

z/OS



# DFSMS Implementing System-Managed Storage

*Version 2 Release 1*

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 267.

This edition applies to Version 2 Release 1 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1988, 2013.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

**Figures . . . . . vii**

**Tables . . . . . ix**

**About this document . . . . . xi**

*z/OS* information . . . . . xi

**How to send your comments to IBM . . . . . xiii**

If you have a technical problem . . . . . xiii

***z/OS* Version 2 Release 1 summary of changes . . . . . xv**

## **Chapter 1. Introducing System-Managed Storage . . . . . 1**

System-Managed Storage . . . . . 1

DFSMS in the System-Managed Storage Environment 1

Benefits of System-Managed Storage . . . . . 2

Managing Data with SMS . . . . . 6

    Using SMS Classes and Groups . . . . . 6

    Using Aggregate Groups . . . . . 14

    Using Automatic Class Selection Routines . . . 14

    Defining the Storage Management Subsystem

    Configuration . . . . . 16

Software and Hardware Considerations . . . . . 16

Implementing System-Managed Storage . . . . . 17

## **Chapter 2. Planning to Implement System-Managed Storage . . . . . 19**

Implementing to Fit Your Needs . . . . . 19

Using DFSMS FIT to Implement System-Managed

Storage . . . . . 19

Using Milestones to Implement System-Managed

Storage . . . . . 20

    Enabling the Software Base . . . . . 20

    Activating the Storage Management Subsystem 21

    Managing Temporary Data . . . . . 21

    Managing Permanent DASD Data . . . . . 21

    Managing Tape Data . . . . . 22

Using Storage Class to Manage Performance and

Availability . . . . . 22

    Using Cache to Improve Performance for

    Directly-Accessed Data Sets . . . . . 23

    Improving Performance for Sequential Data Sets 25

    Improving Performance for VSAM Data Sets . . 27

    Improving Performance with Hiperbatch . . . 28

    Improving Performance with the Parallel Access

    Volume Option . . . . . 28

    Improving Availability. . . . . 29

    Improving Availability during Data Set Backup 30

    Preallocating Space for Multivolume Data Sets . 32

Managing Space and Availability for Data Sets . . 33

    Managing Data with DFSMSHsm . . . . . 36

    Using SMS with DFSMSHsm Commands . . . . 37

Using SMS with Aggregate Backup and Recovery

Support . . . . . 39

Managing DASD Volumes with SMS . . . . . 40

    Pooling Volumes with Storage Groups . . . . 40

    Selecting Volumes with SMS. . . . . 42

    Managing Virtual I/O with SMS . . . . . 44

    Separating Large Data Sets . . . . . 44

    Avoiding Allocation Failures. . . . . 45

Managing Tape Data with DFSMSrmm . . . . . 45

    Using Management Classes to Manage Tape Data 45

    Using Storage Groups for Volume Pooling . . . 45

    Related Reading for Managing Tape Data with

    DFSMSrmm . . . . . 46

Designing Your ACS Routines . . . . . 46

    Using the ACS Language and Variables . . . . 46

    Using ACS Installation Exits. . . . . 47

    Using ACS Indexing Functions . . . . . 47

    Using FILTLIST Statements . . . . . 48

    Using SELECT Statements . . . . . 48

    Using Advanced ACS Routine Design and

    Coding Techniques . . . . . 50

Placing Your Volumes under System Management 51

    Converting with Data Movement . . . . . 52

    Converting Data In-Place. . . . . 54

Gaining Support for SMS from Your Users . . . . . 56

    Identifying SMS Benefits to Users . . . . . 57

    Defining Data Classes to Simplify Data Set

    Allocations . . . . . 58

    Changing the JCL . . . . . 61

    Identifying the User's Role . . . . . 65

## **Chapter 3. Enabling the Software Base for System-Managed Storage . . . . . 67**

Providing Security in the DFSMS Environment . . . 67

    Protecting System-Managed Data Sets . . . . 68

    Protecting SMS Control Data Sets . . . . . 68

    Protecting Functions and Commands . . . . . 68

    Restricting Access to Fields in the RACF Profile 69

    Restricting Access to Classes and Groups . . . 70

    Protecting ISMF Functions . . . . . 71

Using ISMF to View the Starter Set . . . . . 72

    Viewing the Sample SCDS . . . . . 72

    Viewing the Sample ACS Source Routines . . . 72

Using ISMF to Identify Ineligible Data Sets. . . . 73

    Identifying Unmovable Data Sets and Absolute

    Track Allocation . . . . . 73

    Making Unmovable Data Sets Eligible for System

    Management . . . . . 76

Using ISMF to Manage Storage Devices . . . . . 77

Implementing a System-Determined Block Size . . 77

    How System-Determined Block Size Works. . . 78

    Determining the Space Savings. . . . . 78

## Chapter 4. Activating the Storage Management Subsystem . . . . . 79

Activating a Minimal Configuration . . . . .	79
Managing Data with a Minimal SMS Configuration . . . . .	79
Planning to Activate a Minimal SMS Configuration . . . . .	80
Allocating SMS Control Data Sets . . . . .	80
Calculating the SCDS and ACDS Sizes . . . . .	81
Calculating the COMMDS Size . . . . .	81
Defining the Control Data Sets . . . . .	81
Defining GRS Resource Names for SMS Control Data Sets . . . . .	82
Defining a Minimal SMS Configuration . . . . .	83
Defining the SMS Base Configuration. . . . .	83
Defining the System Group . . . . .	87
Defining SMS Classes and Storage Group . . . . .	87
Creating ACS Routines . . . . .	98
Validating the SCDS . . . . .	102
Defining the Storage Management Subsystem . . . . .	103
Defining How to Activate SMS . . . . .	103
Specifying Members to be Used During IPL . . . . .	104
Defining the ACDS and COMMDS . . . . .	104
Activating a New SMS Configuration . . . . .	105
Activating SMS. . . . .	106
Controlling SMS Processing with MVS Operator Commands . . . . .	106
Enforcing Standards . . . . .	108

## Chapter 5. Managing Temporary Data 109

Planning to Manage Temporary Data . . . . .	110
Defining SMS Classes and Groups . . . . .	111
Defining the Storage Classes . . . . .	111
Defining the Storage Groups . . . . .	111
Creating ACS Routines . . . . .	112
Writing the ACS Routines . . . . .	113
Translating the ACS Routines . . . . .	113
Testing the ACS Routines . . . . .	113
Creating ACS Test Cases. . . . .	113
Running ACS Test Cases. . . . .	116
Initializing DASD Volumes for Large and Prime Storage Groups. . . . .	117
Reactivating the Configuration . . . . .	117
Validating the Configuration . . . . .	117
Activating the Configuration . . . . .	118

## Chapter 6. Managing Permanent Data 119

Identifying Types of Permanent Data . . . . .	119
Planning the Migration of Permanent Data . . . . .	119

## Chapter 7. Managing TSO and HFS Data . . . . . 121

Managing TSO Data . . . . .	121
Choosing a Strategy for Placing Your Volumes under System Management. . . . .	122
Determining a Fallback Strategy . . . . .	123
Designing for TSO Performance and Availability Services . . . . .	124
Designing for TSO Data Backup and Space Management . . . . .	124
Designing the Physical Storage Environment . . . . .	126
Placing TSO Data under System Management . . . . .	126

In-Place Conversion . . . . .	126
Using Data Movement . . . . .	127
Automating Data Allocation Using Data Classes . . . . .	128
Sample Data Classes for TSO Data . . . . .	128
Example: ACS Routine Fragments for Assigning Data Classes. . . . .	130
Defining SMS Constructs for HFS Data. . . . .	132

## Chapter 8. Managing Batch Data . . . 135

Understanding the Benefits of Placing Batch Data under System Management. . . . .	135
Planning Your Migration Strategy . . . . .	136
Improving Batch Performance by Using Sequential Data Striping . . . . .	137
Eliminating Dependencies on Pattern GDGs . . . . .	137
Deciding on a Migration Strategy. . . . .	137
Designing for Batch Performance and Availability Services . . . . .	138
Designing for Batch Data Backup and Space Management . . . . .	139
Managing GDGs . . . . .	139
Managing Non-GDG Batch Data Sets . . . . .	140
Designing the Physical Storage Environment . . . . .	141
Designing Data Classes for Batch Data . . . . .	141
Replacing Pattern DSCBs . . . . .	141
Defining Sequential Data Striping . . . . .	142
Migrating Batch Data. . . . .	142

## Chapter 9. Managing Database Data 145

Understanding the Benefits of Placing Your Database Data under System Management . . . . .	145
Planning Your Migration . . . . .	148
Designing for Database Performance and Availability . . . . .	148
Designing for CICS Data . . . . .	149
Designing for IMS Data . . . . .	153
Designing for DB2 Data . . . . .	157
Designing for Database Data Backup and Recovery . . . . .	162
Designing the Management Class ACS Routine . . . . .	164
Relating DB2 STOGROUPs to SMS Storage Groups . . . . .	166
Example: FILTLIST section for database from storage group ACS routine . . . . .	168
Example: SELECT section for database from storage group ACS routine . . . . .	168
Allocating DB2 Partitioned Table Spaces . . . . .	169

## Chapter 10. Planning for System-Managed Tape . . . . . 171

Optimizing Your Current Tape Environment . . . . .	171
Analyzing Tape Usage with the Volume Mount Analyzer . . . . .	172
Converting Tape Volumes to System Management . . . . .	173
Preparing for Tape Volume Conversion. . . . .	174

## Chapter 11. Optimizing Tape Usage 177

Using the Virtual Tape Server (VTS) to Optimize Tape Media . . . . .	179
Migrating Tape Data Sets to System-Managed Storage . . . . .	179

Changing Jobs that Reference Undefined Generics and Esoterics . . . . .	180
Changing Jobs that Use Uncataloged Tape Data Sets . . . . .	180
Analyzing Expiration Date Codes . . . . .	180
Using Data Set Level Serialization . . . . .	181
Classifying Your Tape Data Sets . . . . .	181
Analyzing Your Current Tape Environment . . . . .	182
Summarizing Tape Usage with the Volume Mount Analyzer GFTAXTR Program . . . . .	183
Analyzing Tape Usage with the Volume Mount Analyzer GFTAVMA Program . . . . .	184
Simulating the Tape Mount Management Environment . . . . .	190
Identifying Exception Tape Data . . . . .	190
Defining the Tape Mount Management Environment . . . . .	193
Implementing Advanced Cartridge Hardware . . . . .	197
Using the Improved Data Recording Capability . . . . .	197
Using the Enhanced Recording Capability . . . . .	198
Using Enhanced Capacity Tape . . . . .	198
Using Automatic Cartridge Loaders . . . . .	198
Defining DASD to Satisfy Buffer Requirements . . . . .	198
Defining SMS Classes and Groups . . . . .	199
Defining Data Classes . . . . .	200
Defining Storage Classes . . . . .	202
Defining Management Classes . . . . .	202
Defining Storage Groups . . . . .	203
Creating ACS Routines . . . . .	203
Creating the Data Class ACS Routine . . . . .	204
Creating the Storage Class ACS Routine . . . . .	206
Creating the Management Class ACS Routine . . . . .	207
Creating the Storage Group ACS Routine . . . . .	208
Using Tape Mount Management Techniques . . . . .	209
Data Set Stacking . . . . .	209
Unit Affinity . . . . .	213
Non-Data Set Stacking Allocations . . . . .	214
Using Volume Reference to System-Managed Data Sets . . . . .	215
Using Volume Reference to Non-System-Managed Data Sets . . . . .	216
Volume Reference Chains . . . . .	218
Tuning DFSMSHsm Operation . . . . .	218
<b>Chapter 12. Managing Tape Volumes . . . . .</b>	<b>219</b>
Understanding the Benefits of System-Managed Tape . . . . .	219
Using Automated Tape Libraries . . . . .	219
Using Manual Tape Libraries . . . . .	220
Using Tape Storage Groups . . . . .	220
Using DFSMSrmm . . . . .	222
Planning Your Migration to System-Managed Tape . . . . .	222
Organizing Your Migration Tasks . . . . .	222

Categorizing Tape Volumes for Migration . . . . .	223
Placing Tape Volumes under System Management . . . . .	223
Defining the Tape Environment . . . . .	223
Defining OAM . . . . .	223
Defining the Storage Classes . . . . .	224
Writing the Storage Class ACS Routine . . . . .	224
Designing the Physical Storage Environment . . . . .	224
Writing the Storage Group ACS Routine . . . . .	225
Defining Data Classes for Tape . . . . .	226
Validating the SMS Configuration . . . . .	226
Creating the Tape Configuration Database . . . . .	227
Setting Up an Automated Tape Library . . . . .	227
Maintaining the Tape Library . . . . .	229
Setting Up a Manual Tape Library . . . . .	229
Managing DFSMSHsm Volumes . . . . .	234
Using DFSMSHsm Messages . . . . .	234
Tracking Volumes in Connected Groups . . . . .	234
Recommendations for Volsers . . . . .	235
Sharing an IODF . . . . .	235

**Appendix A. Sample Project Plan for DFSMS Implementation . . . . . 237**

**Appendix B. Sample Classes, Groups, and ACS Routines . . . . . 243**

Data Classes Used in the Sample ACS Routines . . . . .	243
Storage Classes Used in the Sample ACS Routines . . . . .	247
Management Classes Used in the Sample ACS Routines . . . . .	250
Storage Groups Used in the Sample ACS Routines . . . . .	254

**Appendix C. Installation and User Exits . . . . . 259**

DFSMSdfp Installation Exits . . . . .	259
DFSMSHsm Installation Exits . . . . .	261
DFSMSdss Installation Exit . . . . .	262
MVS Installation Exits . . . . .	262

**Appendix D. Accessibility . . . . . 263**

Accessibility features . . . . .	263
Using assistive technologies . . . . .	263
Keyboard navigation of the user interface . . . . .	263
Dotted decimal syntax diagrams . . . . .	263

**Notices . . . . . 267**

Policy for unsupported hardware . . . . .	268
Minimum supported hardware . . . . .	269
Programming interface information . . . . .	269
Trademarks . . . . .	269

**Index . . . . . 271**



---

## Figures

1. Allocating Data Sets or Storing Objects . . . . .	7	49. Management Class Define Panel, Page 3 of 6 . . . . .	98
2. Using Data Class . . . . .	8	50. Writing an ACS Routine . . . . .	99
3. Using Storage Class . . . . .	10	51. Accessing the ISPF/PDF Editor . . . . .	99
4. Using Management Class . . . . .	11	52. Sample Storage Class ACS Routine for the Minimal Configuration . . . . .	100
5. Using Storage Groups . . . . .	13	53. Sample Storage Group ACS Routine for the Minimal Configuration . . . . .	101
6. Processing ACS Routines . . . . .	15	54. Translating an ACS Routine . . . . .	102
7. Paths for Implementing System-Managed Storage . . . . .	18	55. Specifying the SCDS Base Configuration . . . . .	102
8. Storage Class Define Panel, Page 1 of 2 . . . . .	25	56. Validating the SCDS . . . . .	103
9. Management Class Define Panel, Page 1 of 6 . . . . .	34	57. Storage Group and Volume Status for PRIME80 . . . . .	106
10. Management Class Define Panel, Page 2 of 6 . . . . .	35	58. Output from the DEVSERV command . . . . .	107
11. Management Class Define Panel, Page 3 of 6 . . . . .	35	59. System-Managed Temporary Data . . . . .	110
12. Pool Storage Group Define Panel (Page 1) . . . . .	40	60. Creating ACS Test Cases . . . . .	114
13. Pool Storage Group Define Panel (Page 2) . . . . .	41	61. Defining ACS Test Cases . . . . .	114
14. Using the FILTLIST Statement . . . . .	48	62. ACS Test Case Define Panel, Page 1 of 4 . . . . .	115
15. Example of the Format-1 SELECT Statement . . . . .	49	63. ACS Test Case Panel, Page 2 of 4 . . . . .	116
16. Example of the Format-2 SELECT Statement . . . . .	49	64. Test ACS Routines Panel . . . . .	116
17. Data Class Define Panel, Page 1 . . . . .	58	65. Creating an ACS Output Listing . . . . .	117
18. Data Class Define Panel, Page 2 . . . . .	59	66. Migrating Permanent Data . . . . .	120
19. Data Class Define Panel, Page 4 . . . . .	60	67. Sample Fallback Procedure Using DFSMSDss . . . . .	124
20. Data Class Define Panel, Page 6 . . . . .	60	68. FILTLISTs for TSO Data Used in Management Class ACS Routine . . . . .	125
21. Controlling Management Class Assignments . . . . .	70	69. Management Class ACS Routine for TSO Data . . . . .	126
22. Defining Resource Class Profiles . . . . .	70	70. Sample TSO Data Conversion In-Place . . . . .	126
23. Protecting ISMF Functions . . . . .	72	71. Sample TSO Data Conversion with Movement . . . . .	127
24. Selecting Specified Data Sets Using ISMF . . . . .	74	72. Assigning a Data Class for VSAM Processing . . . . .	131
25. Identifying Unmovable Data Sets . . . . .	74	73. Assigning a Data Class Based on the Low-Level Qualifier . . . . .	132
26. ISMF List of ISAM and Unmovable Data Sets by DSORG . . . . .	75	74. Management Class ACS Routine Fragment for Batch Data . . . . .	141
27. ISMF Data Set List . . . . .	75	75. DFSMSDss Job Migrating Batch Data to System Management . . . . .	143
28. Identifying Absolute Track Allocation using ISMF . . . . .	76	76. FILTLIST Section for CICS from Storage Class ACS Routine . . . . .	151
29. ISMF List of Unmovable Data Sets by ALLOC UNIT . . . . .	76	77. Segment to Permit Special Users to Override SMS allocation . . . . .	152
30. Sample Job for Allocating Control Data Sets . . . . .	82	78. SELECT Section for CICS from Storage Class ACS Routine . . . . .	153
31. Minimal SMS Configuration . . . . .	83	79. FILTLIST Section for IMS from Storage Class ACS Routine . . . . .	155
32. CDS Application Selection Panel . . . . .	84	80. ACS Code to Permit Special Users to Override SMS Allocation . . . . .	156
33. SCDS Base Define Panel, Page 1 of 2 . . . . .	84	81. SELECT Section for IMS from Storage Class ACS Routine . . . . .	157
34. SCDS Base Define Panel, Page 2 of 2 . . . . .	85	82. FILTLIST Section for DB2 from Storage Class ACS Routine . . . . .	159
35. Storage Class Application Selection Panel . . . . .	88	83. Logic to Permit Special Users to Override SMS allocation . . . . .	160
36. Storage Class Define Panel, Page 1 of 2 . . . . .	89	84. SELECT Section for DB2 from Storage Class ACS Routine . . . . .	161
37. Listing Storage Classes Defined in the Base Configuration . . . . .	90	85. SELECT Section for DB2 from Storage Class ACS Routine . . . . .	162
38. Initiating a Copy for the SC1 Storage Class . . . . .	90		
39. Copying the SC1 Storage Class Construct . . . . .	91		
40. Defining a Storage Group for the Minimal Configuration . . . . .	92		
41. Defining Pool Storage Group Attributes Page 1 . . . . .	92		
42. Defining Pool Storage Group Attributes Page 2 . . . . .	93		
43. Defining Storage Group System Status . . . . .	93		
44. Defining Non-Existent Volume in Storage Group . . . . .	94		
45. Defining Volume System Status . . . . .	95		
46. Defining a Management Class for the Minimal Configuration . . . . .	96		
47. Management Class Define Panel, Page 1 of 6 . . . . .	96		
48. Management Class Define Panel, Page 2 of 6 . . . . .	97		

86. FILTLIST Section for Database from Management Class ACS Routine . . . . .	165	100. Sample Storage Class ACS Routine for Tape Mount Management . . . . .	207
87. Management Class ACS Routine Sections for Database Data . . . . .	166	101. Sample Management Class ACS Routine for Tape Mount Management . . . . .	208
88. FILTLIST Section for Database from Storage Group ACS Routine . . . . .	168	102. Sample Storage Group ACS Routine for Tape Mount Management . . . . .	209
89. SELECT Section for Database from Storage Group ACS Routine . . . . .	168	103. Sample ACS Routine to Assign Same Storage Group as Referenced Data Set . . . . .	211
90. Redirecting Tape Allocations to DASD Buffer Managed by DFSMS . . . . .	178	104. Storage Class ACS Routine Fragment to Assign Consistent Device Category . . . . .	212
91. Volume Mount Analyzer Helps You Design Your Tape Mount Management Environment .	183	105. Storage Group ACS Routine Fragment to Assign Consistent Device Category . . . . .	212
92. Savings and Cost Summary in Sample Estimate Report. . . . .	186	106. Sample ACS Routine to Assign Different Storage Group than Referenced . . . . .	216
93. Statistical Mount Savings Breakdown in Sample Estimate Report . . . . .	187	107. Sample ACS Routine to Allow Allocation of Non-System-Managed Data Set . . . . .	217
94. Statistical Volume Savings Breakdown: Sample Estimate Report . . . . .	187	108. Sample ACS Routine to Fail Allocation of Non-System-Managed Data Set . . . . .	217
95. DASD Buffer Cost Breakdown: Sample Estimate Report. . . . .	188	109. Example of Volume Reference Chaining	218
96. Example of a Tape Data Set Size Range Table	194	110. Storage Class Routine Fragment for Tape Data . . . . .	224
97. Savings and Cost Summary. . . . .	195	111. Defining a Tape Storage Group . . . . .	225
98. FILTLIST Section of a Sample Data Class ACS Routine for Tape Mount Management . . . . .	205	112. Storage Group ACS Routine Fragment to Assign Tape Storage Groups . . . . .	226
99. Sample Data Class ACS Routine for Tape Mount Sample Data Class ACS Routine for Tape Mount . . . . .	206	113. Defining a Specific Volume Catalog . . . . .	227
		114. Define Device Parameters / Features	230



---

## Tables

1. When A Data Set, Object, or Volume Becomes System-Managed . . . . .	7	27. Data Sets That Can Be Redirected to DASD	182
2. DFSMShsm Commands Replaced by SMS	37	28. Maximum Gigabyte Allocations by Hour Report . . . . .	188
3. Management Class Attribute Equivalents of DFSMShsm Parameters . . . . .	38	29. Maximum Tape Mounts: 18 Day Summary Report . . . . .	189
4. Large Data Sets (by DASD Type) . . . . .	44	30. First Five Entries from a Top Report: Program Names-LARGE = 600 MB . . . . .	189
5. The SMS View of Parallel Sysplexes - Examples . . . . .	87	31. Savings and Cost Summary: Savings From 3490E Implementation . . . . .	192
6. Management Classes for TSO Data . . . . .	124	32. Management Class Report for IDCAMS Data	194
7. Data Classes for VSAM Data Sets . . . . .	129	33. Sample Maximum Gigabyte Allocations by Hour Report . . . . .	196
8. Data Classes for Physical Sequential Data Sets	129	34. Sample Report of Maximum Tape Allocations	197
9. Data Classes for Libraries . . . . .	129	35. SMS Classes and Groups for Tape Data Sets	200
10. Storage Classes for Batch Data . . . . .	138	36. Sample Management Classes for Tape Mount Management . . . . .	202
11. Management Classes for Batch Data . . . . .	139	37. Sample Storage Groups for Tape Mount Management . . . . .	203
12. Sample Data Classes for Striped and Generation Data Sets . . . . .	142	38. Values for &UNIT ACS Read-Only Variable	213
13. SMS Services for Database Data . . . . .	147	39. AFFing DD Volser Values . . . . .	214
14. CICS Data Sets Requiring High Availability	149	40. Sample Data Classes for Data Sets . . . . .	245
15. CICS Data Sets Having High Write Activity	149	41. Sample Storage Classes for Data Sets	248
16. CICS Data Sets Having High Read Activity	150	42. Sample Management Classes for Data Sets	251
17. IMS Data Sets Requiring High Availability	154	43. Sample DASD Storage Groups . . . . .	255
18. IMS Data Sets Having High Write Activity	154	44. Sample Tape Storage Groups . . . . .	256
19. IMS Data Sets Having High Read Activity	155	45. Sample Tape Libraries . . . . .	256
20. DB2 Data Sets Requiring High Availability	158	46. DFSMSdfp Installation Exits . . . . .	259
21. DB2 Data Sets Having High Write Activity	158	47. DFSMShsm Installation Exits . . . . .	261
22. DB2 Data Sets Having High Read Activity	158	48. DFSMSdss Installation Exit . . . . .	262
23. Storage Classes for Database Data . . . . .	161	49. MVS Installation Exits . . . . .	262
24. Management Classes for Database Data	164		
25. Storage Groups for Database Data . . . . .	167		
26. Data Sets That Must Remain on Tape	181		



---

## About this document

This book introduces system-managed storage, an IBM® automated approach to managing storage resources. More importantly, it describes a variety of storage management tasks and considerations. It helps you plan and implement system-managed storage under the DFSMS™ environment.

