non-SMS-managed data sets, deleting 269
non-SMS-managed data sets, processing 269
SMS-managed 219
SMS-managed, deleting from backup volumes 216
data sets
deleting, non-SMS-managed volumes 93
deleting, SMS-managed storage 51
specifying how to scratch 34
dump copies, deleting 179
EXPIREDDATASETS parameter, SETSYS command 34
expiring ABARS activity records 320
extent reduction
daily statistics report output 34
determining
candidate data sets, non-SMS-managed volumes 94
candidates for, SMS-managed storage 53
non-SMS-managed data sets 99
performing on non-SMS-managed volume 99
specifying how to do 34

F

fast replication 9, 238
catalogs, maintaining 247
considerations
copy pool 247
FRBACKUP 241
FRRECOV 243
copy pool 238
decreasing volumes 248
increasing volumes 247
recovery of 242, 249
setup 238
using 238
copy pool backup storage group
decreasing 249
setup 239
using 239
creating 240
holding 244
information, obtaining 245
parallelism
FRBACKUP 242
FRRECOV 243
preparing for 240
recovering 242
releasing 244
serialization 241
setup
copy pool backup storage group type 239

fast replication *(continued)*
setup *(continued)*
copy pool construct 238
target volumes, dumping 246
unneeded backup versions, deleting 244
using
copy pool backup storage group type 239
copy pool construct 238
extend storage group type 239
multiple copy pools 249
multivolume data sets 249
overflow storage group type 239
withdrawing relationship 244
fast subsequent migration
how migration cleanup affects 78
how recall affects 77
performing 54
recalling data sets 69
reconnecting data sets 76
specifying 76
using ARCMDEXT 78
filtering, generation data sets 288
FILTEROUTPUTDATASET parameter, ABACKUP command 312
first qualifier for names of backup versions, specifying 173
first qualifier for names of migrated data sets, specifying 37
FIXCDS
backup volume cleanup process 204
ensuring that CDSs are accurate 475
FlashCopy relationship, withdrawing 244
FORCENONSMS parameter
RECALL command 75, 108
RECOVER command
from a dump copy 236
with BDAM data set 232
with extended sequential data set 233
format
of names of list data sets 80
of names of utility data sets 80
forward recovery 251
FRBACKUP 240, 241, 244
FREEVOL
moving all data sets from level 1 490
moving backup volumes 493
using on migration level 1 volumes 488
FREQUENCY
parameter
ALTERDS command 267
BACKVOL command 269
DEFINE command 158
SETSYS command 261
frequency of backup, non-SMS-managed, specifying 261
frequency of backup, SMS-managed, specifying 165
FROMDUMP
parameter
RECOVER command 235
FROMVOLUME
parameter
BDELETE command 270
RECOVER command 271
recovering a data set 224
recovering a VSAM data set 273
FRRECOV 242, 243, 244
FSR (function statistics record)
and ABARS 287
recording ABARS CPU time and accounting codes 376
FTP (NetView file transfer program) 391
full journal, damaged CDS 585
full journal, undamaged CDS 589
full tapes, converting partial tapes to 407
full-volume dump
automatic, example system 151
invocation compress parameter 426
procedures 441
full-volume restore
combined with volume recovery 251
using APPLYINCREMENTAL subparameter 251
FULLDSNAME parameter
RECYCLE command 499
function
availability management 4, 147
backup
non-SMS-managed 12
SMS-managed 8
changes for SMS, command 569
dump
non-SMS-managed 11
SMS-managed 8
enhanced CDS recovery
implementing using JCL 464
requirements for use 463
non-SMS-managed volumes, DFSMShsm 11
overview of DFSMShsm 5
space management 3, 19
non-SMS-managed 12
SMS-managed 10

G

GDG
management attributes, specifying 27
GDS management class, space management 29
generation data sets
and the ARECOVER TGTGDS command 345
and the SETSYS ARECOVERTGTGDS command 345
deferred 131, 428
filtering 288
processing 345
GENERATION parameter, RECOVER command 225
generic data set profile 573
grouping data sets
with management classes 8
with storage groups 8
GUARANTEED BACKUP FREQUENCY attribute 165
GUARANTEED SPACE attribute and VSAM data sets, availability management 420
GUARANTEED SPACE attribute, space management 135
GUARANTEED SPACE VSAM data sets
recalling 136
changes in storage-class-to-data set mapping 137
changes in the storage group 136
problems 136
problems with conversion of data sets 138
recovering
changes in storage-class-to-data set mapping 422
changes in the storage group 421
problems with conversion of data sets 422

H

HALTERDS command, security checking 268
HARDCOPY parameter, RELEASE command 550
HBDELETE processing, authorization checking 223
HFS data set
eligibility to migrate to an SDSP 59
support during RECOVER 236
HOLD
optional parameters
ABACKUP 543
ALL 543
ARECOVER 544
AUDIT 544
BACKUP 544
COMMONQUEUE 111, 544
DUMP 544
EOD 369
EXPIREBV 545
FRBACKUP 244
FRRECOV 244
LIST 545
LOG 546
MIGRATION 546
RECALL 546
RECOVER 547
RECYCLE 547
REPORT 547
TAPE 546
TAPEDATASET 547
TSO 546
preventing
backup processing 419
dump processing 425
recovery processing 428
restore processing 428
hold-volumes date description 104
holding ABARS functions 369
HOST
parameter
LIST command 522
startup procedure 542
host modes 539

how to obtain
information about
backup volumes 518
conditions in the system when an error occurs 532
data sets 518
DFSMShsm address space 526
DFSMShsm operations 530
DFSMShsm-managed volumes 523
dump classes 520
dump volumes 520
migration and DFSMShsm-managed volumes 522
processor serialization 522
tape volumes 524
user authorization 526
information from
DFSMShsm 517
the control data sets 517
HSMABR
aggregate backup RACF 303
RACF tape volume set 304
HSMEDIT procedure, printing DFSMShsm log information 555
HSMLOG procedure, printing DFSMShsm log information 555

I

ICF catalog, ABARS processing 323, 371
ICFRU Forward Recovery Utility, procedure for using 253
ICHRIN03, RACF started-procedures table, printing DFSMShsm log information 555
IDCAMS
data set 330
EXPORT, use of, for CDS backup 448
identify expired backup versions, EXPIREBV processing, SMS-managed 219
identifying migration level 2 volumes to DFSMShsm 39
implementing enhanced CDS recovery using JCL 464
import utility ARCIMPRT 463
INCLUDE
data set 285
keyword 294
list, processing migrated data sets in 347
incremental backup, automatic, example system 151
INCREMENTAL parameter, BACKVOL command 216
incremental volume recovery 256
INCREMENTALBACKUP parameter, SETSYS command 175
INDATASET parameter, TAPECOPY command 407
initial selection
DASD daily backup volume 194
DASD or tape daily backup volume 198
spill backup volume 202
initial selection *(continued)*
tape daily backup volume 196
inline backup, how to use 415
installation exits
aggregate backup 306
aggregate recovery 344, 358
instruction data set
creating 297
example 297, 337
overview 286
INSTRUCTION parameter, ARECOVER command 352
instruction/activity log file
aggregate backup output 317
description 287
input for recovery 329
integrated catalog facility catalog
backing up, procedure 440
recovering, procedure 440
recovery 273
restoring from a dump copy, procedure 441
interrupting
ABACKUP command processing 543
all DFSMShsm processing 543
and restarting
automatic dump processing 442
backup processing 437
primary space management 141
secondary space management 141
ARECOVER command processing 544
AUDIT command processing 544
backup processing 544
DFSMShsm functions 543
dump processing 544
EXPIREBV command processing 545
LIST command processing 545
long-running commands 570
migration processing 546
recycle processing 547
REPORT command processing 547
interval migration
example 64
non-SMS-managed volumes 101
SMS-managed volumes 64
specifying 32
INTERVALMIGRATION parameter 32, 33
introduction 1, 3
ISMF (Interactive Storage Management Facility)
aggregate group name, panel parameter 299
copies, panel parameter 299
description, panel parameter 299
example of aggregate group panel 300
output data set prefix, panel parameter 299
using
for aggregate group backup 376
for aggregate group recovery 382
for aggregate groups 298
for selection data sets 291