This book is intended for system programmers, storage administrators, and data-processing professionals like you. If you are new to system-managed storage and this book, you should start with Chapter 1, “Introducing System-Managed Storage,” on page 1 so that you can familiarize yourself with the concept, benefits, and tasks of system-managed storage and data management with the Storage Management Subsystem (SMS).

For information about accessibility features of z/OS®, for users who have a physical disability, please see Appendix D, “Accessibility,” on page 263.

---

## z/OS information

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

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

To find the complete z/OS library, including the z/OS Information Center, see z/OS Internet Library (<http://www.ibm.com/systems/z/os/zos/bkserv/>).



---

## How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or provide any other feedback that you have.

Use one of the following methods to send your comments:

1. Send an email to [mhvrcfs@us.ibm.com](mailto:mhvrcfs@us.ibm.com).
2. Send an email from the "Contact us" web page for z/OS (<http://www.ibm.com/systems/z/os/zos/webqs.html>).
3. Mail the comments to the following address:  
IBM Corporation  
Attention: MHVRCFS Reader Comments  
Department H6MA, Building 707  
2455 South Road  
Poughkeepsie, NY 12601-5400  
US
4. Fax the comments to us, as follows:  
From the United States and Canada: 1+845+432-9405  
From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:  
z/OS V2R1.0 DFSMS Implementing System-Managed Storage  
SC23-6849-00
- The topic and page number that is related to your comment.
- The text of your comment.

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

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

---

## If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one of the following actions:

- Contact your IBM service representative.
- Call IBM technical support.
- Visit the IBM Support Portal at z/OS support page (<http://www.ibm.com/systems/z/support/>).



---

## **z/OS Version 2 Release 1 summary of changes**

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

- *z/OS Migration, GA32-0889*
- *z/OS Planning for Installation, GA32-0890*
- *z/OS Summary of Message and Interface Changes, SA23-2300*
- *z/OS Introduction and Release Guide, GA32-0887*





---

## Chapter 1. Introducing System-Managed Storage

This chapter introduces system-managed storage and discusses its many benefits. It also explains data management with Storage Management Subsystem (SMS). In addition, it highlights the five key milestones in the phased implementation of system-managed storage.

---

### System-Managed Storage

System-managed storage is the IBM automated approach to managing storage resources. It uses software programs to manage data security, placement, migration, backup, recall, recovery, and deletion so that current data is available when needed, space is made available for creating new data and for extending current data, and obsolete data is removed from storage.

You can tailor system-managed storage to your needs. You define the requirements for performance, security, and availability, along with storage management policies used to automatically manage the direct access, tape, and optical devices used by the operating systems.

---

### DFSMS in the System-Managed Storage Environment

DFSMS functional components and related program products automate and centralize storage management, based on policies your installation defines for availability, performance, space, and security. DFSMS consists of the following functional components:

- DFSMSd<sup>TM</sup>dfp
- DFSMSd<sup>TM</sup>ss
- DFSMSH<sup>TM</sup>sm
- DFSMSr<sup>TM</sup>mm
- DFSMS<sup>TM</sup>stvs

This section also briefly describes the following related program products or features:

- DFSORT<sup>TM</sup>
- RACF<sup>®</sup>
- DFSMS Optimizer
- Tivoli<sup>®</sup> Storage Manager
- CICSVR

The **DFSMSd<sup>TM</sup>dfp** functional component of DFSMS provides the storage, program, data, and device management functions of z/OS. The Storage Management Subsystem (SMS) component of DFSMSd<sup>TM</sup>dfp is fundamental to providing these functions. DFSMSd<sup>TM</sup>dfp provides the foundation for distributed data access, using the Distributed FileManager to support remote access of z/OS data and storage resources from workstations, personal computers, or other authorized systems in a SNA LU 6.2 network. You can also use the z/OS Network File System server to enable a z/OS system to act as a file server to workstations, personal computers, and other authorized systems, such as UNIX systems and z/OS systems that do not share DASDs, in a TCP/IP network.

The **DFSMSd<sup>TM</sup>ss** functional component of DFSMS copies and moves data for z/OS.

The **DFSMSHsm** functional component of DFSMS provides automation for backup, recovery, migration, recall, disaster recovery (using ABARS), and space management functions in the DFSMS environment.

The **DFSMSrmm** functional component of DFSMS provides the management functions for removable media, including tape cartridges and reels.

The **DFSMSStvs** optional feature of DFSMS allows batch VSAM processing concurrently with CICS® online transactions. DFSMSStvs users can run multiple batch jobs and online transactions against VSAM data, in data sets defined as recoverable, with concurrent updates.

**DFSORT** sorts, merges, and copies data sets. It also helps you to analyze data and produce detailed reports using the ICETOOL utility or the OUTFIL function.

**RACF**, a component of the Security Server for z/OS, controls access to data and other resources in operating systems.

The **DFSMS Optimizer** feature provides analysis and simulation information for both SMS and non-SMS data. For more information on the DFSMS Optimizer feature, see *DFSMS Optimizer User's Guide and Reference*.

**Tivoli® Storage Manager** is a client-server licensed product that provides storage management services in a multiplatform computer environment. The backup-archive client program allows users to back up and archive files from their workstations or file servers to storage, and restore and retrieve backup versions and archived copies of files to their local file systems.

You can use the Tivoli Storage Manager for z/OS to back up and recover individual files within the Hierarchical File System (HFS). The entire data set can also be backed up and recovered using DFSMSHsm or DFSMSdss, though less frequently. For example, on an I/O error, you can restore the entire data set using DFSMSHsm or DFSMSdss and then use the Tivoli Storage Manager client to recover individual files that were backed up since the DFSMSHsm or DFSMSdss backup. This should result in faster recoveries.

You can use the **CICSVR** product to apply forward recovery logs against recoverable CICS® VSAM data sets after they have been restored using DFSMSHsm or DFSMSdss backups. The forward recovery logs are written by CICS and CICSTS.

#### **Related Reading:**

---

## **Benefits of System-Managed Storage**

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

The goals of system-managed storage are:

- Improve the use of the storage media; for example, by reducing out-of-space abends and providing a way to set a free-space requirement.

- Reduce the labor involved in storage management by centralizing control, automating tasks, and providing interactive or batch controls for storage administrators.
- Reduce the user's need to be concerned with the physical details of performance, space, and device management. Users can focus on using information instead of managing data.

There are several benefits of system-managed storage.

### **Simplified Data Allocation**

System-managed storage enables users to simplify their data allocations. For example, without using the Storage Management Subsystem, a z/OS user would have to specify the unit and volume on which the system should allocate the data set. The user would also have to calculate the amount of space required for the data set in terms of tracks or cylinders. This means the user has to know the track size of the device which will contain the data set.

With system-managed storage, users can let the system select the specific unit and volume for the allocation. They can also specify size requirements in terms of megabytes (MB) or kilobytes (KB). This means the user does not need to know anything about the physical characteristics of the devices in the installation.

SMS is required if you want to allocate data sets in extended format, or specify compression or extended addressability. It is also required if you want to specify partial release, system-managed buffering, or a secondary volume allocation amount for VSAM data sets.

With DFSMS, you do not need model DSCBs for creating generation data sets.

### **Ensured Data Integrity on New Allocations**

The system provides data integrity for newly allocated data sets that have not been written to. For these data sets, whether SMS managed or non-SMS managed, DFSMSdftp writes a physical end-of-file character at the beginning of the data set when space for the data set is initially allocated.

This makes it unnecessary to OPEN data sets for the sole purpose of writing an EOF and to avoid reading old data if the data set is read immediately after being allocated.

### **Improved Allocation Control**

System-managed storage enables you to set a threshold for free space across a set of direct access storage device (DASD) volumes. During allocation of new data sets, the system prefers those volumes that are below the specified threshold. This allows existing data sets to be extended on the volumes that are above the threshold.

SMS reduces space-related abends on initial allocation or when extending to a new volume through the following:

- Removing the DADSM "five extent limit"
- Spreading the requested allocation space quantity over multiple volumes
- Reducing the requested space quantity by a specified percentage

These do not apply while extending the data set on the same volume.

Volume selection techniques from the secondary list help to avoid problems, such as over-allocation of all new data sets on a newly added

volume until it reaches a high threshold or until the available free space on the volume reaches the same level as other volumes in the storage group.

You can also set a threshold for scratch tape volumes in tape libraries, to ensure enough cartridges are available in the tape library for scratch mounts.

### **Improved Input/Output (I/O) Performance Management**

System-managed storage enables you to improve DASD I/O performance across the installation and at the same time reduce the need for manual tuning by defining performance goals for each class of data. You can use cache statistics recorded in system management facilities (SMF) records to help evaluate performance. You can also improve sequential performance by using striped extended-format sequential or VSAM data sets. The DFSMS environment makes the most effective use of the caching abilities of the IBM 3990 Model 3 and Model 6 Storage Controls, as well as other models.

You can also use the DFSMS Optimizer feature to perform in-depth analysis of high I/O activity data sets, including recommendations for placement and simulations for cache and expanded storage.

For more information on the DFSMS Optimizer feature, see *DFSMS Optimizer User's Guide and Reference*.

### **Automated DASD Space Management**

System-managed storage enables you to automatically reclaim space which is allocated to old and unused data sets or objects. You can define policies that determine how long an unused data set or object resides in *primary storage* (storage devices used for your active data). You can have the system remove obsolete data by migrating the data to other DASD, tape, or optical volumes, or you can have the system delete the data. You can also release allocated but unused space which is assigned to new and active data sets.

### **Tape Mount Management**

System-managed storage lets you fully use the capacity of your tape cartridges and automate tape mounts. Using tape mount management (TMM) methodology, DFSMSHsm can fill tapes to their capacity. With 3592 tape devices, media types such as IBM Enterprise Advanced Tape Cartridge (MEDIA11), recording modes such as EFMT4 and EEFMT4, and the improved data recording capability, you can increase the amount of data that can be written on a single tape cartridge.

**Recommendation:** With TMM, you must analyze tape mounts, modify ACS routines to redirect allocations intended for tape to a DASD pool, then migrate them to tape with the DFSMSHsm interval migration. Alternatively, you can use the IBM Virtual Tape Server (VTS) to fill tape media, reduce tape mounts, and save system resources. For more information, see "Using the Virtual Tape Server (VTS) to Optimize Tape Media" on page 179.

### **System-Managed Tape**

System-managed storage lets you exploit the device technology of new devices without having to change the JCL UNIT parameter. In a multi-library environment, you can select the drive based on the library where the cartridge or volume resides. You can use the IBM TotalStorage™ Enterprise Automated Tape Library (3494 or 3495) to automatically mount tape volumes and manage the inventory in an automated tape library. Similar function is available in a system-managed manual tape library. If you are not using SMS for tape management, you can still access the IBM

TotalStorage Enterprise Automated Tape Library (3494 or 3495) using Basic Tape Library Storage (BTLS) software.

You can use the Virtual Tape Server (VTS), with or without the tape mount management methodology, to optimize your use of tape media. You might still need to use tape mount methodology for small tape data sets, but VTS improves your use of tape media and reduces tape mounts. Use VTS for volumes that don't require removal from the library for offsite storage. VTS integrates the advanced technology provided by the IBM 3590 tape drives, IBM fault-tolerant RAID disk storage, a RISC-based controller, and the IBM 3494 tape library.

### **Automated Storage Management for Object Data**

System-managed storage enables you to fully use tape, DASD and optical cartridge capacity. Using an IBM 3995 Optical Library Dataserver, you can automatically mount optical volumes and manage the inventory in an automated optical library.

**Related Reading:** For more information about object data, object storage groups, and object backup backup storage groups, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

### **Improved Data Availability Management**

With system-managed storage, you can provide different backup requirements to data residing on the same DASD volume. Thus, you do not have to treat all data on a single volume the same way.

You can use DFSMSHsm to automatically back up the following:

- CICS databases
- DATABASE 2™ (DB2®) databases
- Partitioned data sets extended (PDSEs)
- Physical sequential data sets
- Partitioned data sets
- Virtual storage access method (VSAM) data sets
- Direct access data sets
- Hierarchical file system (HFS) data sets

**Tip:** For HFS data sets, the Tivoli Storage Manager/UNIX System Services client backs up the files and directories inside the HFS data set and DFSMSHsm backs up the data set itself. You can set up your procedures to back up the whole file system less frequently using guaranteed backup frequency, and then use the Tivoli Storage Manager client to back up data within the file system more often. This reduces the total recovery time since it uses the high bandwidth of DFSMSHsm to perform backups and recoveries in case the file system becomes inaccessible.

You can also back up other types of data and use point-in-time copy to maintain access to critical data sets while they are being backed up. Concurrent copy, virtual concurrent copy, SnapShot, and FlashCopy®, along with backup-while-open, have an added advantage in that it avoids invalidating a backup of a CICS VSAM KSDS due to a control area or control interval split.

To backup and recover critical applications requiring concurrent action, such as for disaster recovery, you can create a logical grouping of data sets known as an aggregate group. You define an aggregate group by selecting a management class and specifying backup attributes (such as type of

storage medium, retention period, or destination) which all data sets in the group share. DFSMShsm uses the aggregate backup and recovery support (ABARS) to manage the aggregate group backup process. You can also use ABARS to transfer applications between sites.

You can use the same management class attributes for multiple aggregate groups whose backup copies have the same management needs. These backup attributes are used to manage backup copies and also to create the proper environment for backed-up data sets during recovery. During aggregate recovery, data sets backed up as migrated data sets can be returned to the same level as when backed up, or they can all be recovered to ML1 DASD or ML2 tape. All data sets backed up from user volumes are returned to user volumes.

#### **Simplified Movement of Data to Different Device Types**

With system-managed storage, you can move data to new volumes without requiring users to update their job control language (JCL). Because users in a DFSMS environment do not need to specify the unit and volume which contains their data, it does not matter to them if their data resides on a specific volume or device type. This lets you easily replace old devices with new ones.

You can also use system-determined block sizes to automatically reblock physical sequential and partitioned data sets that can be reblocked.

---

## **Managing Data with SMS**

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

### **Using SMS Classes and Groups**

On systems that do not use DFSMS, storage management consists mostly of manual operations performed on individual data sets, and manual and automated operations performed on volumes. With SMS, you can automate storage management for individual data sets and objects, and for DASD, optical, and tape volumes. You use SMS classes and groups to define the goals and requirements that the system should meet for a data set or object. Figure 1 on page 7 shows the relationship of the classes and groups to your goals and requirements.

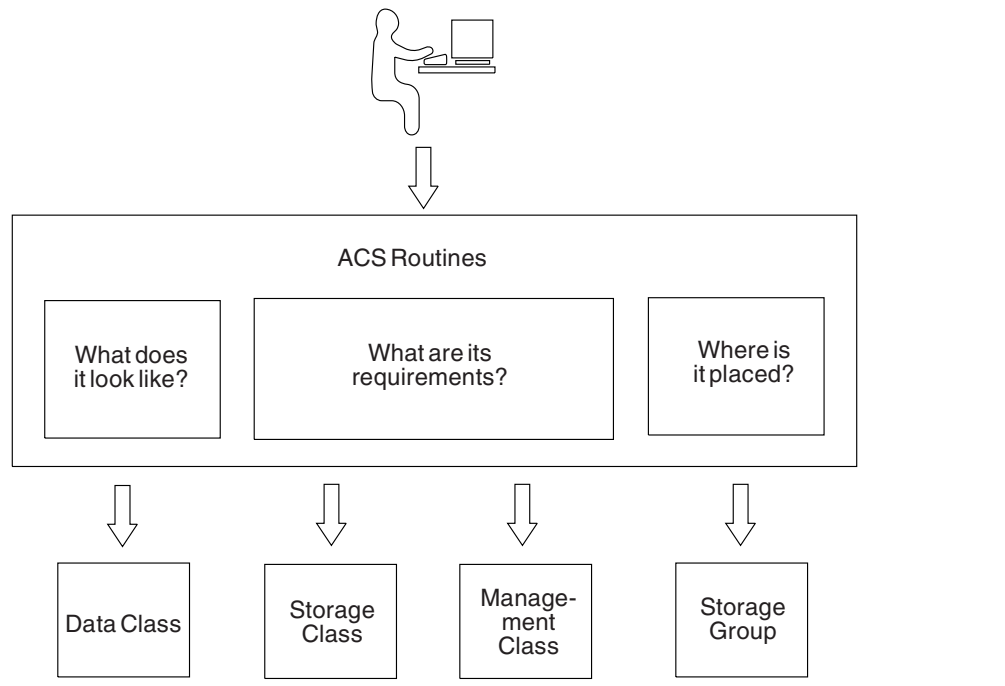


Figure 1. Allocating Data Sets or Storing Objects. You use data class to define model allocation characteristics for data sets; storage class to define performance and availability goals; management class to define backup and retention requirements; and storage group to create logical groupings of volumes to be managed as a unit.

Table 1 shows how a data set, object, DASD volume, tape volume, or optical volume becomes *system-managed*.

Table 1. When A Data Set, Object, or Volume Becomes System-Managed

	DASD	Optical	Tape
Data Set <sup>1</sup>	Assign Storage Class	Not applicable	Not <i>system-managed</i> <sup>2</sup>
Object <sup>3</sup>	Stored	Stored	Stored
Volume	Assign Storage Group	Assign Object or Object Backup Storage Group	Assign Storage Group <sup>4</sup>

#### Rules:

1. A DASD data set is *system-managed* if you assign it a storage class. If you do not assign a storage class, the data set is directed to a *non-system-managed* DASD or tape volume-one that is not assigned to a storage group-unless you specify a specific system-managed tape volume, in which case the data set is allocated on system-managed tape.
2. You can assign a storage class to a tape data set to direct it to a *system-managed* tape volume. However, only the tape volume is considered *system-managed*, not the data set.
3. OAM objects each have a storage class; therefore, objects are *system-managed*. The optical or tape volume on which the object resides is also *system-managed*.
4. Tape volumes are added to tape storage groups in tape libraries when the tape data set is created.



## Using Data Classes

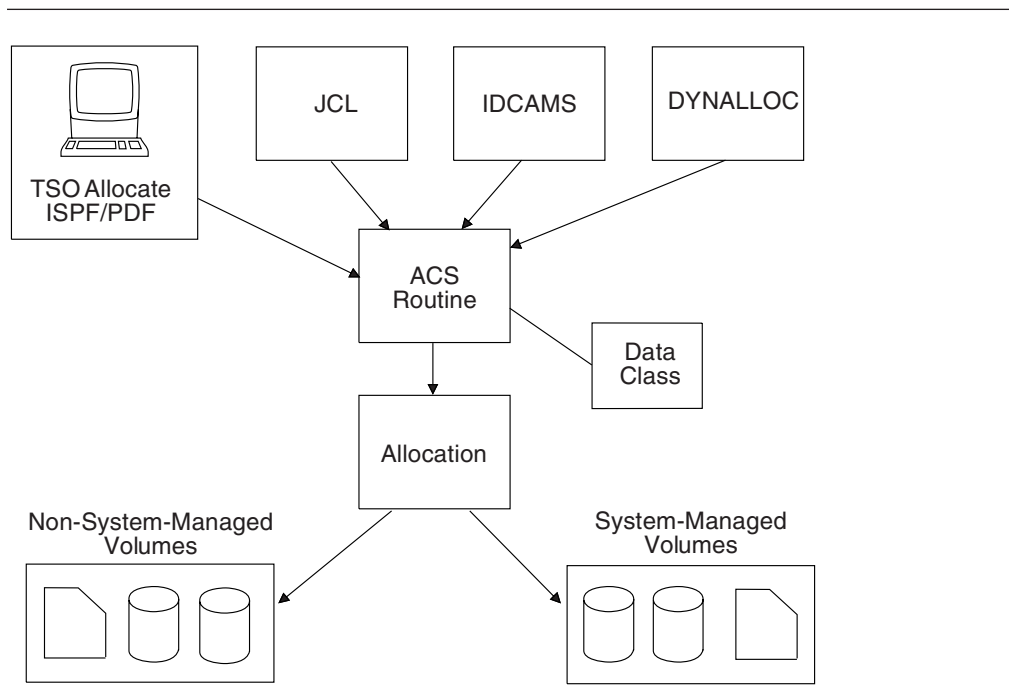
A *data class* is a collection of allocation and space attributes that you define. It is used when data sets are created. You can simplify data set allocation for your users by defining data classes that contain standard data set allocation attributes. You can use data classes with both system-managed and non-system-managed data sets, but some data class characteristics are only available with system-managed requests.

Data class attributes define space and data characteristics of data sets that are normally specified on JCL DD statements, TSO/E ALLOCATE commands, access method services (IDCAMS) DEFINE commands, dynamic allocation requests, and ISPF/PDF panels. For tape data sets, data class attributes can also specify the type of cartridge and recording method, and if the data is to be compacted. Users then need only specify the appropriate data classes to create standardized data sets.

You can use data class to allocate sequential and VSAM data sets in extended format for the benefits of compression (sequential and VSAM KSDS), striping, and large data set sizes (VSAM).

You can also use the data class automatic class selection (ACS) routine to automatically assign data classes to new data sets. For example, data sets with the low-level qualifiers LIST, LISTING, OUTLIST, or LINKLIST are usually utility output data sets with similar allocation requirements, and can all be assigned the same data class.

Figure 2 shows that data sets can be assigned a data class during data set creation.



*Figure 2. Using Data Class.* Data classes can be used for new allocations of both system-managed and non-system-managed data sets.

If you change a data class definition, the changes only affect new allocations. Existing data sets allocated with the data class are not changed, except for the system-managed buffering attribute. With system-managed buffering, the data class attributes are retrieved and used when the data set is opened.



**Recommendations:**

- Assign data classes to system-managed and non-system-managed data, even though data class is optional. The data class name is kept in the catalog entry for system-managed data sets for future reference. The data class name is not saved for non-system-managed data sets, although the allocation attributes in the data class are used to allocate the data set.
- For objects on tape, do not assign a data class using the ACS routines. To assign a data class, specify the name of that data class on the SETOAM command.

**Using Storage Classes**

A *storage class* is a collection of performance goals and availability and accessibility requirements that you define. It is used to select a device to meet those goals and requirements. Only system-managed data sets and objects can be assigned a storage class. Storage classes free users from having to know about the physical characteristics of storage devices and manually placing their data on appropriate devices.

Some of the availability requirements you can specify with storage classes can only be met by DASD volumes attached through one of the following storage control devices, or a similar device:

- 3990 Model 3
- 3990 Model 6
- RAMAC™ Array Subsystem
- Enterprise Storage Server®™ (ESS)

Some of the attributes you can specify require the use of the dual copy device of the 3990 Model 3 or Model 6 Storage Control or the RAID characteristics of RAMAC or ESS. The performance goals you set can be met through devices attached through storage controls with or without cache.

Figure 3 on page 10 shows the storage control configurations needed to use all storage class attribute values.

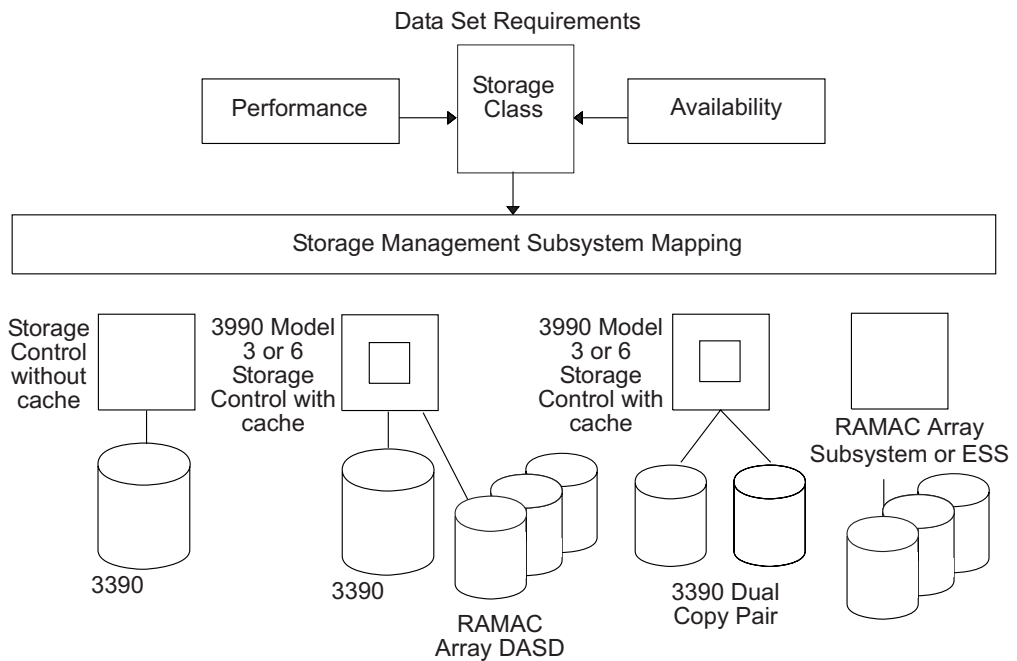


Figure 3. Using Storage Class. Storage classes make the best use of fault-tolerant devices

You can use the storage class Availability attributes to assign a data set to fault-tolerant devices, in order to ensure continuous availability for the data set. The available fault-tolerant devices include dual copy devices and RAID architecture devices, such as RAMAC and ESS.

You can use the storage class Accessibility attribute to request that point-in-time copy be used when data sets or volumes are backed up.

You can specify an I/O response time objective with storage class. During data allocation, the system attempts to select the available volume closest to the specified performance objective.

For objects, the system uses the performance goals you set in the storage class to place the object on DASD, optical, or tape volumes. The storage class is assigned to an object when it is stored or when the object is transited. The ACS routines can override this assignment.

If you change a storage class definition, the changes affect the performance service levels of existing data sets that are assigned that class when the data sets are subsequently opened. However, the definitional changes do not affect the location or allocation characteristics of existing data sets.

### Using Management Classes

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

If you do not explicitly assign a management class to a system-managed data set, the system uses the default management class. You can define your own default management class when you define your SMS base configuration.

For objects, you can do one of the following:

- Assign a management class when it is stored. Assign it explicitly, using the default specified for the collection the object belongs to, or use the management class ACS routine for a new collection.
- Assign a new management class when the object is transited.
- Change the management class by using the OAM Application Programming Interface (OSREQ CHANGE function).

The ACS routines can override this assignment for objects.

Figure 4 shows that you can use management class attributes to perform the following tasks:

- Use early migration for old generations of a generation data group (GDG) by specifying the maximum number of generations to be kept on primary storage, and determine what to do with rolled-off generation data sets.
- Delete selected old and unused data sets from DASD volumes.
- Release allocated but unused space from data sets.
- Migrate unused data sets to tape or DASD volumes.
- Specify how often to back up data sets, and whether point-in-time copy should be used during backup.
- Specify how many backup versions to keep for data sets.
- Specify how long to save backup versions.
- Specify the number of versions of aggregate backups to keep and how long to retain those versions.
- Specify the number of backup copies of objects (1 or 2)
- Establish the expiration date for objects.
- Establish transition criteria for objects.
- Indicate if automatic backup is needed for objects.

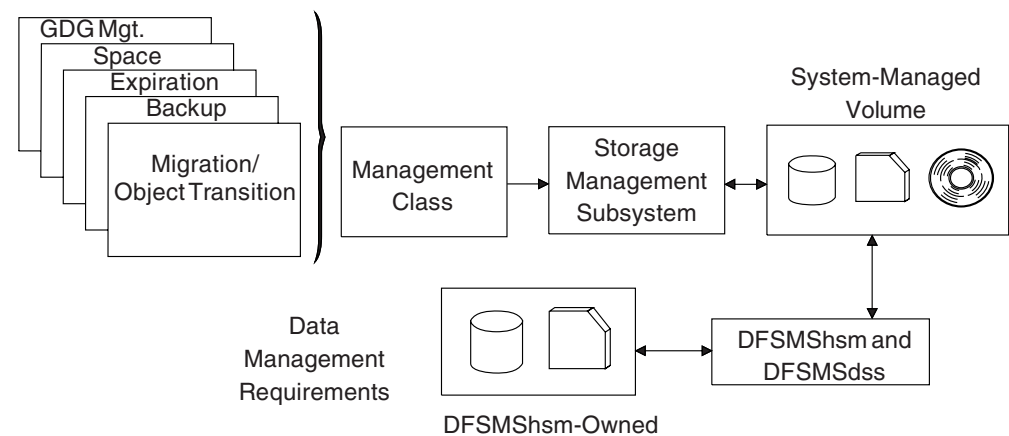


Figure 4. Using Management Class

By classifying your data according to management requirements, you can define unique management classes to fully automate your data set and object management. For example, you can use management classes to control the migration of CICS user databases, DB2 user databases and archive logs, test systems and their associated data sets, and IMS™ archive logs. You can specify that

DB2 image copies, and IMS image copies and change accumulation logs, be written to primary volumes and then migrated directly to migration level 2 tape volumes.

For objects, you use the class transition attributes to define when an object is eligible for a change in its performance objectives or management characteristics. For example, after a certain number of days you might want to move an object from a high-performance DASD volume to a slower optical volume. You can also use the management class to specify that the object should have a backup copy made when the OAM Storage Management Component (OSMC) is running.

If you change a management class definition, the changes affect the management requirements of existing data sets and objects that are assigned that class.

You can reassign management classes when data sets are renamed.

## Using Storage Groups

A *storage group* is a collection of storage volumes and attributes that you define. The collection can be a group of any of the following storage volumes:

- System paging volumes
- DASD volumes (actual or virtual)
- Tape volumes
- Optical volumes
- Combination of DASD and optical volumes that look alike
- DASD, tape and optical volumes treated as a single object storage hierarchy

Storage groups, along with storage classes, help reduce the requirement for users to understand the physical characteristics of the storage devices which contain their data.

You can direct new data sets to as many as 15 storage groups, although only one storage group is selected for the allocation. The system uses the storage class attributes, volume and storage group SMS status, MVS™ volume status, and available free space to determine the volume selected for the allocation. In a tape environment, you can also use tape storage groups to direct a new tape data set to an automated or manual tape library.

DFSMSHsm uses some of the storage group attributes to determine if the volumes in the storage group are eligible for automatic space or availability management.

Figure 5 on page 13 is an example of using storage groups to group storage volumes for specific purposes.

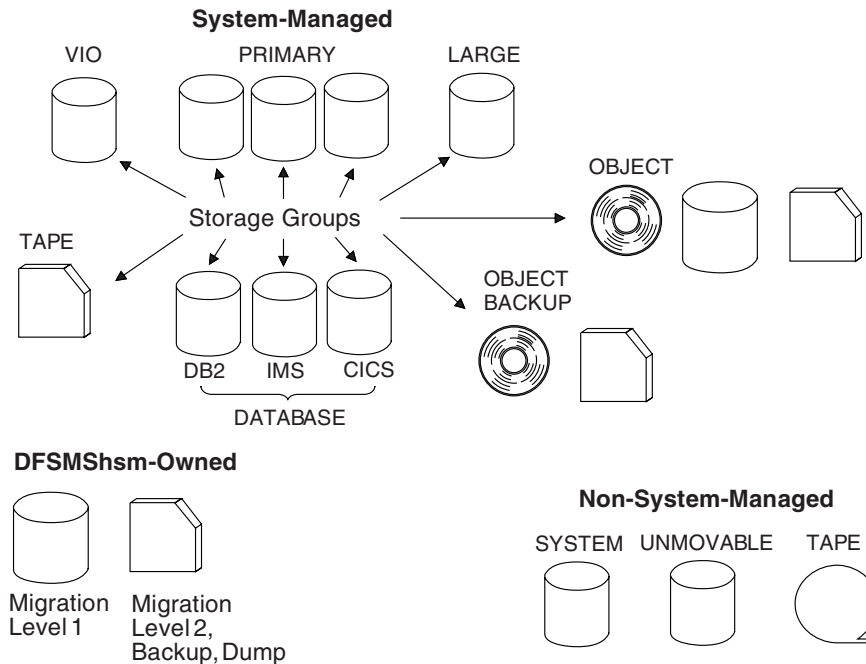


Figure 5. Using Storage Groups. In this example, DASD volumes are grouped so that primary data sets, large data sets, DB2 data, IMS data, and CICS data are all separated.

The virtual input/output (VIO) storage group uses system paging volumes for small temporary data sets. The tape storage groups contain tape volumes that are held in tape libraries. The object storage group can span optical, DASD and tape volumes. An object backup storage group can contain either optical or tape volumes within one OAM invocation. Some volumes are not system-managed, and DFSMSHsm owns other volumes for use in data backup and migration. DFSMSHsm migration level 2 tape cartridges can be system-managed if you assign them to a tape storage group.

You can use data-set-size-based storage groups to help you deal with free-space fragmentation, and reduce or eliminate the need to perform DFSMSdss DEFrag processing. See “Pooling Volumes with Storage Groups” on page 40 for more information.

For objects, there are two types of storage groups: object and object backup. OAM assigns an object storage group when the object is stored. The first time an object is stored to a collection, the storage group ACS routine can override this assignment. You can specify one or two object backup storage groups for each object storage group.

**Recommendation:** Discourage users from directly requesting specific devices. Unlike data, storage, and management classes, users cannot specify a storage group when allocating a data set, although they can specify a unit and volume. Whether or not you honor their unit and volume request is your decision, but it is more effective for your users to specify the logical storage requirements of their data by storage and management class, which you can then verify in the automatic class selection routines.

## Using Aggregate Groups

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

You can use aggregate groups as a supplement to using management class for applications that are critical to your business. You can associate an aggregate group with a management class. The management class specifies backup attributes for the aggregate group, such as the copy technique for backing up DASD data sets on primary volumes, the number of aggregate versions to retain, and how long to retain versions. Aggregate groups simplify the control of backup and recovery of critical data sets and applications.

Although SMS must be used on the system where the backups are performed, you can recover aggregate groups to systems that are not using SMS. You can use aggregate groups to transfer applications to other data processing installations or migrate applications to newly-installed DASD volumes. You can transfer the application's migrated data, along with its active data, without recalling the migrated data.

## Using Automatic Class Selection Routines

You use automatic class selection (ACS) routines to assign class and storage group definitions to data sets and objects. You write ACS routines using the ACS language, which is a high-level programming language. Once written, you use the ACS translator to translate the routines to object form so they can be stored in the SMS configuration.

The ACS language contains a number of read-only variables, which you can use to analyze new data allocations. For example, you can use the read-only variable `&DSN` to make class and group assignments based on data set or object collection name, or `&LLQ` to make assignments based on the low-level qualifier of the data set or object collection name. You cannot alter the value of read-only variables.

You can use another read-only variable, `&SECLABEL`, to assign storage groups based on the type of information in the data set. For example, you might want to store all of the data for a classified project on specific sets of volumes.

You use the four read-write variables to assign the class or storage group you determine for the data set or object, based on the routine you are writing. For example, you use the `&STORCLAS` variable to assign a storage class to a data set or object.

**Related Reading:** For a detailed description of the ACS language and its variables, see *z/OS DFSMSdfp Storage Administration*.

For each SMS configuration, you can write as many as four routines: one each for data class, storage class, management class, and storage group. Use ISMF to create, translate, validate and test the routines.

Figure 6 on page 15 shows the order in which ACS routines are processed. Data can become system-managed if the storage class routine assigns a storage class to the data, or if a user-specified storage class is assigned to the data. If this routine does not assign a storage class to the data, the data cannot reside on a system-managed volume, unless a specific system-managed tape volume is

specified, in which case the data set is allocated on system-managed tape.

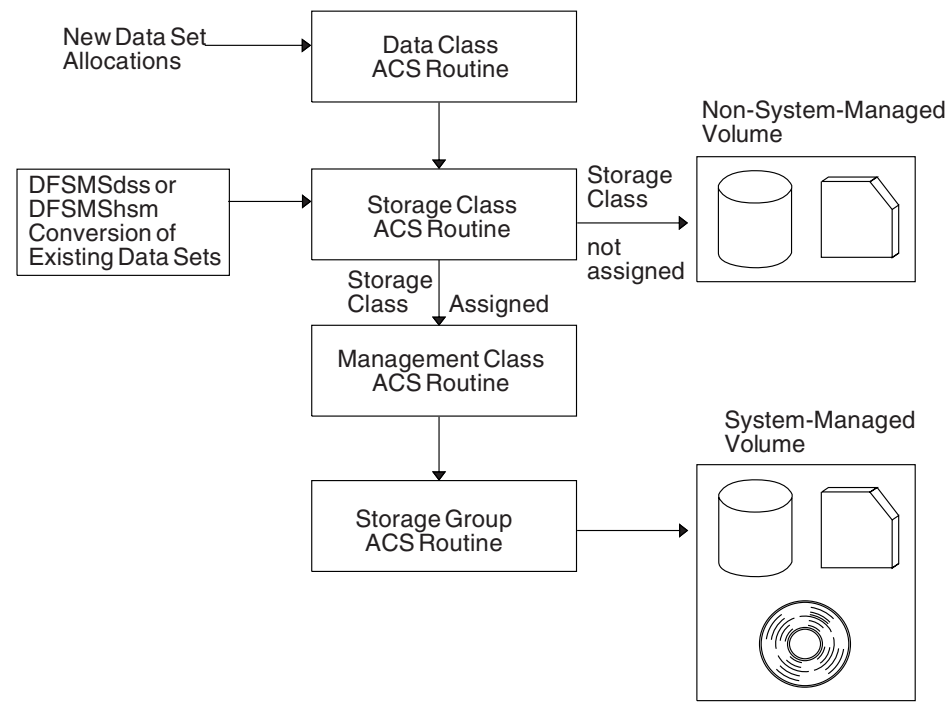


Figure 6. Processing ACS Routines

Because data allocations, whether dynamic or through JCL, are processed through ACS routines, you can enforce installation standards for data allocation on system-managed and non-system-managed volumes. ACS routines also enable you to override user specifications for data, storage, and management class, and requests for specific storage volumes.

You can use the ACS routines to determine the SMS classes for data sets created by the Distributed FileManager/MVS. If a remote user does not specify a storage class, and if the ACS routines decide that the data set should not be system-managed, the Distributed FileManager/MVS terminates the creation process immediately and returns an error reply message to the source. Therefore, when you construct your ACS routines, consider the potential data set creation requests of remote users.

You can also use your ACS routines to detect a reference to non-SMS-managed data sets using VOL=REF, and then either allow or fail the referencing allocation. This is done by testing the &ANYVOL or &ALLVOL read-only variable for a value of 'REF=NS'. This gives the ACS routines control over whether a new, non-SMS-managed data set can be allocated on a non-SMS-managed volume or not. SMS fails the allocation if the ACS routines attempt to make the referencing data set SMS-managed, since this could cause problems attempting to locate that data set with DISP=OLD or DISP=SHR and lead to potential data integrity problems.

For data set allocations that use volume referencing or unit affinity, your ACS routines can determine the storage residency of the referenced data sets.



## Defining the Storage Management Subsystem Configuration

An SMS configuration is composed of a set of data class, management class, storage class, storage group, optical library and drive definitions, tape library definitions, and ACS routines to assign the classes and groups. It also includes the aggregate group definitions and the SMS base configuration. The SMS base configuration contains default information such as default management class and default device geometry. It also identifies the systems and system groups (or a combination of both) in the installation for which the subsystem manages storage.

This information is stored in SMS *control data sets*, which are VSAM linear data sets. You can define these control data sets using the access method services DEFINE CLUSTER command.

**Related Reading:** For detailed information on creating SMS control data sets, see *z/OS DFSMSdfp Storage Administration*.

You must define the control data sets before activating SMS. Although you only need to allocate the data sets from one system, the active control data set (ACDS) and communications data set (COMMDS) must reside on a device that can be accessed by every system to be managed with the SMS configuration.

SMS uses the following types of control data sets:

### Source Control Data Set (SCDS)

This contains the information that defines a single storage management policy, called an SMS configuration. You can define more than one SCDS, but only one can be used to activate a configuration at any given time.

### Active Control Data Set (ACDS)

This contains the output from an SCDS that has been activated to control the storage management policy for the installation. When you activate a configuration, SMS copies the existing configuration from the specified SCDS into the ACDS. By using copies of the SMS classes, groups, volumes, optical libraries, optical drives, tape libraries, and ACS routines rather than the originals, you can change the current storage management policy without disrupting it. For example, while SMS uses the ACDS, you can perform the following actions:

- Create a copy of the ACDS
- Create a backup copy of an SCDS
- Modify an SCDS
- Define a new SCDS

The ACDS must reside on a shared device to ensure that all systems in the installation use the same active configuration.