J

JES2
pools in systems running under 87
JES3
considerations, DELVOL command 495
deletion of migrated data sets fails, non-SMS-managed 105
DFSMShsm commands that affect pool configuration 87
DFSMShsm migration window 98
directed recall request 99
example of pooling 89
migration considerations, non-SMS-managed volumes 98
parameter, SETSYS command 88
pools in systems running under 87
prevent migration request 99
recall considerations, non-SMS-managed data sets 82, 104
specifying pools 87
volume pooling 89
journal copy utility ARCBJRNL 463
journal damaged, CDS undamaged 588
journal data set
DASD backup data sets 453
disabled 478
moving 474
multiple
backup data sets 450
tape backup data sets 455
nulling 457
recovering, examples 475
replacing, using multiple DASD backup data sets 478
journal full, undamaged CDS 589
journal I/O error, recovering from 473
JOURNAL parameter, UPDATEC command 471
journal records missing, damaged CDS 589
journal records missing, time sequence gaps in CDS records 589
JRNLBACKUPDSN subparameter, CDSVERSIONBACKUP parameter, SETSYS command 451

K

key range volumes, ML2, making available 506
keyboard 641

L

L0VOLS parameter, DEFINE command 339
last backup date, resetting 188
last-backup date, resetting, recovery processing 234
last-processed key 218
LBACKUP dates in catalog 399
level 0 volumes
moving all data sets to other level 0 volumes 483

LEVEL 1 DAYS NON-USAGE
attribute 27
level-1-to-level-2 migration
command 488
example 48
exits called 48
process 45
specifying when non-SMS-managed data sets become eligible 81
to a DASD level 2 volume 47
to a tape level 2 volume 46
licensed documents xviii
limit
daily
not reached, DASD or tape daily backup volume selection 199
reached, DASD or tape daily backup volume selection 198
on size of DFSMShsm DASD copies 6
LIMIT parameter, RECYCLE command 498
LIST
optional parameters
OUTDATASET 517
SELECT 518
SYSOUT 517
TERMINAL 517
required parameters
BACKUPVOLUME 518
DATASETNAME 518
DUMPCLASS 520
DUMPVOLUME 520
HOST 522
list and utility data sets, deleting, non-SMS-managed volumes 93
list data sets
format of names 80
specifying age for deleting 80
LIST parameter
HOLD command 545
RELEASE command 550
LOG parameter
HOLD command 546
RELEASE command 550
log, DFSMShsm, printing 553
LOGY data set, printing DFSMShsm log information 555
long-running commands 570
LookAt message retrieval tool xvii

M

macro invocation of DFSMShsm functions 515
main host 539
maintaining
control data sets 447
password protection for recovered data sets 432
maintenance and use of the data set date-last-referenced field for space management 61
management classes
aggregate backup overview 284
management classes *(continued)*
attributes
GDG ELEMENTS ON PRIMARY 27
ADMIN OR USER COMMAND BACKUP 168
AUTO BACKUP 169
BACKUP COPY TECHNIQUE 449
BACKUP FREQUENCY 165
COMMAND OR AUTO MIGRATE 27
EXPIRE AFTER DATE/DAYS 25
EXPIRE AFTER DAYS NON-USAGE 25
LEVEL 1 DAYS NON-USAGE 27
NUMBER OF BACKUP VERSIONS 166
PARTIAL RELEASE 26
PRIMARY DAYS NON-USAGE 27
RETAIN DAYS EXTRA BACKUP VERSIONS 167
RETAIN DAYS ONLY BACKUP VERSION 167
ROLLED-OFF GDS ACTION 28
considerations 301
default 28
example
backup 169
space management 29
grouping data sets with 8
large, space management 30
retrieval of
availability management 433
space management 138
support for CDS backup 449
use in backup 164
use of 8
managing
DFSMShsm media 483
SDSP data sets 511
storage with DFSMShsm, tasks for 5
manual restarts
of automatic primary space management 142, 143
of automatic secondary space management 143
MAXABARSADDRESSSPACE parameter, SETSYS command 307, 341
MAXBACKUPTASKS parameter, SETSYS command 174
MAXDUMPTASKS parameter, SETSYS command 155
MAXEXTENTS parameter, SETSYS command 34
maximum number of
backup tasks in each processor, specifying 174
backup versions to keep, specifying 166
dump tasks 155
MAXINTERVALTASKS parameter, SETSYS command 36
MAXMIGRATIONTASKS parameter, SETSYS command 35
MAXRECALLTASKS parameter, SETSYS command 36
MAXRECYCLETASKS parameter, SETSYS command 497
MAXSSMTASKS parameter, SETSYS command 35
MCDSBACKUPDSN subparameter, CDSVERSIONBACKUP parameter, SETSYS command 451
media management 483
MENTITY parameter, ARECOVER command 344, 353
message retrieval tool, LookAt xvii
messages
ARC0026E 585, 588, 589
ARC0734I 593
ARC0909E 589
ARC1102I 607
ARC1187E 607
ARC1502I 608
ARC1506E 607
messages, selecting for ABARS activity logs 308
migrate
data sets, SMS-managed storage 56
to an SDSP, size eligibility for data sets 59
MIGRATE
command for SMS-managed data sets 73
deleting non-SMS-managed data sets from migration volumes 490
parameters
CONVERT, non-SMS-managed data sets 484
CONVERT, SMS-managed data sets 484
DAYS 489
DELETEBYAGE 490
DELETEIFBACKEDUP 490
MIGRATIONLEVEL1 489
VOLUME 490
VOLUME, SMS-managed data sets 74
using for migration level 1 volumes 489
migrate candidates for
both migration and extent reduction, SMS-managed storage 55
extent reduction, SMS-managed storage 56
MIGRATE subparameter, ADDVOL command 84
MIGRATE SYSTEM/SYSPLEX NAME attribute 25
migrated data sets
backing up 185
deleting 71
GDG FORCE, caution with 72
names 57
profile backup 190
recovering 506
results with different commands 71
uncataloging 582
migrated data sets that got deleted 604
MIGRATEDDATA parameter, ARECOVER command 353