### Communications Data Set (COMMDS)

Contains the name of the ACDS and enables communication between SMS systems in a multisystem environment. The COMMDS also contains space statistics, SMS status, and MVS status for each system-managed volume.

**Recommendation:** Although only one COMMDS is used at a time for an SMS installation, ensure that you have more COMMDSs on different volumes for recovery purposes.

---

## Software and Hardware Considerations

See *z/OS DFSMS Using the New Functions* for discussions on coexistence requirements and hardware considerations.



---

## Implementing System-Managed Storage

You can implement system-managed storage in several ways. This publication describes implementing in phases or milestones, and also refers you to the DFSMS Fast Implementation Techniques (FIT), which you can use to tailor your implementation. These two methods are some of the most widely used methods of implementing system-managed storage.

You do not have to implement and use all of the functions in SMS. For example, you can implement system-managed tape functions first, without also implementing SMS on DASD. You can also set up a special pool of volumes (a storage group) to only exploit the functions provided by extended format data sets, such as compression, striping, system-managed buffering, partial release, and candidate volume space amount, to name just a few.

You can effectively implement system-managed storage in phases or milestones. The milestones are identifiable DFSMS implementation events that provide measurable benefits. You begin with low-risk implementation activities that establish a base for the staged migration of your data to system management. In later milestones, your earlier experience is used to achieve greater data and storage automation. The following five key milestones mark the phased implementation of system-managed storage:

- Enabling the System-Managed Software Base
- Activating the Storage Management Subsystem
- Managing Temporary Data
- Managing Permanent Data
- Managing Tape Data
  - Optimizing Tape Data
  - Managing Tape Volumes

These milestones can be implemented in a different order, based on your storage priorities. Figure 7 on page 18 shows common implementation paths.

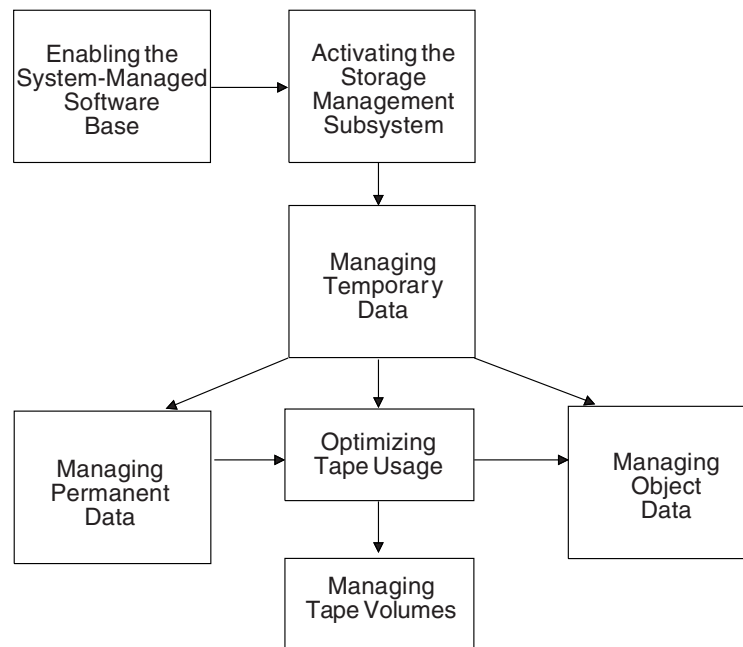


Figure 7. Paths for Implementing System-Managed Storage

You can use the DFSMS Fast Implementation Techniques (FIT) to guide you in implementing DFSMS quickly and simply. DFSMS FIT uses a question-and-answer approach and a data classification process to create a DFSMS design tailored to your installation. DFSMS FIT also includes a number of tools, sample jobs and code, and actual installation examples to help shorten the implementation process.

You can also use IBM NaviQuest for z/OS in conjunction with DFSMS FIT.

**Related Reading:**

- For more information on implementing DFSMS, see “Using Milestones to Implement System-Managed Storage” on page 20.
- For information about DFSMS FIT, see “Using DFSMS FIT to Implement System-Managed Storage” on page 19.
- This book does not discuss how to implement system-managed storage to support objects. If you are implementing the DFSMS environment only to support objects, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for guidance.
- Alternatively, if your goal is to manage both data and objects in the DFSMS environment, consider implementing the first three milestones using this guide first, and then using *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* to help you customize the DFSMS environment for objects.

---

## Chapter 2. Planning to Implement System-Managed Storage

This chapter highlights major DFSMS services and describes considerations for implementing system-managed storage, including:

- Using DFSMS Fast Implementation Technique (FIT) to implement system-managed storage
- Using milestones to implement system-managed storage
- Planning for DASD data set performance and availability requirements
- Planning for DASD data set space, backup, and recovery requirements
- Planning for DASD application disaster/backup and recovery
- Planning for management of DASD volumes
- Planning for the design of your ACS routines
- Determining a DASD data migration approach
- Determining how to present DFSMS to your users
- Determining how to optimize tape usage

**Related Reading:** For a sample project plan for DFSMS implementation, see Appendix A, “Sample Project Plan for DFSMS Implementation,” on page 237.

---

### Implementing to Fit Your Needs

You can implement SMS so that it fits your specific needs. You do not have to implement and use all of the functions in SMS. Rather, you can implement the functions you are most interested in first.

For example, you can implement system-managed tape functions without also implementing SMS on DASD. You can also set up a special pool of volumes (a storage group) to only exploit the functions provided by extended format data sets, such as compression, striping, system-managed buffering (SMB), partial release, and candidate volume space amount, to name just a few. You can put all your data (for example, database and TSO) in a pool of one or more storage groups and assign them appropriate policies at the storage group level to implement DFSMSHsm operations in stages, or to benefit from such SMS features as compression, extended format, striping, and record-level sharing (RLS).

---

### Using DFSMS FIT to Implement System-Managed Storage

The DFSMS FIT process is a proven technique for implementing DFSMS in phases. It uses a question-and-answer approach to create a DFSMS design tailored to your installation's needs, and a data classification system that lets you use your data set naming standards that are already in place and helps you quickly identify the different types of data to which you want to assign specific data set level, SMS-management policies.

DFSMS FIT also includes the following features:

- Installation examples that you can use for guidance
- Sample jobs to help you perform the following tasks:
  - Set up your DFSMS environment
  - Create ACS routines quickly, by using the ACS code fragments provided as models
  - Migrate data to SMS-managed volumes

- Operate and control your DFSMS environment after data migration

In conjunction with DFSMS FIT, you can use NaviQuest, a testing and reporting tool developed specifically for DFSMS FIT. With NaviQuest you can perform the following tasks:

- Automatically test your DFSMS configuration
- Automatically test your ACS routines
- Perform storage reporting, through ISMF and with DCOLLECT and VMA data
- Report functions on ISMF table data
- Use REXX EXECs to run ISMF functions in batch
- Assist the storage administrator in creating ACS routines

**Related Reading:**

- For more information about DFSMS FIT, see *Get DFSMS FIT: Fast Implementation Techniques*.
- For more information about NaviQuest, see *z/OS DFSMSdfp Storage Administration*.

---

## Using Milestones to Implement System-Managed Storage

You can use milestones to implement system-managed storage in a flexible, low-risk, phased migration. You can use the DFSMS Fast Implementation Techniques (FIT) in conjunction with the milestones approach to plan an easy and quick implementation.

The *starter set* shipped with DFSMS consists of a sample base configuration and sample Automatic Class Selection (ACS) routines that can assist you in implementing system-managed storage. It also contains an SMS configuration, which is a VSAM linear data set with typical SMS classes and groups. This sample source configuration data set (SCDS) contains SMS classes and groups that can be used for your first activation of SMS and for later milestones that manage more of your data.

**Tip:** The examples in this book might be more current than samples in the starter set.

You can use these samples along with this book to phase your implementation of system-managed storage. The completion of each phase marks a milestone in implementing system-managed storage.

There are five major implementation phases:

- Enabling the software base
- Activating the storage management subsystem
- Managing temporary data
- Managing permanent data
- Managing tape data

Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243 contains sample ACS routines that correspond to each of the milestones that have SMS active.

### Enabling the Software Base

This milestone focuses on implementing DFSMS capabilities. During this phase, you can perform the following tasks:

- Protect critical storage administration facilities before SMS activation

- Familiarize yourself with Interactive Storage Management Facility (ISMF) applications for the storage administrator and with the starter set
- Use the ISMF Data Set Application to determine eligibility of your data sets for system management
- Use the ISMF Volume Application to initialize DASD volumes and set the cache-capable 3990 storage control's extended functions
- Implement system-determined block size for your data sets

Chapter 3, “Enabling the Software Base for System-Managed Storage,” on page 67 discusses this milestone.

## Activating the Storage Management Subsystem

You can use the sample ACS routines for the activating milestone and the SMS configuration in the sample SCDS to assist you with the following tasks:

- Establish a minimal SMS configuration
- Define SMS to z/OS as a subsystem
- Control SMS processing with operator commands
- Use simplified job control language to allocate data sets
- Enforce standards

Chapter 4, “Activating the Storage Management Subsystem,” on page 79 provides a detailed description.

## Managing Temporary Data

You can use the sample ACS routines for the temporary milestone and the SMS configuration in the sample SCDS, along with the minimal configuration you developed using Chapter 4, “Activating the Storage Management Subsystem,” on page 79, to let SMS allocate your temporary data on system-managed volumes. During this phase, you use DFSMSHsm to clean up temporary data sets left on system-managed volumes.

Chapter 5, “Managing Temporary Data,” on page 109 describes how to tailor your DFSMS environment to manage temporary data.

## Managing Permanent DASD Data

Before you manage permanent data, you should understand the service requirements for your data and how data and storage management is currently done.

**Recommendation:** Develop the following documents before you start to migrate permanent data to system management:

- A storage management plan that documents your storage administration group's strategy for meeting storage requirements
- Formal service-level agreements that describe the services that you agree to provide

You can use the sample ACS routines for the permanent milestone and the SMS configuration in the sample SCDS, along with Chapter 6, “Managing Permanent Data,” on page 119, to assist your migration to system-managed permanent data. Managing permanent data is divided into the following stages that are based on your major data set classifications:

**Managing TSO data**

Migrating TSO and HFS data is described in Chapter 7, “Managing TSO and HFS Data,” on page 121.

**Managing batch data**

Migrating batch data is described in Chapter 8, “Managing Batch Data,” on page 135.

**Managing database data**

Migrating database data is described in Chapter 9, “Managing Database Data,” on page 145, which identifies requirements for CICS/VSAM, IMS, and DB2 data.

**Related Reading:** For information on using DFSMS FIT to create a DFSMS design tailored to your environment and requirements, and for information on using NaviQuest to test and validate your DFSMS configuration, see “Using DFSMS FIT to Implement System-Managed Storage” on page 19.

## Managing Tape Data

You can use the sample ACS routines for the tape milestone and the configuration you developed to manage permanent data to assist your migration of tape data to system management.

Optimize your tape operation by using tape mount management techniques to control tape usage to achieve the following benefits:

- Reduced operator mounts
- Reduced tape library size
- Improved use of tape media
- Improved batch job turnaround

You can also use the IBM Virtual Tape Server (VTS) with or without tape mount management to optimize your use of tape media.

**Related Reading:**

- For more information about tape mount management, see Chapter 11, “Optimizing Tape Usage,” on page 177.
- After using tape mount management, you can migrate tape volumes to system management. See Chapter 12, “Managing Tape Volumes,” on page 219 for more information about setting up system-managed tape libraries.
- For information about using DFSMS FIT to create and implement a DFSMS design tailored to your DASD or tape environment and requirements, see “Using DFSMS FIT to Implement System-Managed Storage” on page 19.

---

## Using Storage Class to Manage Performance and Availability

Storage class defines the response time objectives and availability requirements for data. The storage class attributes are used to determine where the data is allocated and the level of service it receives when the data set is in use. Each system-managed data set must have a storage class assigned.

Storage class definitions help you develop a hierarchy of performance and availability service levels for data sets and help you automate performance and availability management. SMS attempts to select the hardware services that best meet the performance and availability objectives you specify for the data. Be careful how you specify the attributes because they can affect how volumes are selected for allocation. Make sure you assign the proper storage class in your ACS

routines. When you have a mixture of devices within and between the ACS-selected storage groups, an improper balance of storage class assignments can result in device over-utilization. For example, if you assign the same storage class to all data sets, SMS does not select other devices until the closest devices are filled to high threshold.

With the introduction of control units with large caches, sophisticated caching algorithms, large bandwidths, and such features as the IBM ESS parallel access volume and multiple allegiance, you no longer have to use the storage class performance values. But you can still use these values if you want to influence system-managed buffering for VSAM data sets or require sequential data striping for performance critical data sets.

The ESS allows for concurrent data transfer operations to or from the same volume on the same system. A volume used in this way is called a Parallel Access Volume (PAV). If you are using ESS devices, you can define DFSMS storage classes with the parallel access volume (PAV) option enabled. If the data set being allocated is assigned to this new or modified Storage Class, then the outcome of the volume selection process will be influenced by the way in which the PAV option was specified. This is described in more detail later.

Design your storage classes early and use the RACF facility class to authorize users access to such items as the VTOC or VTOC index. You can also use job accounting, or RACF user or group information available to your ACS routines to identify users that require specialized services. For example, you can use the &JOB, &PGM, and &USER read-only variables to distinguish Distributed FileManager/MVS data set creation requests. If you do not provide a storage class for Distributed FileManager/MVS data sets, only limited attribute support is available, affecting performance and function. Distributed FileManager/MVS rejects file creation requests that do not result in system-managed data sets.

The following objectives can help you identify the storage hardware services you require and the storage classes that you need to design:

- Improve performance for directly-accessed data sets
- Improve performance for sequentially-accessed data sets
- Improve data set backup performance
- Improve data set availability
- Place critical data sets on specific volumes
- Preallocate space for multivolume data sets

## **Using Cache to Improve Performance for Directly-Accessed Data Sets**

Dynamic cache management can be used to improve performance for data sets that are primarily accessed directly.

To use enhanced dynamic cache management, you need cache-capable 3990 storage controls with the extended platform.

If you use dynamic cache management in a storage environment that includes cache-capable 3990 storage controls, you can establish a performance hierarchy for data sets that are primarily accessed using direct access methods.

System-managed data sets can assume the following three states with 3990 cache and DASD fast write services:



- *Must-cache* data sets if the data set's storage class performance objective recorded as the direct millisecond response or sequential millisecond response demands cache facilities.
- *Never-cache* data sets if the data set's direct millisecond response or sequential millisecond response is specified as 999.
- *May-cache* data sets if direct millisecond response and sequential millisecond response values do not require the use of cache. These data sets can use cache facilities only if the must-cache data sets do not fully use cache and non-volatile storage required for DASD fast write.

The enhanced dynamic cache management of DFSMS ensures that must-cache data sets have a priority on 3990 cache and DASD fast write services, and that the may-cache data sets that benefit most from cache and DASD fast write receive these specialized performance services. You can get the best performance by assigning most data to the may-cache category. The enhanced dynamic cache management then supports performance management automation, but lets you designate selected data as must-cache data sets.

The system selects cache candidates so that the components of a cache-capable 3990 storage control are not over-committed. Data set level statistics are recorded in SMF so you can monitor the effect of cache services on I/O performance. *MVS/ESA SML: Managing Data* describes the data that is collected.

The cacheability of data sets also depends on the applications. Some applications could access the same data several times, while others (for example, sequential access) might not.

## Using Enhanced Dynamic Cache Management

The dynamic cache management enhancement monitors reads and writes to determine the cacheability of a may-cache data set. It also keeps a write hit ratio to ensure the availability of DASD fast write services.

When first opened, may-cache data sets are cached; DFSMS calculates their hit ratios to determine whether the data sets are good cache candidates. It does this by comparing the hit ratios to a specific threshold. If the total hit ratio is less than the read threshold, reads are inhibited for the data set. If the write hit ratio is less than the write threshold, DASD fast write is inhibited for the data set.

After a specified number of I/O operations, the data set is again eligible for caching and fast write, and is evaluated again.

## Requesting Dynamic Cache Management

You can use storage class attributes to control the selection of volumes supported by storage controls that use dynamic cache management. Once the data set is allocated on a volume, storage class attributes determine how the services of the storage control are used to satisfy the data set's performance requirements.

The bias attributes, Sequential and Direct, interact with the Millisecond Response (MSR) attributes, Sequential and Direct, to determine if a data set requires the services of a cache-capable storage control. If the Direct or Sequential Millisecond Response attribute's value can only be satisfied by a cache-capable storage control, the Direct and Sequential bias attributes are evaluated to see if the data set is primarily read (R) or written (W). SMS attempts to allocate data sets on the device that most closely matches the MSR and BIAS that you choose.



If the data set is allocated on a cache-capable 3990 storage control, dynamic cache management handles must-cache and may-cache data sets differently, based on the use of the 3990 resources.

Figure 8 shows the ISMF panel that describes storage performance requirements for data sets that must be cached with the DASD fast write services of a cache-capable 3990 storage control.

```

Panel Utilities Scroll Help
-----
                                STORAGE CLASS DEFINE                                Page 1 of 2
Command ==>>

SCDS Name . . . . . : USER6.TEST.SCDS
Storage Class Name  : SC1
To DEFINE Storage Class, Specify:
  Description ==>
    ==>
Performance Objectives
  Direct Millisecond Response . . . . . (1 to 999 or blank)
  Direct Bias . . . . . (R, W or blank)
  Sequential Millisecond Response . . . . . (1 to 999 or blank)
  Sequential Bias . . . . . (R, W or blank)
  Initial Access Response Seconds . . . . . (0 to 9999 or blank)
  Sustained Data Rate (MB/sec) . . . . . (0 to 999 or blank)
  OAM Sublevel . . . . . (1, 2 or blank)
  Availability . . . . . N (C, P ,S or N)
  Accessibility . . . . . N (C, P ,S or N)
  Backup . . . . . (Y, N or Blank)
  Versioning . . . . . (Y, N or Blank)
Use ENTER to Perform Verification; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 8. Storage Class Define Panel, Page 1 of 2

## Improving Performance for Sequential Data Sets

DFSMS supports improved performance for large, physical sequential data sets accessed using QSAM or BSAM access methods.

Data sets that are accessed sequentially can benefit from dynamic cache management; however, improved performance can be more effectively realized through the use of larger block and buffer sizes and parallel I/O processing.

*Sequential data striping* can be used for physical sequential data sets that cause I/O bottlenecks for critical applications. Sequential data striping uses extended-format sequential data sets that SMS can allocate over multiple volumes, preferably on different channel paths and control units, to improve performance. These data sets must reside on volumes that are attached to IBM 9340 or RAMAC Array Subsystems, to IBM 3990 Storage Subsystems with the extended platform, or ESS.

Sequential data striping can reduce the processing time required for long-running batch jobs that process large, physical sequential data sets. Smaller sequential data sets can also benefit because of DFSMS's improved buffer management for QSAM and BSAM access methods for striped extended-format sequential data sets. Chapter 8, "Managing Batch Data," on page 135 describes how sequential data striping can be used in the batch environment.

Evaluate buffer usage in assembler language BSAM or QSAM programs to ensure that there are sufficient buffers to support the extended sequential data sets. A macro, DCBE, lets you specify options for enhanced QSAM, BSAM, and BPAM.

For BSAM, the MULTACC option of DCBE lets BSAM I/O requests be run more efficiently. You can use MULTSDN to calculate a system-determined number of channel programs (NCP). If NCP is omitted, MULTSDN computes a value for NCP by multiplying the number of stripes by the number of blocks that can be stored on a track. As long as enough buffers are available, I/O is scheduled on all the stripes to provide increased performance.

When you update assembler programs to take advantage of improved buffer handling, consider taking advantage of virtual storage constraint relief with BSAM and QSAM support for 31-bit execution mode. Assembler programs using QSAM should specify the RMODE31=BUFF option on the DCBE macro and be recompiled to execute in 31-bit addressing mode. Programs using BSAM should allocate data areas above the 31-bit addressing line and be recompiled to execute in 31-bit addressing mode.

**Recommendation:** Use the logical backup and restore techniques for striped data sets having more than one stripe. These multi-part data sets can only be restored from physical backup copies if you enter an individual restore command for each part.

The benefit from sequential data striping must be evaluated in relationship to your ESCON<sup>®</sup> cache-capable 3990 storage control configuration. For each serially-attached cache-capable 3990 storage control in a storage group, up to four paths are available for concurrent I/O operations. Consequently, four stripes at most can be effectively used per storage control. Newer control units support more than four paths.

**Related Reading:** For more information about the DCBE macro, see *z/OS DFSMS Macro Instructions for Data Sets*.

## Using Sequential Data Striping

You can write striped extended-format sequential data sets with the maximum physical block size for the data set plus control information required by the access method. The access method writes data on the first volume selected until a track is filled. The next physical blocks are written on the second volume selected until a track is filled, continuing until all volumes selected have been used or no more data exists. Data is written again to selected volumes in this way until the data set has been created. A maximum of 59 stripes can be allocated for a data set. “Selecting Volumes for Striped Extended-Format Data Sets” on page 43 describes how the system selects the volumes used to allocate striped extended-format sequential data sets.

For striped data sets, the maximum number of extents on a volume is 123.

## Requesting Sequential Data Striping

To create a striped extended-format sequential data set, specify the data class attribute, Data Set Name Type, as EXTENDED. You can require use of sequential data striping by setting Data Set Name Type to (EXTENDED,R). However, if striping is not possible, allocation fails.

You can request sequential data striping by setting Data Set Name Type to (EXTENDED,P). If striping is not possible, the data set is allocated as non-striped. SMS determines the number of volumes to use for a striped data set based on the value of the Sustained Data Rate in the storage class. Sustained Data Rate is the data transfer rate that DFSMSdfp should keep up during a period of typical I/O activity for the data set.

You cannot request sequential data striping with the JCL parameter, DSNTYPE. You need to use a data class with appropriate attributes.

## Improving Performance for VSAM Data Sets

VSAM data striping allows sequential I/O to be performed for a data set at a rate greater than that allowed by the physical path between the DASD and the processor. The physical characteristics of channels, control units, and DASD limit the data transfer rate. VSAM data striping avoids such limitations by spreading the data set among multiple stripes on multiple control units. The data striping function is designed to improve the performance of applications requiring sequential access to data records. Data striping does not affect direct access to data.

### Using VSAM Data Striping

An equal amount of space is allocated for each stripe. For a data set with the non-guaranteed space attribute of a storage class, the initial allocation quantity is divided across all volumes in the stripe count. If the guaranteed space attribute is used, the specified quantity is allocated to each volume in the stripe count.

**Restrictions:** The RESET/REUSE option is not supported for VSAM data striping. The restrictions for data in the striped format are the same as for other VSAM data sets in the extended format (EF). The KEYRANGE and IMBED attributes are not supported for any VSAM data set types.

VSAM-striped data sets can be extended on the same volume, equivalent to the existing data striping for SAM data sets, or to a new volume, which is not supported for SAM data sets. The ability to extend a stripe, or stripes, to a new volume is called *multi-layering*.

### Requesting VSAM Data Striping

Striped VSAM data sets are in extended format (EF) and internally organized so that control intervals (CIs) are distributed across a group of DASD volumes or stripes. A CI is contained within a stripe.

VSAM striping is used only for the data component of the base cluster of a VSAM data set. It is effective for sequential processing when the data set is processed for non-shared resources (NSR). The following conditions must be met for VSAM data striping:

- The data set must be system-managed.
- The data set must be in the extended format.
- The stripe count must be greater than one.

The storage class SDR value is greater than the minimum for a device type: 4 MB per second for 3390 and 3 MB per second for 3380, when the request is for non-guaranteed space.

**Definition:** *Single-striped data sets* refers to data sets that are in extended format but are not striped under the above conditions. They are, therefore, considered non-striped.

### VSAM System-Managed Buffering

System-managed buffering (SMB) supports batch application processing and processing of data sets with associated alternate indexes (AIX<sup>®</sup>). You can use any of the following four SMB processing techniques:

#### Direct optimized (DO)

Choose this technique for applications that access records in a data set in totally random order.

**Sequential optimized (SO)**

Choose this technique for backup and for applications that read the entire data set from the first to the last record or read a large percentage of a data set in totally sequential order.

**Direct weighted (DW)**

Choose this technique for applications that access records in a data set mostly in random order.

**Sequential weighted (SW)**

Choose this technique for applications that access the entire data set mostly in sequential order.

**Related Reading:** For more information about Implementation techniques and SMB, see *z/OS DFSMS Using Data Sets*.

## Improving Performance with Hiperbatch™

I/O buffers are not shared between batch jobs. When multiple jobs concurrently read the same data set, each job reads the data into its own buffers. You can use the Hiperbatch facility to eliminate a separate I/O for each job when multiple jobs accessing the same data are scheduled to run concurrently on the same processor.

For selected data sets, the shared buffer is established on the first open of the data set by any of the jobs. The access methods use the Data Look-aside Facility to cache the data in the shared buffer. For example, the shared buffer is invalidated when another job opens the data set for output.

The candidate data sets for Hiperbatch are defined to the system. For example, physical sequential data sets accessed using QSAM and VSAM ESDS, RRDS, VRRDS, and KSDS with a control interval size of a multiple of 4096 bytes are eligible for Hiperbatch.

**Related Reading:** For more information about Hiperbatch, see the MVS Hiperbatch Guide on the z/OS Internet Library (<http://www.ibm.com/systems/z/os/zos/bkserv/>)

## Improving Performance with the Parallel Access Volume Option

The Enterprise Storage Server (ESS) allows for concurrent data transfer operations to or from the same volume on the same system. A volume used in this way is called a Parallel Access Volume (PAV). With ESS, you can define alias device numbers to represent one physical device which allows for multiple I/O operations to be started at one time which improves performance.

The Storage Class option for Parallel Access Volumes may be used to influence volume selection in such a way that data sets that require high performance may be directed towards volumes that are being used as Parallel Access Volumes. The DFSMS volume selection process puts eligible volumes into the primary, secondary, or tertiary category. For more information about the volume selection process, see "Selecting Volumes with SMS" on page 42.

The DFSMS PAV capability option includes the following settings based on the volume selection categories:

**Required**

Only volumes with the PAV feature enabled are selected.

**Preferred**

Volumes with the PAV feature enabled are eligible to be primary volumes. Volumes without the PAV feature enabled are only eligible to be secondary volumes.

**Standard**

Volumes without the PAV feature enabled are preferred over volumes with the PAV feature enabled and are eligible to be primary volumes. Volumes with the PAV feature enabled are only eligible to be secondary volumes.

**Nopreference**

Whether the PAV feature is enabled or not for a volume is ignored and has no effect on the volume selection process. This is the default value for this option.

## Improving Availability

The following options can help you improve availability:

**Enterprise Storage Server (ESS)**

Provides such copy services as FlashCopy, extended remote copy (XRC), suspend/resume for unplanned outages, and peer-to-peer remote copy (PPRC). For detailed descriptions of these copy services, see *z/OS DFSMS Advanced Copy Services*.

**Data set separation**

Used to keep designated groups of data set separate, on either the physical control unit (PCU) or volume level, from all the other data sets in the same group. This reduces the effect of single points of failure. For information on how to use data set separation, see *Using Data Set Separation in z/OS DFSMSdfp Storage Administration*.

**Storage class availability attribute**

Used to assign a data set to a fault-tolerant device. Such devices ensure continuous availability for a data set in the event of a single device failure. The fault-tolerant devices that are currently available are dual copy devices and RAID architecture devices, such as RAMAC or ESS.

The following options are available for the availability attribute. To ensure that SMS allocates a data set on a fault-tolerant device, assign the data set a storage class that specifies the Availability attribute as CONTINUOUS.

**CONTINUOUS**

Data is placed on a dual copy or RAID device so that it can be accessed in the event of a single device failure. If neither of the devices is available, allocation fails. Dual copy, RAMAC, or ESS volumes are eligible for this setting.

**PREFERRED**

The system tries, but does not guarantee, to place data on a fault-tolerant RAID device. Dual copy volumes are not candidates for selection.

**STANDARD**

This represents normal storage needs. The system tries to allocate the data set on a non-fault-tolerant device to avoid wasting resources. In this case, processing of a data set stops in the event of a device failure. All except dual copy devices are eligible.

## NOPREF

The system chooses any device, except for dual copy devices. NOPREF is the default.

## Improving Availability during Data Set Backup

Data sets that are consistently in use, such as DFSMSHsm control data sets, databases, and libraries, require specialized facilities to ensure that data set backups are nondisruptive and preserve data set integrity.

Management class attributes let you choose how DFSMSHsm and DFSMSdss process data sets that are in use during the backup. Point-in-time capabilities, using either concurrent copy on the 3990-6, or virtual concurrent copy on the RAMAC Virtual Array, let you use the following backup capabilities:

- Use DFSMSdss to create a *point of consistency* backup of CICS/VSAM, IMS, or DB2 databases without needing to quiesce them during the entire backup process.
- Use DFSMSdss to create backups of data sets without requiring serialization during the entire backup process.

DFSMSdss serializes the data during the concurrent copy initialization period (the time between the start of DFSMSdss and the issuing of the ADR734I message).

- Create and maintain multiple backup versions of DFSMSHsm control data sets, while increasing the availability of DFSMSHsm functions, such as recall.
- Use the backup-while-open capability for CICS VSAM data sets, with DFSMSdss in batch mode or with automated DFSMSHsm, to provide backups with data integrity even when the data sets are being updated. Data integrity is assured for VSAM KSDSs even when CICS access results in control interval or control area splits or data set extends.

## Creating Point-in-Time Copies

You can use either concurrent copy, virtual concurrent copy, or FlashCopy to make point-in-time copies that can be used for backup operations. DFSMSdss uses concurrent copy and virtual concurrent copy for data sets that span a 3990 Model 6 Storage Control, RAMAC Virtual Array subsystem, or ESS.

With concurrent copy, DFSMSdss works with a cache-capable 3990 storage control and SMS to begin and sustain concurrent copy sessions. DFSMSdss determines a list of physical extents by volume that are associated with each session. For each backup session, the storage control ensures that the original track images are preserved in the cache, while writing any updated track images to DASD. Each cache-capable 3990 storage control can sustain up to 64 concurrent copy sessions simultaneously.

When virtual concurrent copy is being used for backup, DFSMSdss uses the SnapShot feature of the RAMAC Virtual Array to create an interim point-in-time copy of the data to be backed up. Once the point-in-time copy is created, serialization is released and the concurrent copy session is logically complete. DFSMSdss then performs I/O from the interim point-in-time copy to create the backup. Once this is done, the backup is physically complete and the job ends. Similar operations are followed for ESS.

You specify whether concurrent copy should be used when backing up data sets or volumes using DFSMSdss. A point-in-time session is established when DFSMSdss is called for volumes or data sets allocated behind a storage subsystem that



supports concurrent copy or virtual concurrent copy. The data set's management class controls concurrent copy usage. To use virtual concurrent copy, you must have an ESS or the SnapShot feature enabled on the RAMAC Virtual Array and the DFSMSdss concurrent copy SnapShot SPE installed.

With DFSMSdss SnapShot copy support and the RAMAC Virtual Array, you can make almost instantaneous copies of data. Once the copy is complete, both the source and target data sets or volumes are available for update.

### **Requesting Point-in-Time Copies for Databases**

The backup service automated by the management class using the point-in-time copy is available with DFSMSHsm for CICS VSAM data sets. IMS and DB2 production data sets are normally backed up with the database image copy utilities. These utilities allow the transactions to be quiesced and the current data in the in-storage buffers to be written out. They then invoke DFSMSdss to back up the databases with the concurrent copy service. DFSMSdss uses the concurrent copy, virtual concurrent copy, or SnapShot service based on the data residency location.

### **Requesting Point-in-Time Copy Support**

You can assign system-managed data sets a storage class with the Accessibility attribute set to CONTINUOUS or PREFERRED:

#### **CONTINUOUS**

If data sets cannot be allocated on copy-capable volumes, allocation fails.

#### **PREFERRED**

SMS attempts to allocate data sets behind a cache-capable 3990 storage control with point-in-time support, IBM RAMAC Virtual Array device, or ESS. If no space is available, data sets are allocated on a volume that is not supported by point-in-time copy.

#### **STANDARD**

The data sets should be allocated on volumes that are not supported by point-in-time copy. If this cannot be done, a data set can be allocated to a volume supported by point-in-time copy.

#### **NOPREF**

The data sets should be allocated to volumes whether the volumes support point-in-time copy or not.

If you specify CONTINUOUS or CONTINUOUS PREFERRED to request point-in-time copy, you must then identify which kind of device you want SMS to use: *versioning* or *backup* devices.

#### **Versioning Device**

creates a "fast" point-in-time version of a data set, which is then available for application testing, reporting, or backup operations. While the version is being made, the data set is unavailable for normal application processing for a minimal period of time. Versioning is done using the SnapShot feature of the RAMAC Virtual Array or the FlashCopy service of the ESS.

#### **Backup Device**

creates a "fast" point-in-time backup copy of a data set. While the backup copy is being made, the data set is unavailable for normal application processing for a minimal period of time. Two methods are supported:

##### **Method 1**

Establish a concurrent copy session with the 3990 DASD controller and make the backup copy.

## Method 2

Take a point-in-time version, using the virtual concurrent copy service of the IBM ESS or the SnapShot feature of the RAMAC Virtual Array, to create a point-in-time copy and the backup to a specified target device.

**Related Reading:** For specific recommendations on how to request point-in-time copy, see *z/OS DFSMSdfp Storage Administration*.

## Preallocating Space for Multivolume Data Sets

The storage class Guaranteed Space attribute lets you preallocate space for multivolume data sets. It supports striped extended-format sequential or VSAM, standard VSAM, and standard physical sequential data sets.

### Using Guaranteed Space for Preallocation

For VSAM, space allocation using guaranteed space works as follows:

1. SMS allocates the primary space requested for the multivolume data set on each volume requested. The volume that is selected first is the primary volume, and the other volumes are candidate volumes.
2. Space is used on the primary extents first. After it is full, secondary extents are created on this volume as required.
3. If these are insufficient, the preallocated space on the next volume is used.
4. Secondary extents on this volume are created as required before using the preallocated space on the next volume.

For non-VSAM data sets, secondary extents are allocated only on the last volume of the multivolume data sets. All volumes except the last one will have only primary extents.

### Requesting Preallocation Using the Guaranteed Space Attribute

Select guaranteed space by setting the storage class Guaranteed Space attribute to YES. SMS uses the supplied volume count when allocating the data set.

With the IBM ESS, the Guaranteed Space attribute of a storage class with specific volumes is no longer required for data sets other than those that need to be separated, such as the DB2 online logs and BSDS, or those that must reside on specific volumes because of their naming convention, such as the VSAM RLS sharing control data sets. The ESS storage controllers use the RAID architecture that enables multiple logical volumes to be mapped on a single physical RAID group. If required, you can still separate data sets on a physical controller boundary for availability beyond what is inherently built into the RAID architecture.

The ESS is also capable of parallel access volumes (PAV) and multiple allegiance. These ESS capabilities, along with its bandwidth and caching algorithms, make it unnecessary to separate data sets from each other for the purpose of performance. Traditionally, IBM storage subsystems allow only one channel program to be active on a disk volume at a time. This means that after the subsystem accepts an I/O request for a particular unit address, this unit address appears "busy" to subsequent I/O requests. This ensures that additional requesting channel programs cannot alter data that is already being accessed. By contrast, the ESS is capable of multiple allegiance, or concurrent execution of multiple requests from multiple hosts. That is, the ESS can queue and concurrently execute multiple requests for the same unit address from multiple hosts, provided that no extent conflict occurs.



In addition, the ESS enables PAV or multiple concurrent accesses to a single volume from a single host. To access a volume concurrently, you must associate multiple device numbers with a single volume. The ESS provides this capability by allowing you to define a PAV-base address and one or more PAV-alias addresses. It allows up to 255 aliases per logical volume. Therefore, you no longer have to separate data sets from each other for performance reasons.

If you specify NO for Guaranteed Space, then SMS chooses the volumes for allocation, ignoring any VOL=SER statements specified on JCL. Primary space on the first volume is preallocated. NO is the default.

Specifying volsers with the Guaranteed Space attribute of the storage class is strongly discouraged. If used, the following considerations must apply:

- Ensure that the user is authorized to the storage class with the Guaranteed Space attribute.
- Write a storage group ACS routine that assigns a storage group that contains the volumes explicitly specified by the user.
- Ensure that all volumes explicitly specified by the user belong to the same storage group, by directing an allocation that is assigned a Guaranteed Space storage class to all the storage groups in the installation.
- Ensure that the requested space is available because there is no capability in SMS to allow specific volume requests except with the Guaranteed Space attribute.
- Ensure that the availability and accessibility specifications in the storage class can be met by the specified volumes.

### **Extending Data Sets on New Volumes**

When the data set cannot be extended on the current volume and there is no candidate volume in the catalog, you can add new candidate volumes by using the access method services ALTER ADDVOLUME command and specifying an asterisk for the volume serial number. This is not effective on the current allocation of the data set, only on the next allocation. This means that a data set that is currently being accessed needs to be unallocated and then reallocated and reaccessed to extend to the new volume.

This procedure is done automatically by the DB2 subsystem for table spaces allocated using DB2 STOGROUPS.

---

## **Managing Space and Availability for Data Sets**

Management class defines the space and availability requirements for data sets. Its attributes control backup, migration, retention of data, and release of unused space. DFSMSHsm uses information from the storage groups to determine what automatic management processes to run for the data sets on the volumes in the storage group. DFSMSHsm manages data sets during the daily availability, primary and secondary space management, and interval migration processes, using the management classes assigned to your data sets from the active SMS configuration.

Assign all your system-managed permanent data sets to management classes, even if their management class attributes specify that no space or availability services are required for the data sets. If a management class is not assigned, DFSMSHsm uses default management class attributes from your SMS configuration, or, if no default management class exists, it uses DFSMSHsm defaults.

**Tip:** You can prevent DFSMSHsm processing at the storage group level.

Data sets with varying management requirements coexist on the same volume. However, you might want to separate certain types of data sets with similar management requirements in their own storage group. An example is the production database data placed in a database storage group. You can use the image copy utilities of DB2 and IMS databases for backup and recovery. Because of the customized procedures required to back up and restore this data, you can separate it from data that uses DFSMSHsm facilities.

Figure 9 through Figure 11 on page 35 show the ISMF management class panel definitions required to define the STANDARD management class.

```

Panel  Utilities  Scroll  Help
-----
                                MANAGEMENT CLASS DEFINE                                Page 1 of 6
Command ==>

SCDS Name . . . . . : YOUR.OWN.SCDS
Management Class Name : STANDARD

To DEFINE Management Class, Specify:

Description ==> This is the management class assigned
                ==> to average data sets in the installation.

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

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

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 9. Management Class Define Panel, Page 1 of 6

The expiration and retention attributes for the STANDARD management class specify that no expiration date has been set in the management class. These data sets are never deleted by DFSMSHsm unless they have explicit expiration dates.

```

Panel Utilities Scroll Help
-----
                                MANAGEMENT CLASS DEFINE                                Page 2 of 6
Command ==>

SCDS Name . . . . . : YOUR.OWN.SCDS
Management Class Name : STANDARD

To DEFINE Management Class, Specify:

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

Migration Attributes
Primary Days Non-usage . . . . 15        (0 to 9999 or blank)
Level 1 Days Non-usage . . . . 30        (0 to 9999, NOLIMIT or blank)
Command or Auto Migrate . . . . BOTH     (BOTH, COMMAND or NONE)

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

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 10. Management Class Define Panel, Page 2 of 6

For single-volume data sets, DFSMSHsm releases any unused space when you specify partial release. Also, if the data set is not referenced within 15 days, it moves to migration level 1, and, after 15 more days, moves to migration level 2.

For all VSAM data sets allocated in the extended format and accessed using the VSAM access method, you can use the Partial Release attribute of the management class to release allocated but unused space. The system releases the space either immediately during close processing, or during DFSMSHsm space management cycle processing. This is similar to how the system processes non-VSAM data sets.

```

Panel Utilities Scroll Help
-----
                                MANAGEMENT CLASS DEFINE                                Page 3 of 6
Command ==>

SCDS Name . . . . . : YOUR.OWN.SCDS
Management Class Name : STANDARD

To DEFINE Management Class, Specify:
Backup Attributes
Backup Frequency . . . . . 1              (0 to 9999 or blank)
Number of Backup Vers . . . . . 2        (1 to 100 or blank)
(Data Set Exists)
Number of Backup Vers . . . . . 1        (0 to 100 or blank)
(Data Set Deleted)
Retain days only Backup Ver . . . 60     (1 to 9999, NOLIMIT or blank)
(Data Set Deleted)
Retain days extra Backup Vers . . 30     (1 to 9999, NOLIMIT or blank)
Admin or User command Backup . . BOTH   (BOTH, ADMIN or NONE)
Auto Backup . . . . . Y                  (Y or N)
Backup Copy Technique . . . . . S        (P, R, S, VP, VR, CP or CR)