MIGRATEPREFIX parameter 37
migrating
all data sets from migration level 1 volume, processing for non-SMS-managed data sets 491
all data sets from migration level 1 volume, processing for SMS-managed data sets 491
data sets 489
data sets to migration level 2, specifying how 32
migration
and recall from DASD, RACF considerations 580
attributes, specifying 27
automatic interval space management tasks, specifying 36
automatic primary space management tasks, specifying 35
cleanup
example 45
process 44
specifying how to do 38
command, of a data set 73
control data set
recovering, using multiple DASD backup data sets 475
recovering, using multiple tape backup data sets 479
recovery and multiple backup data sets, using UPDATEC 469
controlling, for data sets and volumes, non-SMS-managed 107
determining candidate data sets, non-SMS-managed volumes 93
from SDSP data sets 46
interval, non-SMS-managed volumes 101
JES3 considerations, non-SMS-managed volumes 98
level 1 volumes
damaged, recovering backup versions from 510
example 15
moving all data sets 490
using the MIGRATE command 489
level 2
key range volumes, making available 506
specifying desired number of partial tapes 123
minimum age 93
name of
cartridge-type single-file data sets 57
dump copy data sets 181
non-SMS-managed volumes 96
out-of-space condition 60
preventing processing of data sets 131
processing, determining which tape ML2 volumes are available for output 530
RACF-protected data sets 133
reconnecting data sets 77
SMS-managed volumes 74
migration *(continued)*
tape volume contention 117
to a DASD ML2 volume, level 1 to level 2 migration 47
to a tape ML2 volume, level 1 to level 2 migration 46
to both DASD and tape for migration level 2 56
using the FREEVOL command 490
volume contention
single-processing-unit environment 117
volumes
deleting non-SMS-managed data sets with the MIGRATE command 490
obtaining information about 522
recovering 507
with thresholds of occupancy, non-SMS-managed volumes 96
without thresholds of occupancy, non-SMS-managed volumes 96
migration level one volume, damaged 598
MIGRATION parameter
HOLD command 546
HOLD command, preventing processing of data sets 131
MIGRATIONLEVEL2 subparameter 39
MIGRATIONVOLUME parameter, FREEVOL command 490
ML1BACKUPVERSIONS parameter, FREEVOL command 488
RELEASE command 551
migration tape, overwritten 590
MIGRATIONCLEANUPDAYS parameter, SETSYS command 38
MIGRATIONCONTROLDATASET parameter, UPDATEC command 471
MIGRATIONLEVEL1 parameter, MIGRATE command 489
MIGRATIONLEVEL1DAYS parameter, SETSYS command 81
MIGRATIONLEVEL2 parameter, TAPECOPY command 404
minimum migration age 93
missing journal records, damaged CDS, 589
ML1 subparameter, MIGRATEDDATA parameter, ARECOVER command 353
ML1 volume, damaged 598
ML1VOLS parameter, DEFINE command 339
ML2 parameter, RECYCLE command 497
ML2 subparameter, MIGRATEDDATA parameter, ARECOVER command 353
ML2PARTIALSNOTASSOCIATEDGOAL parameter, SETSYS command 123
ML2RECYCLEPERCENT parameter, SETSYS command 496
ML2TAPE subparameter, TAPEMIGRATION parameter, SETSYS command 33
model entity (MENTITY) parameter, ARECOVER command 353
model entity, defining 344
modeling, profile, effects on profile recovery 431
MOVE parameter, ABACKUP command 312
multicluster CDS considerations 456
multiple backup data sets
backing up the control data sets 450
changing
from DASD to tape 459
from tape to DASD 458
choosing
DASD 451
tape 451
DASD, example
recovering the MCDS 475
replacing the journal 478
recovering control data sets and journal data set 475
tape, example of recovering the MCDS 479
multiple-file tapes, making into cartridge-type single-file-format tapes 501
multitasking considerations for SDSP data sets 128

N

names of
backup copy data sets 190
backup versions of data sets 189
backup VTOC copy data sets 187
dump copy data sets 181
dump VTOC copy data sets 181
list data sets, format 80
migrated data sets 57
migrated data sets, specifying first qualifier 37
utility data sets, format 80
NEWNAME parameter, RECOVER command 224, 231, 235
no-migration management class, space management 30
NOBACKUPMIGRATED parameter, ARECOVER command 354
NOFIX parameter, AUDIT command 472
NOMIGRATION parameter, SETMIG command, preventing processing of data sets 131
non-SMS-managed
data sets
deleting expired backup versions 269
extent reduction 99
processing, migrating all data sets from ML1 volume 491
recall 101
storage
automatic space management processing 90
availability management 261
command availability management 267
command space management 106
space management 79

non-SMS-managed *(continued)*
storage *(continued)*
subtasks for availability management 261
subtasks for space management 79
volumes
automatic space management 265
backing up all primary 268
command volume space management 107
data set deletion 95
data set retirement 95
deleting expired data sets 93
deleting list and utility data sets 93
DFSMShsm functions for 11
interval migration 101
migration 96
migration with thresholds of occupancy 96
migration without thresholds of occupancy 96
specifying recall characteristics 81
non-VSAM data set
recovering from a DFSMShsm backup version 230
restoring from a dump copy 231
NONE subparameter, TAPEMIGRATION parameter, SETSYS command 33
NONE(ROUTETOTAPE) subparameter, TAPEMIGRATION parameter, SETSYS command 33
NONSMSVERSIONS parameter, EXPIREBV command 269
NOSPILL parameter, SETSYS command, cleanup processing 204
NOSTACK parameter
ABACKUP command 313
ARECOVER DATASETNAME command 352
SETSYS ABARSTAPES command 307, 342
nulling the journal 457
NUMBER OF BACKUP VERSIONS
attributes 166
relationship to RETAIN DAYS EXTRA BACKUP VERSIONS 168
relationship to RETAIN DAYS ONLY BACKUP VERSIONS 168

O

objective in using DFSMShsm 6
OCDS TTOC record handling 505
offline control data set
recovery and multiple backup data sets, using UPDATEC 469
OFFLINECONTROLDATASET parameter, UPDATEC command 471
operating procedures for DFSMShsm 539
OPTIMIZE parameter, ABACKUP command 308, 311, 312
optimum
DASD blocking option 341
order of data set migration
automatic space management, non-SMS-managed 94
automatic space management, SMS-managed 56
interval migration 64
orderly shutdown 543
organization
of part 1 1
of part 2 20
of part 3 150
of part 4 445
of this manual 4
ORIGINAL subparameter, INCREMENTALBACKUP parameter, SETSYS command 175
ORIGINALVOLUMES parameter
TAPECOPY command 406
TAPEREPL command 407
OUTDATASET parameter
LIST command 517
RECYCLE command 499
output data set prefix, ISMF panel parameter 299
overview
backup and recovery functions
non-SMS-managed 12
SMS-managed 8
dump and restore functions
non-SMS-managed 11
SMS-managed 8
of DFSMShsm functions 5
space management functions
non-SMS-managed 12
SMS-managed 10
overwritten
backup tape 594
migration tape 590
owned dumping volume 424

P

parameters
of the recycle process, defining 496
used in creating backup profiles 191
part 1, organization 1
part 2, organization 20
part 3, organization 150
part 4, organization 445
partial
release, specifying 26
tapes
converting to full tapes 407
specifying desired number of 123
PARTIAL RELEASE attribute 26
partitioned data set
deleting 73
partitioning DFSMShsm activities 539
PDA (problem determination aid)
specifying use of 532
using 531
PDA parameter, SETSYS command 532
PERCENTUTILIZED parameter, ARECOVER command 354
PERCENTVALID parameter, RECYCLE command 497
phase 1
backing up the control data sets 183
DFSMShsm-managed volume space management, SMS-managed storage 50
processing example, SMS-managed storage 54
phase 2
moving backup versions 184
processing example, SMS-managed storage 62
SMS-managed storage 55
phase 3
backing up migrated data sets 185
phase 4
backing up DFSMShsm-managed volumes from storage groups with the automatic backup attribute 186
physical data set restore
HRECOVER considerations 227
pool
data set
definition 15
of volumes
configuration in JES3 87
defining 85
general, configuration 87
general, definition 87
in systems running under JES2 87
requirements and restrictions 86
pool configuration in JES3 88
POOL parameter, DEFINE command 85
pooling
examples of JES3 89
PREFIX parameter
RECYCLE command 499
PREPARE parameter, ARECOVER command 352, 357
preparing for aggregate recovery 331
prevent-migration date, description 98
prevented functions, allowing to run 547
prevention of
ABACKUP command processing 543
all DFSMShsm processing 543
ARECOVER command processing 544
AUDIT command processing 544
backup processing 419, 544
DFSMShsm functions from running 543
dump processing 425, 544
entries in the DFSMShsm log 546
EXPIREBV command processing 545
fast replication processing 247
further migration to a DASD migration volume 493
LIST command processing 545
migration processing 546
processing of data sets during volume space management 131
recall processing 546
recovery processing 428, 547
recycle processing 547
REPORT command processing 547
restore processing 428

PRIMARY DAYS NON-USAGE
attribute 27
PRIMARY parameter
ADDVOL command 262
ADDVOL, pool configuration in JES3 88
BACKVOL command 268
primary space management
automatic restarts of 142
defining a processing window 144
manual restarts of 143
overview 19
primary volume
attributes 84
defining to DFSMShsm 82
space management 92
printing DFSMShsm log information 553
priority
of data set recoveries 234
of recalls and deletions 116
of volume recoveries 258
order
for migration, SMS-managed storage 56
sort data sets, SMS-managed storage 56
PRIVATEVOLUME subparameter, RECALL parameter, SETSYS command 82
pool configuration 87
pool configuration in JES3 88
problems with recalling GUARANTEED SPACE VSAM data sets, space management 136
problems with recovering GUARANTEED SPACE VSAM data sets, availability management 421
procedures
availability management 437
backing up and recovering ICF catalogs 440
backup and recovery 437
full-volume dump 441
interrupting and restarting
automatic backup processing 437
automatic dump processing 442
sending alternate tapes off site 408
space management 141
stopping DFSMShsm 542
using ICFRU Forward Recovery Utility 253
process data set records from BCDS, EXPIREBV processing 219
process window, defining a new one
for primary space management 141
for secondary space management 141
processing
ABARS ICF user catalog and catalog alias 323, 371
cleanup 204
example, backup 182
for expired backup versions of non-SMS-managed data sets 269
processing *(continued)*
for non-SMS-managed data sets, migrating all data sets from migration level 1 volume 491
for SMS-managed data sets, migrating all data sets from migration level 1 volume 491
generation data sets 345
migrated data sets
in the ALLOCATE list 348
in the INCLUDE list 347
order, ARECOVER conflict resolution 367
prevent data set 131
priority of data set recovery 234
priority of recalls and deletions 116
priority of volume recovery 258
time, tracking for ABARS 319, 376
unit serialization information, obtaining 522
units that process each storage group, specifying 25
PROFILE parameter, SETSYS command 174
profile, RACF
authorization checking during recovery 429
backup
deleting 191
for migrated data sets 190
parameters used in creating 191
discrete
profile data set 574
recovering 430
specifying whether to back up 174
generic 573
modeling, effects on profile recovery 431
recovered, characteristics 431
recovering 429
recovery
accessing a data set when profile cannot be recovered 432
authorization checking during 431
effects of profile modeling 431
protect-all environment, RACF 575
protection
password
data sets on DASD 134
recovered data sets, maintaining 432
RACF
data sets 573
discrete data set profiles 574
generic data set profiles 573
PURGE parameter, DELETE command 73
PURGE parameter, DELVOL command 494
purpose of
availability management 4
DFSMShsm commands 567
reblocking 69
space management 3

Q

QUERY
ABARS 309
ARPOOL 340
COMMONQUEUE 527, 563
find request numbers, user IDs, and data set names 552
QUERY command 526
querying held ABARS functions 370

R

RACF (Resource Access Control Facility)
access with RACF FACILITY class 582
ARCCATGP logon group 68
authority needed to perform DFSMShsm functions 577
authority required for space management functions 132
authorization checking
during HALTERDS processing 429
during HBDELETE processing 429
during recovery 429
checking during aggregate backup 304
checking during aggregate recovery 333
considerations 573
discrete data set profiles
for cataloged data sets, backing up 190
recovering 430
specifying whether to back up 174
discrete profiles
data set 574
data set, aggregate recovery 353
for aggregate recovery 344
facility authorization checking, ARECOVER data set name conflict resolution 367
generic data set profiles 573
indication 574
indicator
definition 574
with always-call 575
model entity 344
protect-all environment 575
protecting data sets 573
RACF catalog maintenance group 68
security bypassing for DFSMShsm commands 578
started task table 579
started-procedures table, printing DFSMShsm log information 555
user ID for DFSMShsm 578
RACF and DFSMShsm, concepts 576
RACF-indicated data sets
backing up 581
definition 574
deleting 133
protecting aggregate backup volumes 303

RACF-indicated data sets *(continued)*
recalling 134
recovering or restoring 430
RACF-protected data sets, migrating 133
REASSIGN parameter, DELVOL command 494
reblocking
during recall 69
during recovery 234
purpose 69
recall
additional considerations 68
allocating data sets to volumes, non-SMS-managed 103
attributes 85
automatic
command data set scratching, non-SMS-managed 104
command deletion of VSAM AIX clusters 67
data set renaming 67
during catalog updates 68
during data set allocation 67
during non-VSAM data set opening, non-SMS-managed 103
during VSAM data set opening, non-SMS-managed 103
example system 22
SMS-managed 65
with DFSMSdss 67
command
direct access data sets 75
non-SMS-managed data set to specific volume 108
SMS-managed BDAM data set to non-SMS-managed volume 108
SMS-managed data set to non-SMS-managed volume 108
SMS-managed data sets 75
SMS-managed extended format data set to non-SMS-managed volume 108
data set
migrated for extent reduction 67
SMS-managed BDAM data set to non-SMS-managed volume 108
SMS-managed data set to non-SMS-managed volume 108
SMS-managed extended format data set to non-SMS-managed volume 108
to a specific volume, non-SMS-managed 108
during catalog updates 68
JES3 considerations for non-SMS-managed data sets 82, 104
non-SMS-managed data sets 101
priority 116
RACF-indicated data sets 134
reblocking during 69
SMS-managed BDAM data set to non-SMS-managed volume 108
SMS-managed data set to non-SMS-managed volume 108
recall *(continued)*
SMS-managed extended format data set to non-SMS-managed volume 108
specifying characteristics for non-SMS-managed volumes 81
tape take away by 117
tape volume contention 117
tasks, specifying 36
volume contention
multiple-processing-unit environment 121
volume selection, non-SMS-managed 101
RECALL parameter
DAOPTION parameter 75, 108
FORCENONSMS parameter 75, 108
HOLD command 546
JES3 restrictions and recommendations 105
RELEASE command 551
SETSYS command 82
recalling
extent-reduced data sets 67
problems that can occur
changes in the storage group 136
changes in the storage-class-to-data set mapping 137
conversions of data sets 138
GUARANTEED SPACE VSAM data sets 136
reclaiming unused DASD space
deleting expired backup versions, non-SMS-managed 269
deleting expired backup versions, SMS-managed 216
deleting expired data sets, space management 34
specifying
extent reduction 34
migration level 1 space 38
partial release 26
reconnection 33
record level sharing, accessing CDSs with 459
record, ABARS activity 338
records
for migration cleanup, specifying age 38
RECOVER
optional parameters
APPLYINCREMENTAL 251, 276
DAOPTION 232
DATE 225
DUMPCLASS 235
DUMPVOLUME 235
FROMDUMP 235
FROMVOLUME 271
GENERATION 225
NEWNAME 224, 235
REPLACE 224, 235
recovering a volume 249
required parameters
TOVOLUME 235, 271, 274
UNIT 235, 271, 274
recovered
control data sets, updating 471
recovered *(continued)*
data sets, maintaining password protection 432
profile, characteristics 431
recovering
a migrated data set 506
a migrated volume 507
a non-VSAM data set from a DFSMShsm backup version 230
a VSAM data set from a DFSMShsm backup version 230
ABARS instruction data set 359
backup versions from a damaged ML1 volume 510
damaged single-file ML2 volumes 506
data sets to SMS-managed volumes 229
from journal I/O error 473
ICF catalog, procedure 440
or restoring RACF-indicated data sets 430
problems that can occur
changes in the storage group 421
changes in the storage-class-to-data set mapping 422
conversions of data sets 422
GUARANTEED SPACE data sets 421
RACF discrete profiles 430
specific data sets, non-SMS-managed 271
RECOVERNEWNAMEALL parameter, ARECOVER command 354
RECOVERNEWNAMELEVEL parameter, ARECOVER command 354
recovery
aggregate 350
command 224
control data sets 462
copy pool 249
data set 224
date-of-scratch field 234
direct access data sets 232
examples 475
examples, control data sets 475
extended format data sets 232
fast replication 242
from DASD, RACF considerations 580
incremental 256
journal from DASD, example 478
last-backup date 234
MCDS from DASD, example 475
MCDS from tape, example 479
prevent processing 428
procedures 437
CDS function 463
control data set 462
DASD backup data set support 453
multiple backup data sets for CDS and journal 450
multiple tape backup data set support 455