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; Cancel to Exit.

```

Figure 11. Management Class Define Panel, Page 3 of 6

DFSMSHsm backs up the data set daily if the data set has been changed. The last two versions of the data set are retained as long as the data set exists on primary

storage or migration level 1. If the data set is deleted, only one backup copy is retained. DFSMSHsm does not use the high-availability backup technique, concurrent copy, for this data set.

## Managing Data with DFSMSHsm

DFSMSHsm uses information from your SMS configuration to manage system-managed volumes. DFSMSHsm uses information from the SYS1.PARMLIB member ARCCMDxx to manage non-system-managed volumes. Specifically, DFSMSHsm performs space and availability services using the following controls:

- When DFSMSHsm is processing system-managed volumes, data sets are treated individually. Storage management attributes defined in the management class and storage group parameters control DFSMSHsm processing.
- When DFSMSHsm is processing non-system-managed volumes, all the data sets on the volume are treated as one group. DFSMSHsm uses parameters that are global to the DFSMSHsm subsystem or to a non-system-managed volume to describe how DFSMSHsm should manage data at the volume level.

During your implementation of system-managed storage, you can use DFSMSHsm to manage both non-system-managed volumes and system-managed volumes concurrently.

### Relating ADDVOL to Storage Group Attributes

The ADDVOL command defines non-system-managed volumes that DFSMSHsm manages or owns. ADDVOL parameters specify management options for volumes that differ from the global options. You must supply one ADDVOL command for each non-system-managed volume that you want DFSMSHsm to manage or own.

ADDVOL commands do not define system-managed volumes to DFSMSHsm. Instead, DFSMSHsm uses the contents of the storage groups defined in your active SMS configuration to determine the list of volumes to process. If the Auto Migrate and Auto Backup storage group attributes specify that automatic space management and backup services should be performed for the storage group, all data sets on these volumes are candidates for migration and backup.

DFSMSHsm uses the Auto Dump attribute to determine the storage groups containing volumes that should be dumped with DFSMSdss full volume dump. The Guaranteed Backup Frequency storage group attribute allows you to assign a maximum period of elapse time before a data set is backed up regardless of its change status.

The High and Low Allocation/Migration Threshold attributes control DFSMSHsm space management services. These thresholds apply to all the volumes in the storage group. DFSMSHsm space management consists of the following three processes:

- Primary space management  
This includes deleting, expiring, and releasing unused space, and migrating from primary storage to migration levels 1 and 2. The DFSMSHsm primary space management cycle runs daily and uses the low migration threshold to determine when to stop processing data. If the amount of allocated space exceeds the low migration threshold, DFSMSHsm processes data sets until the threshold is met.
- Secondary space management  
This includes deleting expired migrated data sets and MCDS records, and migrating data sets from level 1 to level 2.
- Interval migration

This includes deleting, expiring, and releasing unused space, and migrating data sets from primary storage to migration levels 1 and 2. DFSMShsm initiates it on an as-needed basis.

You can specify interval migration, DFSMShsm's hourly space management service, at a storage group level. Every hour, DFSMShsm determines if each volume in each storage group has more data than the *interval threshold*, and if so, processes the data on the volume in an attempt to achieve low migration threshold. For all storage groups defined with Auto Migrate=I, the interval threshold is the midway point between the high allocation threshold and the low migration threshold. For storage groups defined with Auto Migrate=Y, the interval threshold is simply the high allocation threshold.

Interval migration using Auto Migrate=I is especially useful for tape mount management storage groups which tend to fill up several times a day. By using the interval threshold instead of the high allocation threshold to trigger migration, DFSMShsm can better keep up with the demand.

### Volume Selection During Recall or Recover

SMS, rather than DFSMShsm, determines the selection of primary storage volumes for data sets recalled or recovered by DFSMShsm to system-managed storage.

For non-system-managed data sets, DFSMShsm returns a migrated data set to a DFSMShsm-managed volume having the most free space during recall.

### Using copy pools

A copy pool is a defined set of pool storage groups that contains data that DFSMShsm can backup and recover collectively, using volume-level fast replication. You can use a copy pool to specify the pool storage groups that you want DFSMShsm to process for fast replication. For more information about how to define copy pools, see *z/OS DFSMSdfp Storage Administration*.

## Using SMS with DFSMShsm Commands

For a detailed explanation of each of the DFSMShsm commands, including which commands are pertinent to a system-managed storage environment, see the DFSMShsm section of *z/OS DFSMSdfp Storage Administration*.

Some DFSMShsm commands or command parameters do not apply to system-managed data sets or volumes. Table 2 lists the DFSMShsm commands that are obsolete when processing system-managed volumes or data sets. It shows the SMS class or group containing the relevant information.

Table 2. DFSMShsm Commands Replaced by SMS

DFSMShsm Command	SMS Class or Group
ADDVOL	storage group
ALTERDS	management class
DEFINE POOL	storage group
DEFINE VOLUMEPOOL	storage group
SETMIG VOL	management class or storage class

The following DFSMShsm commands are affected by SMS:

#### RECALL

Recalls a migrated data set. You can specify the volume and type of unit to which DFSMShsm is to recall the data set, making the volume and unit

type available to your ACS routines. You can use this information to determine where to place the data set, but it is recommended that you let SMS place the data set based on the current requirements.

The FORCENONSMS parameter lets authorized users inhibit SMS allocation services and allocate data sets on non-system-managed volumes. Data sets that can only be allocated as SMS data sets, such as striped and extended data sets, are excluded.

## RECOVER

Causes DFSMShsm to restore data to a data set or volume. You can specify the volume and type of unit to which DFSMShsm is to restore the data. If you do, the volume and unit type are passed to your ACS routines. You can use this information to determine where to place the data set, but it is recommended that you let SMS place the data set based on the current requirements.

FROMVOLUME is an optional parameter indicating that the data set was uncataloged and resided on the volume specified by the volume serial number when DFSMShsm created the backup version. This parameter is not applicable to system-managed volumes.

When replacing an existing non-system-managed data set, DFSMShsm normally recovers or restores the data set to the volume on which the data set is cataloged. However, for system-managed data sets, SMS selects the target volume. In addition, the SMS management class attributes for the data set are the ones currently defined in the management class in the active SMS configuration, not the ones in effect at data set backup or dump time.

When recreating a deleted data set, SMS determines new management class and storage class attributes by calling the management class and storage class ACS routines. Your ACS routines determine if the data set should be system-managed and where to place the data set.

The FORCENONSMS parameter lets you override SMS allocation and recreate the data set as non-system-managed. When this parameter is issued with the RECOVER command, it causes the data set to be recovered to a non-system-managed volume. Data sets that can only be allocated as SMS data sets, such as striped and extended data sets, are excluded.

## SETSYS

Establishes or changes the values of DFSMShsm control parameters. Many SETSYS command parameters do not apply to system-managed data sets. They have been replaced by management class attributes that let you tailor storage management services to the data set level without considering the requirements of other data sets on the volume. The major changes are summarized in Table 3.

*Table 3. Management Class Attribute Equivalents of DFSMShsm Parameters*

DFSMShsm Parameter	Management Class Attribute Equivalent
DAYS	Primary Days Non-usage
MIGRATIONLEVEL1DAYS	Level 1 Days Non-usage
DELETEBYAGE(days)	Expire after Days Non-usage
n/a	Partial Release
VERSIONS	Number of Backup Versions (Data Set Exists)
FREQUENCY	Backup Frequency

Table 3. Management Class Attribute Equivalents of DFSMShsm Parameters (continued)

DFSMShsm Parameter	Management Class Attribute Equivalent
n/a	Admin or User command Backup
n/a	# GDG Elements on Primary
n/a	Rolled-off GDS Action
n/a	Number of Backup Versions (Data Set Deleted)
n/a	Retain days only Backup Version (Data Set Deleted)
n/a	Retain days extra Backup Version
SCRATCHFREQUENCY	Expire after Date/Days
AUTOBACKUP	Auto Backup

## Using SMS with Aggregate Backup and Recovery Support

Aggregate backup and recovery support (ABARS) supplements DFSMShsm availability management services. You can use it to perform the following tasks:

- Create portable aggregates of user-specified data sets
- Manage critical application backup and recovery
- Transfer workloads between sites

ABARS consists of the following three parts:

- Aggregate definition using the ISMF Define Aggregate Application  
This application lets you define data set selection criteria, comparable to DFSMSdss filters, that are used by ABARS to identify and copy the group (or aggregate) of related data sets to tape. You can define an instruction data set that documents recovery and operational procedures using this application. You can also specify other data sets, usually on tape, that should be included in the aggregate. These are known as *accompany* data sets.
- Aggregate backup using the DFSMShsm ABACKUP command  
A DFSMShsm command implemented to initiate backup of the aggregate group, based on the selection criteria that you defined in your SMS configuration. An activity log documents actions taken by the ABACKUP command that includes:
  - Data sets that could not be copied
  - Tape volume serial numbers that should be pulled by the tape librarian or ejected from the Tape Library Dataserver for transfer to the recovery site
 Specialized management class criteria control the retention of ABARS process output. You can use concurrent copy for data sets backed up using ABARS. The Abackup Copy Technique and Copy Serialization management class attributes control the use of concurrent copy. The Abackup Copy Technique attribute specifies whether concurrent copy is used for the data set. The Copy Serialization attribute lets you specify whether backup should proceed if the data set is in use when ABACKUP is started.
- Aggregate recovery using the DFSMShsm ARECOVER command  
A DFSMShsm command implemented to initiate recovery of the aggregate group at your recovery site. SMS does not have to be active at the site to recover most data sets. However, striped data sets and VSAM data sets allocated with JCL are not supported at the recovery site unless SMS is active.



**Related Reading:** For more information about ABARS, see *z/OS DFSMSShsm Storage Administration*.

## Managing DASD Volumes with SMS

Space management optimizes the use of physical space on storage devices. A storage group defines a collection of volumes that have similar management characteristics, although the volumes might have different performance levels.

### Pooling Volumes with Storage Groups

You probably use storage pools to contain data sets that are managed similarly. These pools are defined to z/OS as esoterics using the Hardware Configuration Definition (HCD) facility. You define the individual volumes to z/OS in the PARMLIB member, VATLSTxx. Esoterics and volume lists are difficult to maintain, because any changes to their definition require a system IPL. SMS preserves your ability to do volume pooling, increases flexibility in maintaining your DASD device configuration, and ensures that enough free space is available.

SMS uses storage groups to contain the definitions of volumes that are managed similarly. Each storage group has a high allocation and low migration threshold defined. SMS uses the high allocation threshold to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high allocation threshold are selected in favor over those volumes that contain more data than the high allocation threshold specifies. DFSMSShsm uses the low migration threshold during primary space management, and the interval threshold during interval migration to determine when to stop processing data.

Figure 12 shows how the PRIME90 sample storage group is defined.

```
Panel Utilities Scroll Help
-----
DGTDCSG2                POOL STORAGE GROUP DEFINE                Page 1 of 2
Command ==>>>

SCDS Name . . . . . : YOUR.OWN.SCDS
Storage Group Name  : PRIME90
To DEFINE Storage Group, Specify:
Description ==> Storage group for normal-sized data sets
                ==> to reside on 3390 devices.
Auto Migrate . . Y (Y, N, I or P)  Migrate Sys/Sys Group Name . .
Auto Backup . . Y (Y or N)         Backup Sys/Sys Group Name . .
Auto Dump . . . Y (Y or N)         Dump Sys/Sys Group Name . . .
Overflow . . . . N (Y or N)        Extend SG Name . . . . .
                                     Copy Pool Backup SG Name . . .

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

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

Use ENTER to Perform Selection; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 12. Pool Storage Group Define Panel (Page 1)



```

Panel Utilities Scroll Help
-----
DGTDCSGK          POOL STORAGE GROUP DEFINE          Page 2 of 2
Command ==>>

SCDS Name . . . . . : YOUR.OWN.SCDS
Storage Group Name : PRIME90

To DEFINE Storage Group, Specify:

Allocation/migration Threshold :   High . . 95 (1-99) Low . . 80 (0-99)
Alloc/Migr Threshold Track-Managed: High . . 85 (1-99) Low . . 1 (0-99)
Guaranteed Backup Frequency . . . . . 15 (1 to 9999 or NOLIMIT)
BreakPointValue . . . . . (0-65520 or blank)
Processing Priority . . . . . (1-100)

```

Figure 13. Pool Storage Group Define Panel (Page 2)

In Figure 12 on page 40, data sets on volumes in the PRIME90 storage group are automatically backed up and migrated according to their management class attributes. These volumes are also automatically dumped and one copy of each volume is stored offsite.

SMS tries not to allocate above high threshold, but might allocate a new data set in a storage group which is already at or above threshold if it cannot find another place to put the data. In PRIME90, interval migration is triggered at 50% of the difference between the high and low threshold values. As shown in Figure 12 on page 40, DFSMSShsm lets the volume fill to near 95%, but can trigger interval migration if the volume exceeds 88%, which is midway between the low (80%) and high (95%) thresholds specified on the panel. (For AM=Y storage groups, this requires SETSYS INTERVAL.)

For EAS-eligible data sets on volumes that support cylinder-managed space, the allocation threshold is divided into categories. All categories are assessed to determine the volumes capability of meeting the threshold requirements. These include the volumes capability in meeting the track-managed space or the cylinder-managed space thresholds and the total volume space threshold. Note that the Allocation Threshold percentage applies to both cylinder-managed space and total volume space.

These thresholds are further classified as primary or secondary thresholds:

- For space requests that are less than the break point value, the primary threshold is the track-managed space and the secondary threshold is the total volume space.
- For space requests that are equal to or greater than the break point value, the primary threshold is the cylinder-managed space and the secondary threshold is the total volume space.

Volume pools should not contain mixed device types (with different track geometries) because data set extensions to multiple volumes might result in problems. You can design the storage group ACS routine to direct allocation to up to 15 storage groups. You can thus preserve your existing volume pooling structure.

As an example, the starter set's SMS configuration has a PRIMARY storage group that has been subdivided into two storage groups, PRIME80 and PRIME90, because our storage configuration contained both 3380 and 3390 device types. We

balanced allocation to both the 3380 and 3390 devices by coding the following statement in the storage group ACS routine:

- ```
SET &STORGRP = 'PRIME80', 'PRIME90'
```

This statement results in all volumes in these two storage groups being considered equally for allocation. The system selects volumes based on their ability to satisfy the availability and performance criteria that you specify in the storage class that is assigned to the data set.

Extended address volume (EAV) and non-EAV volumes may reside in the same volume pools. SMS prefers EAV volumes for EAS-eligible data sets that are equal to or larger than the BPV. For non- EAS-eligible requests that are smaller than the BPV, SMS will not have a preference for EAV volumes.

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

**Related Reading:** For more details about the attribute fields that are displayed on the Pool Storage Group Define panel and other ISMF panels, see *z/OS DFSMSdftp Storage Administration*.

## Selecting Volumes with SMS

SMS determines which volumes are used for data set allocation by developing a list of all volumes from the storage groups that your storage group ACS routine has assigned. Volumes are then either removed from further consideration or flagged as primary, secondary, or tertiary volumes.

For nonmultistriped data sets, SMS classifies all volumes in the selected storage groups into the following four volume categories:

### Primary Volumes

Primary volumes are online, below threshold, and meet all the specified criteria in the storage class. Both the volume status and storage group status are enabled. Volume selection starts from this list.

For EAS-eligible data sets on devices with cylinder-managed space, both the track-managed space / cylinder-managed space threshold and the total volume space threshold will be assessed to determine if the volume gets placed on the primary volume list.

### Secondary Volumes

Secondary volumes do not meet all the criteria for primary volumes. SMS selects from the secondary volumes if no primary volumes are available.

### Tertiary Volumes

Volumes are classified as tertiary if the number of volumes in the storage

group is less than the number of volumes that are requested. SMS selects from the tertiary volumes if no secondary volumes are available.

### **Rejected Volumes**

Rejected volumes are those that do not meet the required specifications. They are not candidates for selection.

If allocation is not successful from the primary list, then SMS selects volumes from the secondary volume list and subsequently from the tertiary volume list. Selection continues until the allocation is successful, or until there are no more available volumes and the allocation fails.

For multistriped data sets, volumes are classified as primary and secondary. Primary volumes are preferred over secondary volumes. A single primary volume is randomly selected for each unique controller, and all other eligible volumes behind the same controller are secondary. Secondary volumes are randomly selected if initial allocation on the primary volume is unsuccessful. If the controller supports striping, there is no preference in different controller models.

You can mix devices of varying performance characteristics within one storage group. For example, if you specify a nonzero IART in the storage class, mountable volumes are considered before DASD volumes. If the IART is zero, mountable volumes are not considered and a DASD volume is selected. You can also add new devices into an existing z/OS complex and take advantage of different performance and availability characteristics.

After the system selects the primary allocation volume, that volume's associated storage group is used to select any remaining volumes requested.

SMS interfaces with the system resource manager (SRM) to select from the eligible volumes in the primary volume list. SRM uses device delays as one of the criteria for selection and does not prefer a volume if it is already allocated in the jobstep. This is useful for batch processing when the data set is accessed immediately after creation. It is, however, not useful for database data that is reorganized at off-peak hours.

SMS does not use SRM to select volumes from the secondary or tertiary volume lists. It uses a form of randomization to prevent skewed allocations, in instances such as when new volumes are added to a storage group or when the free space statistics are not current on volumes. You can force SMS not to use SRM by specifying a non-zero IART value.

**Related Reading:** For more information about volume selection and data set allocation, see *z/OS DFSMSdfp Storage Administration*.

### **Selecting Volumes for Striped Extended-Format Data Sets**

When multiple storage groups are assigned to an allocation, SMS examines each storage group and selects the one that offers the largest number of volumes attached to unique control units. This is called control unit separation.

Once a storage group has been selected, SMS selects the volumes based on available space, control unit separation, and performance characteristics if they are specified in the assigned storage class.

**Related Reading:** For information about striping volume selection with SMS, see *z/OS DFSMSdfp Storage Administration*.

## Managing Virtual I/O with SMS

**Recommendation:** Set up a virtual input/output (VIO) storage group. The VIO storage group is treated differently from pool storage groups. If the data set being allocated is eligible for virtual I/O and its primary space request does not exceed the VIO storage group's VIOMAXSIZE parameter, the VIO storage group is selected regardless of the order of its concatenation. This is called *preferred selection*.

You can control VIO usage on each system that shares your SMS configuration through the use of SMS STORAGE GROUP STATUS. In this way, different VIO storage groups having VIOMAXSIZEs tailored for the system can be selected. By setting a specific system's VIO SMS STORAGE GROUP STATUS to DISABLE(ALL), VIO allocation can be prevented on that system.

## Separating Large Data Sets

SMS allocation services perform optimally when you define a limited number of storage groups to contain your volumes. However, you might want to isolate large data sets in their own storage group because of their unique space management requirements. Additionally, by separating large data sets from other data sets, you prevent allocation failures that can occur due to volume fragmentation, since allocation can fail if the requested quantity cannot be satisfied in five extents.

The space requirements for large data sets can limit the free space available to other data sets. Also, more space must be reserved on volumes to support the new allocation or DFSMSHsm recall of large data sets. Because of this, the high allocation/migration thresholds for storage groups containing large data sets should be set lower than for storage groups containing normal-sized data sets.

Table 4 provides a list of recommended sizes to determine what constitutes a large data set, according to the type of DASD volume.

*Table 4. Large Data Sets (by DASD Type)*

| DASD Model           | Minimum Data Set Size (MBs) | Minimum Data Set Size (Cylinders) |
|----------------------|-----------------------------|-----------------------------------|
| 3380 Standard Models | 65                          | 90                                |
| 3380 Model E         | 130                         | 180                               |
| 3380 Model K         | 190                         | 270                               |
| 3390 Model 1         | 95                          | 115                               |
| 3390 Model 2         | 190                         | 225                               |
| 3390 Model 3         | 285                         | 335                               |
| 3390 Model 9         | 855                         | 1005                              |
| 9345 Model 1         | 100                         | 145                               |
| 9345 Model 2         | 150                         | 220                               |

Table 4. Large Data Sets (by DASD Type) (continued)

---

| DASD Model | Minimum Data Set Size (MBs) | Minimum Data Set Size (Cylinders) |
|------------|-----------------------------|-----------------------------------|
|------------|-----------------------------|-----------------------------------|

---

**Note:** MB and cylinder values are rounded up to the nearest multiple of 5.

---

Your own definition of large should provide for successful DFSMSHsm recall in five or fewer extents, based on the threshold you have set for the storage group.

Management class can be used to limit the negative effects of large data sets during the space management process. As an example, you can migrate large data sets directly to migration level 2, rather than letting them move to migration level 1.

Another management option is to place large data sets in your primary storage group and assign a management class that prevents migration. This is a good solution if you have only a few large data sets that need to be immediately available on DASD.

## Avoiding Allocation Failures

You can reduce instances of allocation failures due to space constraints or volume fragmentation by assigning a data class with a value of YES for the Space Constraint Relief attribute. When you specify space constraint relief and normal allocation fails, SMS bypasses the DADSM five extent limit. It also tries to spread the requested space among multiple volumes, as long as the data set is eligible to be allocated on multiple volumes. SMS also reduces the requested quantity by the percentage you have specified for the subparameter of the Space Constraint Relief data class attribute.

---

## Managing Tape Data with DFSMSrmm

DFSMSrmm invokes the SMS ACS routines to derive storage group and management class names for non-system-managed tape data sets.

### Using Management Classes to Manage Tape Data

With the SMS ACS &ACSENVIR read-only variable set to RMMVRS, DFSMSrmm invokes the management class ACS routine to determine the management class name for tape data sets on non-system-managed storage. DFSMSrmm then uses the VRS policy defined for this management class name in a similar way as it assigns VRS policies for tape data sets on SMS-managed storage.

### Using Storage Groups for Volume Pooling

For a system-managed manual tape library, the operator can select from a common scratch pool or be directed by the tape management system to a specific scratch pool. The volume stays in this storage group while in private status. Pooling in a system-managed manual tape library can be managed by your tape management system. See SC23-6874 for information about how DFSMSrmm can use the storage group name as the pool name for both MTL and non-system managed volumes. The volume stays in this storage group even after it is returned to scratch status.

## Related Reading for Managing Tape Data with DFSMSrmm

For more information about managing tape data with DFSMSrmm, see *z/OS DFSMSrmm Implementation and Customization Guide* and *z/OS DFSMSdfp Storage Administration*.

---

### Designing Your ACS Routines

The milestone introduced in Chapter 6, “Managing Permanent Data,” on page 119 divides data into major categories that have similar management characteristics.

After you define the SMS classes and groups needed to support your data, you develop ACS routines to assign these classes and groups to new data sets, as well as to data sets that result from migrating volumes to system management, or from copying or moving data to system-managed volumes.

### Using the ACS Language and Variables

---

#### Programming Interface Information

---

ACS routines use a simple, procedure-oriented language. ACS routines begin with a PROC statement that identifies the type of ACS routine, such as storage class ACS routine, and terminates with an END statement. Variables that are generated from the allocation environment for the data, such as data set name and job accounting information, are supplied to your routines as ACS read-only variables. You can use these variables to assign a class or group to a data set. This is done by assigning the corresponding read/write variable in your ACS routine. Each routine sets only its own read/write variable, as follows:

- The data class ACS routine sets &DATACLAS
- The storage class ACS routine sets &STORCLAS
- The management class ACS routine sets &MGMTCLAS
- The storage group ACS routine sets &STORGRP

The ACS language consists of the following statement types:

#### **DO/END**

Use this to define a set of actions that should be run in sequence. Typical use is to assign a class or group to the data set and exit the ACS routine.

**EXIT** Enables you to stop processing your ACS routine. You can assign a non-zero return code, using the EXIT statement, to stop allocation.

#### **FILTLIST**

Lets you assign a variable to a set of literals or masks. The system uses these variables, defined by filter lists, in comparisons with ACS variables associated with the data set allocation to help assign classes and groups.

#### **IF/THEN**

Enables you to compare ACS variables with your own variables created using FILTLIST statements to help assign classes and groups.

#### **SELECT/WHEN**

Lets you test variables and run a set of actions logically enclosed by DO and END statements.