recovery *(continued)*
profile
accessing a data set when DFSMShsm cannot recover a discrete profile 432
authorization checking during 431
effects of profile modeling on 431
restore of data sets 149
SDSP 512
use of date last referenced field 428
volume 249
VSAM data set 273
RECOVERY parameter
HOLD command 547
RELEASE command 551
recycle
eligibility 499
process, defining parameters 496
process, initiating 497
selecting tape volumes during 500
volume contention
multiple-processing-unit environment 121
single-processing-unit environment 119
tape 117
RECYCLE
optional parameters
FULLDSNAME 499
LIMIT 498
OUTDATASET 499
PERCENTVALID 497
PREFIX 499
SELECT 498
TAPELIST 498
required parameters
ALL 497
BACKUP 498
DAILY 497
DISPLAY 497, 499
EXECUTE 498
ML2 497
SPILL 497
VERIFY 499
VOLUME 497
RECYCLE parameter
HOLD command 547
RELEASE command 552
RECYCLEOUTPUT parameter, SETSYS command 497
RECYCLEPERCENT parameter, SETSYS command 496
RECYCLETAPEALLOCATION parameter, SETSYS command 497
redefining the backup cycle 415
reel-type tapes, making into cartridge-type single-file-format tapes 501
relationship of
frequency and maximum number of backup versions 167
NUMBER OF BACKUP VERSIONS and RETAIN DAYS attributes 168
RELEASE
BACKUP or ALL, interrupting and restarting backup processing 438
RELEASE *(continued)*
DUMP or ALL, interrupting and restarting dump processing 443
held
backup processing 419
dump processing 425
fast replication processing 244
recovery processing 428
restore processing 428
optional parameters
ABACKUP 547
ALL 547
ARECOVER 547
AUDIT 548
BACKUP 548
COMMONQUEUE 549
DUMP 548
EXPIREBV 550
FRBACKUP 244
FRRECOV 244
HARDCOPY 550
LIST 550
LOG 550
MIGRATION 551
RECALL 551
RECOVER 551
RECYCLE 552
REPORT 552
TAPE 551
TAPEDATASET 551
TSO 551
releasing
automatic hold 551
backup volume 206
held ABARS functions 370
held backup processing 419
held dump processing 425
held fast replication processing 244
held recovery processing 428
held restore processing 428
unused space, SMS-managed storage 54
removal of
ML1 volume that contains an SDSP 513
residual dump and backup VTOC copy data sets from ML1 volumes 493
SDSPs from a ML1 volume 514
volumes from DFSMShsm management and control 494
RENAMESOURCE subparameter, DATASETCONFLICT parameter, ARECOVER command 353, 361
RENAMETARGET subparameter, DATASETCONFLICT parameter, ARECOVER command 353, 361
REPLACE parameter
ARECOVER command 353, 362
RECOVER command 224, 235
REPORT 530
REPORT parameter
AUDIT command 472
HOLD command 547
RELEASE command 552
requirements
for using enhanced CDS recovery function 463
RESET subparameter, DUMPCLASS parameter, DEFINE command 158
restart data set 330, 359
RESTART keyword 562
restarting
after interrupted automatic dump processing 442
after interrupted backup processing 437
automatic primary space management when the current time is not in the process window 143
automatic secondary space management when the current time is not in the process window 143
DFSMShsm after an abnormal end 562
primary and secondary space management
automatic 141
command 141
restore
and recovery of data sets 149
command 224
data sets or volumes with DFSMSdss 427
multiple-volume data set 236
prevent processing 428
VSAM data sets 237
restoring
a complete volume from a dump copy 254
a non-VSAM data set from a dump copy 231
a VSAM data set from a dump copy 232
individual data sets from dump tapes, controlling 157
restoring stacked output tapes with a down-level MVS image 606
restricted command authority
ABACKUP 305
ARECOVER 334
RESUME parameter, EXPIREBV command 216
RETAIN DAYS EXTRA BACKUP VERSIONS
attribute 167
relationship to NUMBER OF BACKUP VERSIONS 168
RETAIN DAYS ONLY BACKUP VERSIONS
attribute 167
relationship to NUMBER OF BACKUP VERSIONS 168
RETENTIONPERIOD parameter
BACKVOL command 207
DEFINE command 159
retirement (delete if backed up)
definition 95
handling of retired versions
creating 96
handling during EXPIREBV 269

retirement (delete if backed up) *(continued)*
handling of retired versions *(continued)*
handling during spill processing 201
recovery 276
restriction to non-SMS-managed volumes 79
retrieval of
list of management class definitions, EXPIREBV processing 218
management classes, availability management 433
management classes, space management 138
storage groups, availability management 433
storage groups, space management 138
return codes
ARCBJRNL 468
ARCIMPRT 468
returning dump tapes to use 511
reuse of dump tapes, controlling automatic 156
RLS mode, accessing CDSs with 459
ROLLED-OFF GDS ACTION attribute 28
running
automatic functions concurrently 563

S

SAF (system authorization facility) 573
SCRATCHFREQUENCY parameter, SETSYS command 81
SDSP (small data set packing)
backup and recovery 512
data sets
contention 125
format for name 512
managing 511
multitasking considerations 128
reorganizing 511
security 582
removing 513
selection 60
specifying 36
searching for eligible dump copies, recovery processing 229
SECLABEL CLASS 573
second-level migration data set exit ARCMMEXT 48, 74
secondary address space, maximum 341
specifying frequency of backup, SMS-managed 165
specifying GDG management attributes 27
specifying interval migration 32
specifying migration tasks 36
specifying space requirements 22
secondary host promotion
automatic CDS backup 447
backup processing 183
dump processing 179
during DFSMShsm stop 542
secondary host promotion *(continued)*
for SSM hosts 43, 145
secondary space management
automatic restarts of 142
defining a processing window 144
manual restarts of 143
overview 19
security
checking for HALTERDS command 268
checking with discrete profiles 268
considerations for automatic space management 132
considerations for availability management 429
small data set packing 582
tape
when using DFSMShsm with DFSMSrmm 538
SELECT parameter 498
LIST command 518
RECYCLE command 498
selection
backup volumes 201
DASD and tape daily backup volumes 197
initial DASD and tape daily backup volumes 198
initial spill backup volumes 202
messages to write to ABARS activity logs 308
SDSP for migration 60
subsequent DASD and tape daily backup volumes 199
subsequent spill backup volumes 203
tape volumes during recycling 500
volume
daily backup 191
DASD daily backup 193
initial DASD daily backup 194
initial tape daily backup 196
subsequent DASD daily backup 195
subsequent tape daily backup 197
tape daily backup 196
tape dump 182
target, incremental backup version recovery processing 225
target, physical dump copy restore 225
selection data set
ACCOMPANY data set 285
ACCOMPANYEXCLUDE data set 285
ALLOCATE data set 285
ALLOCATEEXCLUDE data set 285
creating 291
examples 296
EXCLUDE data set 285
INCLUDE data set 285
keyword descriptions 294
keyword syntax 291
overview 285
sending alternate tapes off site 408
sequential data striping
aggregate backup processing considerations 396
sequential data striping *(continued)*
aggregate recovery processing considerations 397
automatic space management processing considerations 139
backup processing considerations 434
recall processing considerations 139
volume space management processing considerations 139
serialization of
dump volume 181
processor information, obtaining 522
SETMIG
purpose of 568
using to control migration of non-SMS-managed data sets and volumes 107
SETSYS command
ANYSTORAGEVOLUME parameter, pool configuration 88
BACKUP, interrupting and restarting backup processing 438
BACKUP, interrupting and restarting dump processing 443
BACKUP, specifying handling of data sets INUSE 176
CDSVERSIONBACKUP, multiple backup data sets, recovering the CDSs using UPDATEC 469
defining aggregate backup options 306
examples for aggregate backup 308
JES3, defining a JES3 system 88
NOEMERGENCY, interrupting and restarting backup processing 438
NOEMERGENCY, interrupting and restarting dump processing 443
optional parameters
ABARSACTLOGMSGLVL 308
ABARSBUFFERS 307
ABARSDELETEACTIVITY 318
ABARSOPTIMIZE 308
ABARSPROCNAME 307, 341
ABARSTAPES 307, 342
ACTLOGTYPE 307
ANYSTORAGEVOLUME 82
ARECOVERPERCENTUTILIZED 342
ARECOVERTGTGDS 343
ARECOVERUNITNAME 342
AUTOBACKUPSTART 172
BACKUP 173
BACKUPCOPIES 450
BACKUPDEVICECATEGORY 451
BACKUPPREFIX 173
BCDSBACKUPDSN 451
CDSVERSIONBACKUP 448
CHANGEDONLY 175
COMMONQUEUE 111, 113, 114
DAYS 81
DEBUG 533
DUMPIO 154
DUPLEX 402
EXITOFF 307, 341
EXITON 307, 341
EXPIREDDATASETS 34
FREQUENCY 261

SETSYS command *(continued)*
optional parameters *(continued)*
INCREMENTALBACKUP 175
INTERVALMIGRATION 32
JRNLBACKUPDSN 451
MAXABARSADDRESSSPACE 307, 341
MAXBACKUPTASKS 174
MAXDUMPTASKS 155
MAXEXTENTS 34
MAXINTERVALTASKS 36
MAXMIGRATIONTASKS 35
MAXRECALLTASKS 36
MAXRECYCLETASKS 497
MAXSSMTASKS 35
MCDSBACKUPDSN 451
MIGRATEPREFIX 37
MIGRATIONCLEANUPDAYS 38
MIGRATIONLEVEL1DAYS 81
ML2PARTIALSNOTASSOCIATEDGOAL 123
ML2RECYCLEPERCENT 496
ML2TAPE 33
NOCDSVERSIONBACKUP 456
NONE 33
NONE(ROUTETOTAPE) 33
NOSPILL 204
OCDSBACKUPDSN 451
ORIGINAL 175
PDA 532
PRIMARYSPMGMTSTART 31
PRIVATEVOLUME 82
PROFILEBACKUP 174
RECALL 82
RECYCLEOUTPUT 497
RECYCLEPERCENT 496
RECYCLETAPEALLOCATION 497
SCRATCHFREQUENCY 81
SECONDARYSPMGMTSTART 31
SMALLDATASETPACKING 36
SPILL 175
TAPE 173
TAPEMAXRECALLTASKS 36
VERSIONS 262
PRIVATESTORAGEVOLUME, pool configuration in JES3 88
UNLOAD, releasing a backup volume 206
shortcut keys 641
shutdown, orderly 543
single-processing-unit-environment volume contention 117
size
eligibility for data sets to migrate to an SDSP 59
of data sets eligible for small data set packing 36
SMALLDATASETPACKING parameter 36
SMS (Storage Management Subsystem) purpose 3
SMS class definitions, displaying during ABACKUP and ARECOVER 370
SMS-managed
attributes 334
automatic recall 65
backup frequency 165
class names 334
SMS-managed *(continued)*
data sets 334
command recall of 75
processing, migrating all from ML1 volume 491
recalling BDAM to non-SMS-managed volume 108
recalling extended format to non-SMS-managed volume 108
recalling to non-SMS-managed volume 108
guaranteed frequency of backup 165
requirements
for aggregate backup 290
for aggregate recovery 334
storage
availability management of 151
space management of 21
volumes
command migration of 74
functions for 7
recovering data sets to 229
sort candidate data sets into priority order, non-SMS-managed data sets 94
SOURCELEVEL subparameter, MIGRATEDDATA parameter, ARECOVER command 354
space management
Access Method Services considerations 135
automatic processing for non-SMS-managed storage 90
catalog considerations 61
command
all primary volumes, non-SMS-managed 107
example system 22
functions supported 71
non-SMS-managed 106
daily
defining a new start window 439
data sets with discrete data set profiles 133
date-last-referenced field, maintenance and use 61
deleting
expired data sets, non-SMS-managed volumes 93
list and utility data sets, non-SMS-managed volumes 93
end of level 0 volume processing 62
example differences for non-SMS-managed storage 79
extent reduction, non-SMS-managed data sets 99
functions 19
overview, non-SMS-managed 12
overview, SMS-managed 10
how to do 19
integrity age 40
interrupting
and restarting primary 141
and restarting secondary 141
interval migration example 64
JES3 considerations, non-SMS-managed volumes 98
space management *(continued)*
maintenance and use of the date-last-referenced field 61
manual restarts 142
non-SMS-managed volumes
pass 1 92
pass 2 94
of non-SMS-managed storage 79
of SMS-managed storage 21
of VSAM data sets 58
other considerations 109
performing on non-SMS-managed volumes 95
preventing processing of data sets 131
primary
automatic restarts of 142
manual restarts of 143
processing window 144
volumes, non-SMS-managed 92
volumes, pass 1 92
volumes, pass 2 94
priority order, non-SMS-managed data sets 94
procedures 141
RACF authority required 132
restarting
primary 141
secondary 141
when the current time is not in the process window 143
secondary
automatic restarts of 142
manual restarts of 143
processing window 144
specifying the number of tasks 35
security considerations 132
specific tapes subsequent volume selection management processing 200
subtasks for non-SMS-managed storage 79
volume, command, non-SMS-managed 107
VSAM data sets, non-SMS-managed volumes 97
when primary needs to be restarted 141
when secondary needs to be restarted 141
where it restarts 144
writing to a cartridge=type single-file tape 62
space management and backup exit ARCSAEXT 51
space requirements, specifying 22
specific data sets, delete backup versions 222
specifying age for deleting list data sets 80
specifying age for deleting utility data sets 80
specifying automatic interval space management tasks 36
specifying automatic primary space management tasks 35