**SET** Enables you to assign a class or group.

## WRITE

You can format messages for users requesting the data set allocation. You can display text or any read-only or read/write variable.

Read-only variables provided to your ACS routines are dependent on the allocation environment. All variables are available for new allocations. However, DFSMSdss CONVERTV, DFSMSrmm RMMVRS or RMMPOOL, and DFSMSshsm RECALL or RECOVER environments have a subset of read-only variables. Also, the storage group ACS routine has a restricted set of read-only variables available. *z/OS DFSMSdfp Storage Administration* describes the read-only variables and the ACS language.

End Programming Interface Information

## Using ACS Installation Exits

### Programming Interface Information

DFSMS contains three sample ACS installation exits: one each for data class, storage class, and management class. After each ACS routine runs, the corresponding ACS exit routine is called. In an exit, you can call other programs or services. The parameters passed to the exit are the same as the ACS routine read-only variables.

You can also use installation exits to override the ACS-assigned classes or to alter the ACS variables and rerun the ACS routine.

A storage class ACS exit is provided in the SYS1.SAMPLIB distributed in the Custom Built Installation Processing Option (CBIPO) product for the z/OS operating environment. You can use it to display all the ACS variables for an allocation, or to debug any ACS routine design or coding errors.

If you do not need these installation exits, do not supply dummy exits. Otherwise, you incur unnecessary overhead.

**Tip:** A pre-ACS routine exit is available with the VTS Export/Import SPE. The purpose of this exit is to let tape management systems provide read-only variables to the ACS routines to facilitate tape-related decision making.

**Related Reading:** For more information on using installation exits, see *z/OS DFSMS Installation Exits*.

End Programming Interface Information

## Using ACS Indexing Functions

### Programming Interface Information

The ACS language provides the following three indexing functions to help you make class or group assignments:

- &ALLVOL and &ANYVOL
- Accounting information
- Data set qualifier



The &ALLVOL and &ANYVOL indexing functions let you compare the volume serial numbers specified on input with a comparison variable, such as FILTLIST.

You can use the accounting information indexing function to refer to specific fields in the JOB or STEP accounting information.

With the data set qualifier indexing function, you can index the DSN variable to refer to specific qualifiers. For example:

- Value for &DSN is 'A.B.C.D'  
Value for &DSN(3) is 'C'  
Value for &DSN(&NQUAL) is 'D'

**Related Reading:** For detailed descriptions of variable syntax and use, see *z/OS DFSMSdfp Storage Administration*.

End Programming Interface Information

## Using FILTLIST Statements

### Programming Interface Information

The FILTLIST statement defines a list of literals or masks for variables used in IF or SELECT statements. The following example shows the syntax for this statement:

- FILTLIST name < INCLUDE(filter list) >  
          < EXCLUDE(filter list) >

FILTLIST statements cannot contain numeric values and, therefore, cannot be used in comparisons with the numeric variables &NQUAL, &NVOL, &SIZE, &MAXSIZE, &EXPDT, or &RETPD.

The ACS routine fragment in Figure 14 shows how you can use the FILTLIST masks to compare with a read-only variable (such as &DSN) to determine which data sets should receive specific performance services.

```
FILTLIST LIST1 INCLUDE (SYS1.**)  
                  EXCLUDE ('SYS1.PHONBOOK')  
FILTLIST LIST2 INCLUDE (IMS*, TSO*)  
                  EXCLUDE ('TSO256','IMS191')  
IF &DSN = &LIST1 THEN  
  SET &STORCLAS = 'FAST'  
IF &ANYVOL = &LIST2 THEN  
  SET &STORCLAS = 'STANDARD'
```

Figure 14. Using the FILTLIST Statement

End Programming Interface Information

## Using SELECT Statements

### Programming Interface Information

The SELECT statement defines a set of conditional execution statements. This statement has two formats: *format-1* and *format-2*. Format-1 has an implied = operator between the SELECT variable and the WHEN value, as follows:



- ```

SELECT ( variable )
  WHEN ( constant )
      statement
  WHEN ( constant or
        filtlist_name )
      statement
  OTHERWISE
      statement
END

```

A format-2 SELECT statement has no select variable:

- ```

SELECT
  WHEN ( condition )
      statement
  WHEN ( condition )
      statement
  OTHERWISE
      statement
END

```

All IBM-supplied ACS environment conditions are tested by the WHEN clauses. The comment, `/* INSTALLATION EXIT */`, indicates that the OTHERWISE clause is run only if an installation exit has set `&ACSENVIR` to a value you defined in an ACS routine exit. Figure 15 shows a format-1 SELECT statement with the select variable `ACSENVIR`:

```

SELECT (&ACSENVIR)
  WHEN ('ALLOC')
    DO ... END
  WHEN ('RECALL')
    DO ... END
  WHEN ('RECOVER')
    DO ... END
  WHEN ('CONVERT')
    DO ... END
  OTHERWISE /* INSTALLATION EXIT */
    DO ... END
END

```

Figure 15. Example of the Format-1 SELECT Statement

When coding ACS routines, remember that the WHEN clauses of the SELECT statement are tested serially. The first WHEN that is true causes its clause to be run. After the first true WHEN is encountered, the rest of the SELECT is not run. Figure 16 shows a format-2 SELECT statement:

```

SELECT
  WHEN (&RETPD < 7)
    SET &MGMTCLAS = 'DELSOON'
  WHEN (&RETPD < 30)
    SET &MGMTCLAS = 'FREQBKUP'
  WHEN (&RETPD < 60)
    SET &MGMTCLAS = 'NORMAL'
  OTHERWISE
    SET &MGMTCLAS = 'DELNEVER'
END

```

Figure 16. Example of the Format-2 SELECT Statement

End Programming Interface Information

## Using Advanced ACS Routine Design and Coding Techniques

For optimal results, use the following guidelines when you develop your ACS routines.

### Selecting ACS Variables

#### Programming Interface Information

Design your ACS routines to function reliably for the long term. Use ACS variables that sustain their meaning over time and in various environments, such as &DSN rather than &ACCT\_JOB. Variables such as &ACCT\_JOB can assume different values depending on the operating environment. &ACCT\_JOB might be a significant variable to test to determine a management class in the environment when the data set is being allocated. However, if DFSMSHsm recalls the data set, the value of the &ACCT\_JOB variable changes.

Within your SMS complex, you might need to know the system and Parallel Sysplex<sup>®</sup> where an ACS routine is executing to direct the allocation to a storage group that is accessible from the current system. You can use the &SYSNAME (set to the system name) and &SYSPLEX (set to the Parallel Sysplex name of the system where the ACS routine is executing) read-only variables. You can use these variables to distribute a single set of ACS routines to all the systems in your enterprise. The receiving sites do not have to make any changes to the ACS routines. However, if you have some down-level systems, they do not support these variables. Also, you should be careful using the &SYSNAME and &SYSPLEX variables on JES3 systems, because the system where the ACS routines are run might be different from the system where the job is run.

#### End Programming Interface Information

### Implementing Coding Techniques

#### Programming Interface Information

Use the same coding rules to ensure maintainability that you would use if coding one of your applications:

- Divide the routine into logical sections, one for each significant data classification, such as TSO, database, and so on.
- Keep a change log in the beginning of each ACS routine that includes a description of the coding change, the initials of the person making the change, and the change date.
- Create meaningful names for FILTLIST variables. A FILTLIST variable can be 31 characters long. Use the underscore character to make the variable more readable.
- Create meaningful names for classes and groups. The name should describe the type of service rather than the data classification.
- When you code a SELECT, always code an OTHERWISE.
- When you make a class or group assignment, code an EXIT statement.
- Use SELECT/WHEN in preference to the IF/THEN structure to enhance reliability and maintainability of the routine.
- Use comments freely throughout the routine to relate the ACS routine design objectives to the code that implements the design.

## Testing Your ACS Routines

### Programming Interface Information

You can use NaviQuest to test changes to your ACS routines. For more information about NaviQuest, see *z/OS DFSMSdfp Storage Administration*.

---

## Placing Your Volumes under System Management

Each of your DASD volumes is either system-managed or non-system-managed. A data set on DASD is system-managed when it satisfies the following requirements:

- It has a valid SMS storage class assigned.
- It resides on a volume in an SMS storage group that has been initialized as a system-managed volume or has been migrated to system management.

The value of SMS VOLUME STATUS shows the relationship of the volume to SMS. Your volumes can assume three states:

### Converted

Indicates that the volume is fully available for system management. All data sets on the volume have a storage class and are cataloged in an integrated catalog facility catalog.

**Initial** Indicates that the volume is not fully available for system management because it contains data sets that are ineligible for system management.

An attempt was made to place the volume under system management, but data sets were determined to be ineligible for system management based either on SMS eligibility rules or on the decisions made in your ACS routines. Temporary failure to migrate to system management occurs when data sets are unavailable (in use by another application) when the migration is attempted.

No new data set allocations can occur on a volume with initial status. Also, existing data sets cannot be extended to another volume while the volume is in this state.

You can place volumes in initial status as you prepare to implement system management.

**Tip:** You can use the DFSMSdss CONVERTV function with the TEST option to determine if your volumes and data sets are eligible for system management. See “Testing the Eligibility of Your Volumes and Data Sets” on page 55 for more information on the TEST option.

### Non-SMS

The volume does not contain any system-managed data sets and has not been initialized as system-managed.

You can either do data conversion with movement using DFSMSdss's COPY or DUMP/RESTORE functions, or you can convert in-place using DFSMSdss's CONVERTV function. The approach you use to place your data under system management depends on the following considerations:

- The degree of centralization of the storage management function

If most of your data resides in centrally-managed storage pools, you might be able to use the in-place conversion approach. TSO and database data might be migrated using this technique, because these data categories are likely to have similar management characteristics and reside in storage pools.

In contrast, if your storage is mainly application-owned, you must do data conversion with movement. Also, batch application data requires data movement, because this data has more diverse management requirements and is less likely to be pooled. For this data category, implementing system management on an application-by-application basis is most appropriate.

- Your inventory of available DASD volumes  
If few volumes are available for new allocation, you can use in-place conversion for the initial migration.
- The eligibility of your data to be system-managed  
For example, data sets that are unmovable or ISAM cannot be system-managed. Eligible data sets that coexist on these volumes must be moved to system-managed volumes.
- The availability of your data sets  
The in-place conversion technique requires exclusive use of the data sets on the volumes to be migrated. You should do this during off-peak hours.
- The ability of your storage configuration hardware to deliver target performance and availability services  
Use the conversion with data movement approach if:
  - Your storage hardware configuration has cached and non-cached storage controls, to achieve better performance and greater automation of performance management.
  - You plan to use sequential data striping.
  - You plan to use point-in-time copy, to ensure that data sets requiring high-performance backup are placed on volumes attached to cache-capable 3990 storage controls with the extended functions, or behind the RAMAC Virtual Array with the SnapShot feature and the DFSMSdss virtual concurrent copy SPE installed.
  - You plan to use dual copy, to place the data sets on volumes attached to cache-capable 3990 storage controls with the extended functions.

One benefit of doing conversion with data movement is that the data is allocated according to the allocation thresholds that you set for the storage groups, so that space usage can be balanced.

**Tip:** When doing conversions-in-place, consider that the ACS variables available to your routines are more limited when using the DFSMSdss CONVERTV function. For more information, see the DFSMSdss section of *z/OS DFSMSdss Storage Administration*.

## Converting with Data Movement

When you are converting data with movement, do the following tasks:

1. Define and initialize system-managed volumes.
2. Check the eligibility of the data sets.
3. Move data sets to system-managed volumes.
4. If necessary, revert back to non-system-management.

## Defining and Initializing System-Managed Volumes

Establish a set of new volumes in a storage group. Use the ICKDSF INIT command to initialize the volume with an indexed VTOC. The STORAGEGROUP keyword requests that ICKDSF initialize the volume as system-managed. You can enter this command from the ISMF Volume Application. Preallocate the VVDS using access method services, or let DFSMSdfp automatically allocate one when the first data set is allocated.

**Related Reading:** For more information on using AMS to define a BCS and VVDS, see *z/OS DFSMS Managing Catalogs*.

## Checking the Eligibility of the Data Sets

Use DFSMSdss's CONVERTV TEST against the volumes to check the eligibility of the data sets and the classes assigned.

## Moving Data Sets to System-Managed Volumes

Use DFSMSdss's logical COPY function to move the data to system-managed volumes. Specify one or more non-system-managed volumes as target output volumes to receive the data sets that your ACS routines determine should be non-system-managed.

When you use the logical COPY function to move the data to system-managed volumes, DFSMSdss performs the following steps:

1. Searches any catalogs for data sets cataloged outside the standard search order.  
You cannot use JOBCAT and STEPCAT statements to locate system-managed data sets. When placing data sets under system management, you can use the DFSMSdss INCAT keyword to specify catalogs that must be searched to find data sets cataloged outside the standard search order.
2. Copies all eligible data sets to system-managed volumes.  
These data sets are also assigned an SMS storage class and management class.
3. Places ineligible data sets on the non-system-managed volumes you specified on the OUTDD or OUTDYNAM parameter.  
When you copy or restore a data set when SMS is active, the storage class ACS routine is always run. The management class and storage group ACS routines are run if the storage class ACS routine determines that the data set should be system-managed.
4. Catalogs all system-managed data sets in the standard search order.

## Removing Data from System Management

This section identifies keywords that DFSMSdss supports that you can use to control SMS class assignments and remove data sets from system management.

**Restrictions:** These keywords have the following restrictions:

- You must be authorized by RACF for the classes specified for STORCLAS and MGMTCLAS.
- You must be authorized to use the RACF facility, BYPASSACS.

To restrict the use of the BYPASSACS keyword for the DFSMSdss COPY and RESTORE functions, define the following RACF profiles with the RACF facility resource class:

```
STGADMIN.ADR.COPY.BYPASSACS
STGADMIN.ADR.RESTORE.BYPASSACS
```

Using the BYPASSACS keyword lets you bypass the ACS routines for this run of DFSMSdss COPY or RESTORE. Either the class names now assigned to the data

set, or the class names specified using the STORCLAS or MGMTCLAS keywords, are used instead of having the ACS routines determine them. Then you can determine the storage and management class assigned to the data set. However, the storage group ACS routine is never bypassed.

Conversely, you can use the COPY and RESTORE commands to remove a data set from system management. Do this by specifying both the BYPASSACS and NULLSTORCLAS keywords as part of the command. The ACS routines are then bypassed, and the data set reverts to non-system-managed status.

**Related Reading:** For more information about using these DFSMSdss facilities, see *z/OS DFSMSdss Storage Administration*.

## Converting Data In-Place

DFSMSdss's CONVERTV function verifies that the following volume-level and data set-level requirements are met before placing the volume under system management:

- Verifies that the volume is in a valid storage group in the active SMS configuration
- Verifies that the volume contains an indexed VTOC
- Verifies that the volume is online
- Determines the eligibility of all data sets on the volume to be system-managed. The following data sets are ineligible for system management:
  - ISAM data sets
  - Absolute track allocated data sets
  - Unmovable data sets
  - GDG pattern DSCBs
  - Data sets cataloged in multiple catalogs
  - Data sets that are currently open
  - VSAM spheres, if all components are not defined in the same catalog
  - Multivolume data sets that have parts on volumes with the following criteria:
    - Do not have an indexed VTOC
    - Are not online
    - Are not defined in a storage group or are not in the same storage group

CONVERTV then allocates a VVDS if one does not already exist. CONVERTV also updates the basic catalog structure (BCS) and VVDS with the SMS storage and management classes assigned by your ACS routines for data sets that meet SMS eligibility requirements.

If the volume and data sets do not meet all SMS requirements, DFSMSdss sets the volume's physical status to initial. This status lets data sets be accessed, but not extended. New allocations on the volume are prohibited. If all requirements are met, DFSMSdss sets the volume status to converted.

Use the following required steps to prepare for in-place data conversion:

1. Design your SMS configuration including classes, groups, and ACS routines.
2. Determine how your DASD volumes should be assigned to SMS storage groups.
3. Determine if your volumes and data sets are eligible for system management, and remove any ineligible ones.
4. Stabilize the volumes prior to placing them under system management.
5. Place the volumes under system management.

6. Evaluate the results.
7. Revert back if necessary.

## Designing Your SMS Configuration

You can effectively implement system-managed storage by migrating one part of your data at a time. Use any of the following methods:

- The implementation-by-milestone approach allows you to achieve an orderly, phased implementation. With this method, you develop classes and groups required for a class of data, such as TSO, and then code the corresponding ACS routines. Once you have migrated a class of data, modify your configuration to support an additional type of data until all your eligible data has been placed under system management. Your ACS routines can then evolve to manage more and more of your data sets.
- Convert all eligible data to system management. Make sure to assign proper management class policies to prevent, for example, DFSMSHsm migration or backup of database data.
- Use facilities unavailable under non-SMS storage, including compression, striping, and extended addressability (>4 GB) VSAM.
- Use the DFSMS FIT approach as discussed in “Using DFSMS FIT to Implement System-Managed Storage” on page 19.

## Assigning Volumes to SMS Storage Groups

Assign each volume to a storage group in your active SMS configuration. Ensure that your ACS routine does not assign storage classes that do not match the capabilities of the devices being placed under system management. For example, do not assign a data class which specifies IF EXT=REQUIRED, or a storage class with the following attributes to a storage group which does not have volumes supporting these features:

- Availability=CONTINUOUS
- Accessibility=CONTINUOUS
- IART=0

## Testing the Eligibility of Your Volumes and Data Sets

You must ensure that the VTOC is indexed and that adequate space is available in the VVDS to contain the updates to the VSAM volume records (VVRs) and NVR records that are built for the non-VSAM data sets.

Use the TEST option of DFSMSdss's CONVERTV function to verify that your volume and data sets are eligible to be placed under system management.

You can use the ISMF Data Set Application to identify ineligible data sets before you perform the in-place conversion. This is described in Chapter 3, “Enabling the Software Base for System-Managed Storage,” on page 67. Also, you can use CONVERTV as a line operator from the ISMF Volume Application to build the job that performs the volume migration. Running CONVERTV with the TEST keyword simulates the migration to system-managed storage, letting you evaluate the eligibility of your volume without actually migrating the data. Be aware that the simulation reports on permanent, rather than transient, error conditions. As an example, data sets that cannot be serialized are not reported as exceptions during the simulation.

**Related Reading:** For more information on using the CONVERTV command, see *z/OS DFSMSdss Storage Administration*.



## Stabilizing the Volumes Prior to Placing them under System Management

Use CONVERTV with the PREPARE option to prevent new allocations on a volume that is ready to be placed under system management without actually migrating eligible data sets to system management. This stabilizes the number of data sets in preparation for migration. The PREPARE option sets the volume status to initial and gives you time to move ineligible data sets from the volume.

## Placing the Volumes Under System Management

Schedule the migration for a time when the data sets on the volumes can be serialized by CONVERTV. Consider that the elapsed time required is proportional to the number of data sets on the volume. Use CONVERTV with the SMS option to migrate the volume to system-managed storage.

## Evaluating the Results

If your ACS routine class assignments are in error, you can use the REDETERMINE keyword of CONVERTV to run your revised ACS routines. This reassigns storage class and management class for all data sets on a volume, even for those that already have classes and storage groups associated with them.

## Reverting Back to Non-System-Management

Using the NONSMS parameter of CONVERTV, you can change a system-managed volume to a non-system-managed volume. All data sets on the volume are removed from system management if the DFSMSdss function is successful. After running this command, only data sets that are not system-managed can be allocated on the volume.

You can specify the TEST keyword with NONSMS. The actual change is not performed, but the function generates a report showing data sets eligible for removal from system management. It also indicates whether the volume as a whole is eligible for placement under system management.

---

## Gaining Support for SMS from Your Users

A major goal of SMS is to increase data and storage management automation for both the storage administrator and user. If your installation was not centrally-managed before implementing system-managed storage, your users should be greatly relieved to have the burden of doing their own space and availability management removed. Even centrally-managed installations should experience greater productivity because of the greater degree of automation achieved using SMS facilities.

**Recommendation:** Some SMS enhancements require JCL and procedural changes that might affect your users. Train your users to understand SMS's capabilities and teach them about their new role in the data and storage management process.

The remaining sections in this chapter summarize the following topics:

- The benefits of SMS's enhanced data and storage management from the user's perspective
- The planning considerations for implementing the new data management techniques
- Some considerations for evaluating the effect of SMS on your existing procedures and JCL
- Some considerations for using the ISPF/PDF and ISMF user interfaces



Detailed information is contained in *z/OS DFSMSdfp Utilities* and *z/OS MVS JCL User's Guide*.

## Identifying SMS Benefits to Users

SMS data set allocation is simpler and more uniform across user interfaces such as JCL, TSO, and dynamic allocation, because of the following benefits:

- Removal of the need to specify UNIT or VOLUME
- Extensions to VSAM allocation facilities
- Improvements to application-oriented space allocation
- Enhancements to data set creation using models
- Automatic incorporation of your company's policies on data protection, performance, and storage cost
- Data Class space override attribute can now override user specified space information, such as JCL, TSO and IDCAM Define.

### Allocating without Specifying a Device

SMS enhances device-independent space requests. SMS assigns the volumes that meet the data set's availability requirements and most closely meet its performance requirements without overcommitting the storage subsystem's available space.

### Allocating VSAM Data Sets

VSAM data sets can be allocated using JCL, TSO ALLOCATE, and dynamic allocation. Use the REORG parameter to identify the type of VSAM data set that is allocated, such as key-sequenced. Default VSAM allocation parameters are the same as for the access method services DEFINE command.

VSAM component names are derived from DSNAME. For a key-sequenced VSAM cluster, the data component is named DSNAME.DATA and the index component is named DSNAME.INDEX.

VSAM data sets can now be allocated as temporary data sets.