specifying automatic secondary space management tasks 35
specifying DASD for CDS backup 452
specifying device type for backup of the CDSs 451
specifying DFSMSdss as data mover for CDS backup 449
specifying DFSMShsm as data mover for CDS backup 450
specifying first qualifier for names of backup versions 173
specifying first qualifier for names of migrated data sets 37
specifying maximum number of backup tasks in each processor 174
specifying maximum number of dump copies to stack on a volume 160
specifying maximum number of dump tasks 155
specifying migration attributes 27
specifying migration tasks 35
specifying recall characteristics for non-SMS-managed volumes 81
specifying recall tasks 36
specifying tape for CDS backup 452
specifying volumes to be dumped
  and dump classes to use,
    SMS-managed 152
  non-SMS-managed 263
spill
  backup volume
    DASD, subsequent spill volume selection 203
    moving DASD backup volumes to spill volumes 493
    none available during initial spill volume selection 203
    tape, subsequent spill volume selection 203
  initial backup volume selection 202
  processing 200
  processing, specifying whether to do 175
  subsequent backup volume selection 203
SPILL parameter
  RECYCLE command 497
  SETSYS command 175
STACK parameter
  ABACKUP command 313
  ARECOVER DATASETNAME command 352
  DEFINE command 155, 160
  SETSYS ABARSTAPES command 307
stacked output tapes, restoring with a down-level MVS image 606
stacking multiple dump copies on a tape 155, 160
standard interval migration compared to TMM 64
standard management class, space management 29
start window
  automatic primary space management, defining a new 144
  automatic secondary space management, defining a new 144
start window *(continued)*
  when automatic primary space management is not running 145
  when automatic secondary space management is not running 145
starter set 4
starting
  automatic primary space management 49
  automatic secondary space management 43
  DFSMShsm 540
STARTKEY parameter, EXPIREBV command 216
startup
  procedure
    HOST parameter 542
    multiple backup data sets and CDS recovery using UPDATEC 469
  window, defining for automatic dump 443
statistical information about DFSMShsm operations, obtaining 530
status
  DFSMShsm volumes and data sets, determining 567
stop DFSMShsm 542
storage
  groups
    ALLOCATION/MIGRATION THRESHOLD attribute 22
    AUTO BACKUP attribute 163
    AUTO DUMP attribute 152
    AUTO MIGRATE attribute 22
    backing up all eligible changed data sets to one or more of 216
    backing up data sets to one or more of 215
    BACKUP SYSTEM/SYSPLEX NAME attribute 163
    control of automatic dump 152
    control of dump classes 152
    control of number of dump copies 152
    copy pool backup storage group type 239
    DUMP CLASS attribute 152
    DUMP SYSTEM/SYSPLEX NAME attribute 153
    dumping volumes to one or more of 208
    example, space management 25
    grouping data sets with 8
    GUARANTEED BACKUP FREQUENCY attribute 165
    MIGRATE SYSTEM/SYSPLEX NAME attribute 25
    retrieval of, availability management 433
    retrieval of, space management 138
    use of 8
storage class, use of 8
storage to be managed, specifying 22
subsequent selection
  DASD daily backup volume 195
subsequent selection *(continued)*
  DASD or tape daily backup volume 199
  DASD spill backup volume 203
  spill backup volume 203
  tape daily backup volume 197
  tape spill backup volume 203
subtasks
  availability management of non-SMS-managed storage 261
  availability management of SMS-managed storage 152
  space management of non-SMS-managed storage 79
  space management of SMS-managed storage 22
  SYSOUT parameter
    AUDIT command 472
    LIST command 517
support activities 445
SWAPLOG 553
swapping log data sets 553
switching backup tapes 212
sysplex name support 11
system
  non-SMS-managed 13
  running under
    JES2, pools 87
    JES3, pools 87
  SMS-managed 13
system conditions when an error occurs, obtaining information about 532

T

tape
  and DASD daily backup volume, selecting 197
  backing up CDSs to multiple backup data sets 455
  backup overwritten 594
  changing from DASD to tape 459
  changing to DASD backup data sets 458
  choosing, for multiple backup data sets 451
  daily backup volume selection 196
    initial 196
    subsequent 197
  data set exit ARCTDEXT 48, 74
  data set name
    backup tape 190
    migration tape 57
    TAPECOPY of backup tape 190
    TAPECOPY of migration tape 58
  determining unit type for aggregate backup 309
  devices, example 15
  how to access single-file format data set 57
  migration level 2
    partial, specifying number of 123
    volume contention 117
  migration overwritten 590
  name of backup copy data sets 190
  or DASD, subsequent daily backup volume selection 199

tape *(continued)*
security
when using DFSMShsm with DFSMSrmm 538
selection
with duplex tape processing 403
with TAPECOPY processing 405
specifying for CDS backup 452
subsequent selection, backup 200
take away 213
using HMIGTAPE.DATASET to access single-file format data sets 57
volume
alternate, replacing an original with 407
backup, determining which are available for output 530
consolidating valid data 495
contention 117
determining which are available for output 530
DFSMShsm-owned, copying categories 404
DFSMShsm-owned, making backup copies 401
DFSMShsm-owned, making copies 402
DFSMShsm-owned, protecting 514
dump, determining which are available for output 526
dump, selecting 182
exit, ARCTVEXT 406
migration level 2, determining which are available for output 530
obtaining information 524
retaining, by DFSMSrmm vital record processing 538
retaining, by expiration date 537
selecting during recycling 500
tape copy needed (TCN) record 403
tape data set exit ARCTDEXT 48, 74
tape drive
3592 Model J considerations 502
tape media contents and OCDS TTOC records, handling 505
tape mount management (TMM)
compared to standard interval migration 64
used with interval migration 23
TAPE subparameter, RECALL parameter
HOLD command 546
RELEASE command 551
tape units used for dump volumes, specifying 160
tape utilization
specifying 502
TAPECOPY
data set name of TAPECOPY of backup tape 190
data set name of TAPECOPY of migration tape 58
optional parameters
ALTERNATEVOLUMES 406
required parameters
ALL 404
TAPECOPY *(continued)*
required parameters *(continued)*
BACKUP 404
INDATASET 407
MIGRATIONLEVEL2 404
take away by recall 117
TAPEDATASET subparameter, RECOVER parameter
HOLD command 547
RELEASE command 551
TAPEEXPIRATIONDATE subparameter, DUMPCLASS parameter, DEFINE command 159
TAPELIST parameter
RECYCLE command 498, 499
TAPEMAXRECALLTASKS parameter, SETSYS command 36
TAPEREPL
required parameters
ORIGINALVOLUMES 406
TARGETUNIT parameter, ARECOVER command 342, 355
tasks
<gerund phrase>
steps for 621, 622, 623, 624
automatic
availability management, SMS-managed storage 152
space management, SMS-managed storage 22
availability management, non-SMS-managed storage 261
backup 163
backup, maximum number in each processor, specifying 174
maintaining control data sets 447
managing storage with DFSMShsm 5
space management, non-SMS-managed storage 79
TCN (tape copy needed) record 403
temporary data set
deleting 186
deleting, SMS-managed storage 51
TERMINAL parameter, LIST command 517
TGTGDS parameter, ARECOVER command 343, 350, 355
threshold of occupancy, backup volume, definition 200
THRESHOLD parameter, ADDVOL command 84
backup volume cleanup process 205
time sequence gaps in CDS records, journal records missing 589
TMM (tape mount management)
compared to standard interval migration 64
used with interval migration 23
TOTAL parameter, BACKVOL command 215
TOVOLUME parameter, RECOVER command 235, 271, 274
tracing
DFSMShsm processing 531
TRAP command 532
TSO subparameter, RECALL parameter
HOLD command 546
TSO subparameter, RECALL parameter *(continued)*
RELEASE command 551
types of backup volumes 148

U

UNASSIGN parameter, DELVOL command 494
uncataloged data sets, backing up 265
UNCATALOGEDDATA subparameter, NONSMSVERSIONS parameter, EXPIREBV command 269
uncataloging
excess dump VTOC copy data sets 182
migrated data sets 582
undamaged CDS, damaged journal, 588
undamaged CDS, full journal, 589
UNIT parameter
ABACKUP command 312
ARECOVER command 351
BACKDS command 268
RECOVER command 224, 235, 271, 274
UNIT parameter, ARECOVER command 352
UNIT subparameter, DUMPCLASS parameter, DEFINE command 160
UNLOAD parameter, SETSYS command, releasing a backup volume 206
unmanaged volumes, example 15
unmatched data sets
backup 186
space management 51
unused space, releasing, SMS-managed storage 54
UPDATEC
optional parameters
JOURNAL 471
required parameters
ALL 471
BACKUPCONTROLDATASET 471
MIGRATIONCONTROLDATASET 471
OFFLINECONTROLDATASET 471
updating recovered CDSs 471
use of
concurrent copy to back up the CDSs 449
data class 8
DFSMShsm with DFSMSrmm 537
JCL to implement enhanced CDS recovery function 464
management classes 8
storage class 8
storage groups 8
user authorization information, obtaining 526
utility data sets
formats of names 80
specifying age for deleting 80

V

valid data on tape volumes, consolidating 495

values, defining for DFSMShsm 567
varying dump invocation 425
VCAT copy data sets 204
verification
  aggregate backup 314
  aggregate recovery 350, 358
VERIFY parameter
  ABACKUP command 312
  ARECOVER command 350
  RECYCLE command 499
VERSION parameter, ARECOVER command 352, 353
VERSIONS parameter
  BACKVOL command 269
  BDELETE command 270
  SETSYS command 262
versions, backup, moving 184
volume
  backup
    cleanup process 205
    spill process 200
    spill volume selection, initial 202
  backup, defining to DFSMShsm 177
  command migration, SMS-managed 74
  commands to directly manage 568
  contention
    migration needs volume 117
    multiple-processing-unit environment 121
    recycle needs volume 119
    single-processing-unit environment 117
    tapes 117
  controlling migration, non-SMS-managed 107
  daily backup, selecting 191
  define for DFSMShsm 567
  determining status 567
  disaster alternate 409
  dump
    defining to DFSMShsm 162
    specifying concurrent copy with 154
    specifying the kind of tape units to use 160
    specifying what to do with 158
  dumping, non-SMS-managed 267
  initial selection
    DASD daily backup 194
    DASD or tape daily backup 198
    spill backup 202
    tape daily backup 196
  making disaster backup copies 406
  migration
    deleting non-SMS-managed data sets with the MIGRATE command 490
    using the FREEVOL command for 490
  migration level 1
    damaged, recovering backup versions from 510
    moving all data sets 490
    using the MIGRATE command 489
  migration level 2, contention 117
volume *(continued)*
  non-SMS-managed
    automatic availability management of 265
    backing up all primary 268
  online, ensuring 306
  pool
    configuration in JES3 88
    defining 85
  pooling
    example 15
    JES2 87
    JES3 89
  recovery, incremental backup 276
  selecting
    backup 201
    DASD and tape daily backup 197
    DASD daily backup 193
    recall, non-SMS-managed 101
    tape daily backup 196
  serialization, dump 181
  source, backup 266
  space management, preventing processing of data sets 131
  subsequent selection
    DASD daily backup 195
    DASD or tape daily backup 199
    DASD spill 203
    spill backup 203
    tape daily backup 197
    tape spill 203
  tape
    alternate, replacing an original with 407
    consolidating valid data 495
    DFSMShsm-owned, copying categories 404
    DFSMShsm-owned, making backup copies 401
    DFSMShsm-owned, making copies 402
    DFSMShsm-owned, protecting 514
    dump, selecting 182
    selecting during recycling 500
  to be backed up, specifying 262
  to be dumped, SMS-managed 152
  to be dumped, specifying 263
  with storage groups 8
VOLUME parameter 86, 88
  BACKDS command 268
  BACKVOL command 207
  MIGRATE command 490
  RECYCLE command 497
volume, damage on an ML1 598
VOLUMEPOOL parameter, DEFINE command 88
VOLUMES parameter, ARECOVER command 351
VSAM considerations for control data sets 472
VSAM data set
  accessing during backup 188
  backing up 188
  GUARANTEED SPACE attribute
    availability management 420
VSAM data set *(continued)*
  GUARANTEED SPACE attribute *(continued)*
    recalling, changes in the storage group 136
    recalling, changes in the storage-class-to-data set mapping 137
    recalling, problems 136
    recalling, problems with conversions of data sets 138
    recovering 421
    recovering, changes in the storage group 421
    recovering, changes in the storage-class-to-data set mapping 422
    recovering, problems 421
    recovering, problems with conversions of data sets 422
    space management 135
  recovering 273
  recovering from a DFSMShsm backup version 230
  restore 237
  restoring from a dump copy 232
  space management 58
VTOC
  copy data set 204
    backup, names 187
    create 186
    volume recovery 276
  dump copies
    example 160
    specifying number to keep 160
VTOCCOPIES subparameter, DUMPCLASS parameter, DEFINE command 161

W

wait for a spill backup volume 203
what our example system shows, space management 21
who can back up data sets by command, specifying 168
withdrawing FlashCopy relationship 244
write dump VTOC copy data set 181
writing to a cartridge-type, single-file format tape during migration 62

X

XMIT parameter, ARECOVER command 351

Z

z/OS images 539

Readers' Comments — We'd Like to Hear from You

z/OS
DFSMShsm Storage Administration Guide

Publication No. SC35-0421-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name

Address

Company or Organization

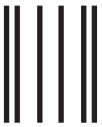
Phone No.

Readers' Comments — We'd Like to Hear from You
SC35-0421-03



Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400



Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line



Program Number: 5694-A01

Printed in USA

SC35-0421-03