### Using Application-Oriented Allocation

The SPACE parameter supports allocation for records. The AVGREC keyword lets you specify a unit (or scale), such as bytes, KBs, or MBs, and an average record size. Primary and secondary space values are expressed in terms of the number of records expected initially and from update activity. If you specify BLKSIZE=0 or omit BLKSIZE, the system calculates the most space-efficient block size based on the volume selected.

### Using Models to Create Data Sets

Two features assist you to more easily create data sets by using existing data sets as models:

**Using LIKE and REFDD:** The LIKE keyword lets you copy data characteristics from an existing data set to a new one. Using the LIKE keyword, you can copy the REORG, RECFM, SPACE, LRECL, KEYLEN, KEYOFF attributes to a data set that you are creating. REFDD can be used when the data set is allocated earlier in the job.

**Using DATACLAS:** You can either specify the DATACLAS keyword or have data characteristics assigned automatically.

*External specification of DATACLAS*

Users can specify the DATACLAS keyword to select a set of model data set

characteristics for a new allocation. Users can override any model characteristics by specifying the appropriate JCL DD statement parameter. With the new space override attribute in Data Class, users can override the JCL DD statement SPACE related parameters with what is define in the Data Class. This override is all inclusive, which means, if override is ON, then the space parameters in Data Class will be use instead of the parameters specified in JCL.

*Automatic assignment of data characteristics*

Users can have data attributes assigned by your data class ACS routine.

## Defining Data Classes to Simplify Data Set Allocations

A data class defines what the data looks like and contains attributes that correspond to parameters that can be coded on JCL DD statements, TSO ALLOCATE commands, or requests for dynamic allocation. It is a collection of allocation and space attributes used to create a data set. You define data classes for data sets that have similar attributes. When end users allocate a data set and refer to a data class, SMS allocates the data set using the attribute values of its associated data class, with the exception of override Data Class attribute. In that case, only the Data Class space attributes will be used.

Figure 17, Figure 18 on page 59, Figure 19 on page 60 and Figure 20 on page 60 show examples of the ISMF Data Class Define panel, with the attributes used to allocate an extended format, VSAM key-sequenced data set (KSDS).

**Related Reading:** For detailed descriptions of the data class attributes listed on the Data Class Define panel, see *z/OS DFSMSdfp Storage Administration*.

You can use Figure 17 to define record and space attributes.

```

Panel Utilities Scroll Help
-----
                                DATA CLASS DEFINE                                Page 1 of 6
Command ==>>

SCDS Name . . . : USER6.MYCDS
Data Class Name : DC1

To DEFINE Data Class, Specify:
Description ==>
==>
Recfm . . . . . (any valid RECFM combination or blank)
Lrecl . . . . . (1 to 32761 or blank)
Override Space . . . . N (Y or N)
Space Avgrec . . . . . (U, K, M or blank)
  Avg Value . . . . . (0 to 65535 or blank)
  Primary . . . . . (0 to 999999 or blank)
  Secondary . . . . . (0 to 999999 or blank)
  Directory . . . . . (0 to 999999 or blank)
Retpd or Expdt . . . . . (0 to 9999, YYYY/MM/DD or blank)
Volume Count . . . . . 1 (1 to 255 or blank)
Add'l Volume Amount . . . (P=Primary, S=Secondary or blank)
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 17. Data Class Define Panel, Page 1

You can specify additional volume and VSAM attributes on the second panel, shown in Figure 18 on page 59.

```

Panel Utilities Scroll Help
-----
                                DATA CLASS DEFINE                                Page 2 of 6
Command ==>>

SCDS Name . . . : USER6.MYSCDS
Data Class Name : DC1

To DEFINE Data Class, Specify:

Data Set Name Type . . . . . (EXT, HFS, LIB, PDS, Large or blank)
If Ext . . . . . (P, R or blank)
Extended Addressability . . N (Y or N)
Record Access Bias . . . . . (S, U, D0, DW, S0, SW or blank)
RMODE31 . . . . . (ALL, BUFF, CB, NONE or blank)
Space Constraint Relief . . . N (Y or N)
Reduce Space Up To (%) . . . (0 to 99 or blank)
Dynamic Volume Count . . . . (1 to 59 or blank)
System Managed Buffering . . . (1K to 2048M or blank)

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 18. Data Class Define Panel, Page 2

On this panel, you can specify the following attributes for a data class:

- Whether to allocate VSAM data sets in extended format.
- Whether to allocate a VSAM data set to use extended addressability, so that it can grow beyond the four gigabyte (GB) size (the data set must be allocated in extended format to be eligible for extended addressability)
- Whether to let VSAM determine how many and which type of buffers to use when allocating VSAM data sets in extended format
- Whether to retry new volume allocations or extends on new volumes that have failed due to space constraints
- Whether to dynamically add volumes to a data set when a new volume is needed, and how many to add (up to 59; valid only when space constraint relief is Y). See *z/OS DFSMSdfp Storage Administration* for more information about dynamic volume count.
- Whether to support VSAM data sets (both system-managed and non-system-managed data sets) with spanned record formats. Spanned record formats are those in which a data record can cross control interval boundaries
- Whether extended format KSDSs are able to contain compressed data. You can request that physical sequential data sets be compressed using either tailored or generic compression dictionaries. You can use the access method services DCOLLECT command, the ISMF display function, and SMF type 14, 15, and 64 records to assess the overall space savings due to compression
- Whether the data set could support extended attributes (format 8 and 9 DSCBs) and optionally reside in EAS. (EATTR option) .

Use page 4 of the ISMF Data Class Define panel, shown in Figure 19 on page 60, to further modify data classes.

```

Panel Utilities Scroll Help
-----
                                DATA CLASS DEFINE                                Page 4 of 6
Command ==>>>

SCDS Name . . . : Y421252.MYSCDS
Data Class Name : DCDC1

To DEFINE Data Class, Specify:

Media Interchange
Media Type . . . . . (1 to 13 or blank)
Recording Technology . . (18,36,128,256,384,E1,E2-E4,EE2-EE4 or ' ')
Performance Scaling . . . (Y, N or blank)
Performance Segmentation (Y, N or blank)
Block Size Limit . . . . . (32760 to 2GB or blank)
Recorg . . . . . (KS, ES, RR, LS or blank)
Keylen . . . . . (0 to 255 or blank)
Keyoff . . . . . (0 to 32760 or blank)
CISize Data . . . . . (1 to 32768 or blank)
% Freespace CI . . . . . (0 to 100 or blank)
CA . . . . . (0 to 100 or blank)
Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 19. Data Class Define Panel, Page 4

Use page 6 of the ISMF Data Class Define panel, shown in Figure 20, to specify whether to assign attributes for VSAM record-level sharing (RLS) to system-managed data sets.

You can specify whether the data set is eligible for backup-while-open processing and whether the data set is recoverable. You can specify the name of the forward recovery log stream. You can also specify the size of VSAM RLS data that is cached in the CF cache structure that is defined to DFSMS. You can also specify whether SMSVSAM is allowed to use 64-bit addressable virtual storage for its data buffers, moving them above the 2 gigabyte bar.

```

Panel Utilities Scroll Help
-----
                                DATA CLASS DEFINE                                Page 6 of 6
Command ==>>>

SCDS Name . . . : USER6.MYSCDS
Data Class Name : DC2

To DEFINE Data Class, Specify:
Shareoptions Xregion . . . (1 to 4 or blank)
Xsystem . . . (3, 4 or blank)
Reuse . . . . . N (Y or N)
Initial Load . . . . . R (S, R or blank)
BWO . . . . . (TC, TI, NO or blank)
Log . . . . . (N, U, A or blank)
Logstream Id . . . . .
FRLog . . . . . (A, N, R, U or blank)
RLS CF Cache Value . . . . A (A, N, U or D)
RLS Above the 2-GB Bar . . N (Y or N)
Extent Constraint Removal N (Y or N)
CA Reclaim . . . . . Y (Y or N)
Log Replicate . . . . . N (Y or N)

```

Figure 20. Data Class Define Panel, Page 6

You can develop data classes as you migrate permanent data to system-managed storage, or you can defer this until after most permanent data has been migrated. First, have users externally specify the keyword DATACLAS to select these data

classes. Later, you can further automate by assigning them in your ACS routines. This latter level of automation requires a plan to identify specific data set types by interrogating the values of ACS variables. A simple example is the ACS variable, &LLQ, which represents the data set's low-level qualifier.

Data classes can assist you in enforcing standards for data set allocation. The need to maintain MVS or DFSMSdftp installation exits that enforce allocation standards might be eliminated by using the data class ACS routine facilities. All keywords on the JCL DD statement, TSO ALLOCATE command, or dynamic allocation are passed to your data class ACS routine to help you determine how to allocate the data set. You can issue a warning message to your users if their allocation violates standards, such as specifying a specific volume or not using a secondary allocation request. Or, you can fail an allocation by setting a non-zero return code in your data class ACS routine. DFSMSdftp enables you to include any ACS variable in the informational messages you create.

You can override a data class attribute using JCL or dynamic allocation parameters. However, overriding a subparameter of a parameter can override ALL of the subparameters for that parameter. For example, SPACE=(TRK,(1)) in JCL can cause primary, secondary, and directory quantities, as well as AVGREC and AVGUNITS, in the data class to be overridden. However, if you also specify DSNTYPE=PDS, the directory quantity is taken from the data class.

If you want the data class to supply the default value of a parameter, do not specify a value for that parameter in the JCL or dynamic allocation. Users cannot override the data class attributes of dynamically allocated data sets if you use the IEFDB401 user exit. By default, SMS cannot change values that are explicitly specified because doing so would alter the original meaning and intent of the allocation. The default for the 'override space' attribute in Data Class is NO, but you can use this attribute to allow SMS to change explicitly specified values.

## Changing the JCL

IBM utilities support system-managed data sets. However, SMS catalog usage and device-independence might affect your jobs. You should evaluate the differences in SMS processing among the following areas:

- Disposition processing
- Locate processing
- VOL=SER usage
- VOL=REF usage
- UNIT usage
- UNIT=AFF usage
- IEHLIST processing
- IEHMOVE processing
- IEHPROGM processing
- IEBGENER processing
- Generation data group (GDG) processing

### Effects of Changed Disposition Processing

All system-managed DASD data sets must be cataloged. You must change all JCL that relies on data sets not being cataloged to work. System-managed DASD data sets are cataloged at creation rather than during disposition processing. As a result, you cannot reference a data set as being both new and old in a single step.

Temporary data sets are defined using `DSN=&&dsname` or without the `DSN` parameter supplied. JCL that allocates data sets temporarily by specifying `DISP=(NEW,PASS)` or `DISP=(NEW,DELETE)` are affected by the changed disposition processing. Data sets allocated with these dispositions are now cataloged and can conflict with cataloging or security access restrictions or, more commonly, with other jobs using the same data set name.

## Locate Processing

No STEPCATs or JOBCATs are permitted if system-managed data sets are referenced, because SMS only supports data set locate using the standard order of search. SMS does locate processing during step setup, so do not reference data sets as old that will not be cataloged until the job actually executes. If you do, the locate fails and you get a JCL error.

You can use DUMMY storage groups to minimize JCL changes for existing data sets that are located by specifying the volume. If you have placed the data set under system management and it no longer exists on the original volume or the volume no longer exists, you can add the volume to a DUMMY type storage group. When the data set is referenced (and not found), the system uses the catalog to locate the data set. However, you cannot use dummy storage groups to handle volume allocations. That is, you cannot use DD statements like:

```
//DD1      DD  VOL=SER=DUMMY1,UNIT=SYSDA,DISP=SHR
```

This type of statement (with `DISP=OLD` or `SHR`) allocates a specific volume. SMS manages data set allocations, and because this is not really a data set, SMS cannot take control of it.

## VOL=SER Usage

When you use `VOL=SER` for *data set stacking*, that is, to place several data sets on the same tape volume or set of tape volumes, at least one of the volume serial numbers specified must match one of the volume serial numbers for the data set on which this data set is being stacked. Use `VOL=SER` when stacking multivolume, multi-data sets within the same job step.

**Related Reading:** For more information on data set stacking, and on when to use `VOL=SER` versus `VOL=REF`, see “Data Set Stacking” on page 209.

## VOL=REF Usage

Without SMS, when you specify `VOL=REF` on the DD statement it indicates that the data set should be allocated on the same volume as the referenced data set. With SMS, specifying `VOL=REF` causes the storage class of the referenced data set to be assigned to the new data set. The storage group of the referenced data set is also passed to the storage group ACS routine, so you can code your ACS routine to perform one of the following three actions:

- Assign the same storage group to the new data set (required for SMS-managed tape `VOL=REFs`).

Storage group might be blank if you entered volumes into a tape library with a blank storage group name. However, you can assign storage group based on library name in this case.

- Assign a different storage group to the new data set.
- Fail the allocation.

The ACS routines are passed the following values in the `&ALLVOL` and `&ANYVOL` read-only variables when `VOL=REF` is used:

'REF=SD' - The reference is to an SMS-managed DASD or VIO data set

'REF=ST' - The reference is to an SMS-managed tape data set

'REF=NS' - The reference is to a non-SMS-managed data set

**Restrictions:** Consider the following restrictions when using VOL=REF:

- If the referenced data set is on SMS-managed tape, then the new data set must be allocated to the same storage group.
- For SMS-managed DASD and VIO data sets, the two data sets must be assigned to compatible *types* of storage groups to ensure consistency for locates. For example, if the referenced data set is allocated on DASD, then allocating referencing data set on tape could result in potential locate problems.
- For references to non-SMS-managed data sets, either allow or fail the allocation. This gives the ACS routines control over whether a new, non-SMS-managed data set can be allocated on a non-SMS-managed volume or not. SMS fails the allocation if the ACS routines attempt to make the referencing data set SMS-managed, since this could cause problems attempting to locate that data set with DISP=OLD or DISP=SHR, and lead to potential data integrity problems.

Remember these restrictions when you design your storage groups and ACS routines, and eliminate any use of VOL=REF that might violate these restrictions.

**Related Reading:** For examples of ACS routines used when allocating data sets using VOL=REF, see the following sections:

- “Data Set Stacking Using Volume Reference” on page 211
- “Using Volume Reference to System-Managed Data Sets” on page 215
- “Using Volume Reference to Non-System-Managed Data Sets” on page 216

With the IBM TotalStorage Enterprise Automated Tape Library (3494 or 3495), data sets do not have to be cataloged. If a data set is not cataloged, constructs (classes) assigned to it are lost. Even if the data set is cataloged, unlike the data sets on DASD, the SMS constructs assigned to the data set are not retained in the catalog. However, because it is now cataloged, a VOL=REF can be done to it by data set name.

When no storage class is available from a referenced data set, the storage class routine must assign a storage class to the referencing data set, enabling the allocation to be directed to the storage group of the referencing data set. Otherwise, the allocation fails.

## UNIT Usage

If your existing JCL specifies an esoteric or generic device name in the UNIT parameter for a non-system-managed data set, then make certain that it is defined on the system performing the allocation. Although the esoteric name is ignored for a system-managed data set, the esoteric must still exist for non-system-managed data sets, including DD DUMMY or DD DSN=NULLFILE statements that include the UNIT parameter.

An allocation which specifies a volume, unit, DISP=OLD or DISP=SHR, and omits a data set name, as shown in the example below, results in an allocation to the volume.

```
//DD1 DD VOL=SER=SMSVOL,UNIT=SYSDA,DISP=OLD  
//DD1 DD VOL=SER=SMSVOL,UNIT=SYSDA,DISP=SHR
```

If the volume is not mounted, a mount message is issued to the operator console. This occurs regardless of whether a volser is defined to an SMS storage group or to an SMS dummy storage group.



## IEHLIST Processing

You should review jobs that call IEHLIST, because this utility is highly device-oriented.

## IEHMOVE Processing

IEHMOVE is not supported for system-managed data sets.

## IEHPROGM Processing

You should review jobs that call IEHPROGM because this utility is highly device-oriented. The following changes apply to IEHPROGM processing for system-managed data sets:

- CATLG and UNCATLG options are not valid.
- SCRATCH, RENAME VOL parameter device type and volume lists must accurately reflect the actual device where the data set was allocated. You and SMS—not the user-control device selection.
- SCRATCH causes the data set to be deleted and uncataloged.
- RENAME causes the data set to be renamed and recataloged.
- SCRATCH VTOC SYS supports both VSAM and non-VSAM temporary data sets.

## IEBGENER Processing

Jobs that call IEBGENER have a system-determined block size used for the output data set if RECFM and LRECL are specified, but BLKSIZE is not specified. The data set is also considered to be system-reblockable.

**Tip:** You can use DFSORT's ICEGENER facility to achieve faster and more efficient processing for applications that are set up to use the IEBGENER system utility. For more information, see *z/OS DFSORT Application Programming Guide*.

## GDG Processing

When you define a generation data group (GDG), you can either scratch or uncatalog generation data sets (GDSs) as they exceed the GDG limit. Because SMS-managed data sets cannot be uncataloged, be aware of the following:

- How generation data sets become part of the generation data group
- How the NOSCRATCH option operates
- How you alter the GDG's limit

**Creating generation data sets:** A GDS state, *deferred roll-in*, preserves JCL that creates and references GDSs. When a new GDS is created, it is cataloged but assigned a status of deferred roll-in, so that the oldest generation is not deleted. The new GDS is only rolled in at disposition processing at job or step termination in an existing job when you code DISP=(OLD,CATLG) in a different job or use the access method services ALTER ROLLIN command. This has special implications on restarts after a job failure.

**Using NOSCRATCH:** Generation data groups that are defined using the NOSCRATCH option operate differently in the SMS environment. Generations that exceed the limit are assigned a status of *rolled-off*. Special management class attributes affect these data sets. You can have DFSMSHsm migrate them immediately or expire them.

**Altering GDG limit:** If you use the access method services ALTER command to increase the GDG limit, no rolled-off GDSs or deferred roll-in GDSs are rolled in. When you increase the GDG limit, you must use the ALTER ROLLIN command to



roll in the rolled-off generations if you need to access them. In contrast, GDSs are immediately rolled off if you decrease the limit.

## Identifying the User's Role

Your users create and manage data sets using ISPF/PDF. These utilities support SMS classes and device-independent data set allocation. Option M of ISPF/PDF 3.2 supports enhanced data set allocation, letting users enter allocation requests in a device-independent format. Data set information options are available in the following applications:

- Library utility (3.1)
- Data set utility (3.2)
- Data set list utility (3.4)

## Using the Interactive Storage Management Facility

Consider making ISMF available to your users. Several ISMF applications can assist users in maintaining data sets, including:

- Data set
- Data class
- Storage class
- Management class
- Aggregate group

The Data Set Application displays all data set characteristics including, for SMS-managed data sets, the classes assigned to the data set. Users can also view the most recent date that the data set has been backed up by DFSMShsm, or the number of stripes that the system has used to allocate the data set for any striped data sets. By using the Data Class, Storage Class, and Management Class applications, users can interactively review the attributes that you are using to manage their data sets. These online applications can supplement your own user's guide.

Data set lists can be generated by using data set filters. ISMF uses the same filter conventions that are standard throughout DFSMS. Users can also filter by data set characteristics. DFSMShsm and DFSMSdss commands can be used as ISMF line operators in the Data Set Application to perform simple storage management functions that supplement the automatic functions provided by DFSMS. For example, DFSMSdss's COPY command might be useful. Users can select COPY options on ISMF panels that result in a fully-developed batch job, which can be submitted and run in the background. Using ISMF, you do not need to know DFSMSdss syntax. Other useful commands are DFSMShsm's RECALL or RECOVER.

If you encourage users to understand and use these commands to do simple recovery actions, your storage administration tasks are greatly simplified.



---

## Chapter 3. Enabling the Software Base for System-Managed Storage

This chapter describes how you can prepare for your implementation of system-managed storage before you activate the Storage Management Subsystem (SMS).

Before activating SMS, you must perform the following activities:

- Provide RACF security for storage administrator tools and data sets unique to the DFSMS environment.

The tools implementing system-managed storage can alter allocation rules for all your data sets. Protect these tools from unauthorized users.

RACF also allows you to establish default data, storage, and management classes associated with a data set owner. Use these values in your ACS routines to help you assign storage resources and management services.

- Use ISMF to familiarize yourself with the starter set.

All SMS configuration changes are controlled through ISMF applications. The starter set provides some useful examples for implementing system-managed storage that can be browsed or altered using ISMF applications for the storage administrator.

- Use ISMF to identify data sets that cannot be system-managed.

Data sets must be movable and cataloged to qualify for system management. Most of the ineligible data sets can be identified and handled as part of the later implementation-by-milestone phases. If you know of applications that are candidates for early migration to system-managed storage, consider using ISMF to help you identify any data sets that are potential problems.

- Use ISMF to manage storage devices.

Many DFSMS performance and availability services depend on the functions of a cache-capable 3990 storage control. ISMF can effectively control the use of the extended functions, in addition to letting you accomplish routine storage management functions such as initializing DASD volumes and analyzing DASD hardware errors.

- Implement system-determined block size.

System-determined block size helps you efficiently use available space. You can then take advantage of DFSMS performance enhancements.

---

### Providing Security in the DFSMS Environment

You should work with your security administrator to create a security policy that supports system-managed data sets, and controls access to SMS control data sets, programs, and functions. RACF lets you define users and groups of users, their various attributes, and their rights and privileges to access data and use system facilities. RACF can also provide default data, storage, and management classes associated with a data set owner to your ACS routines to help you determine the storage resources and management services required by the data set.

With system-managed storage, RACF controls access to the following functions:

- System-managed data sets
- SMS control data sets
- SMS functions and commands

- Fields in the RACF profile
- SMS classes
- ISMF functions

**Related Reading:**

- For information on using the RACF control features, see *z/OS Security Server RACF Security Administrator's Guide*.
- For information on using RACF in a DFSMS environment, see *z/OS DFSMSdfp Storage Administration*.

## Protecting System-Managed Data Sets

You can use RACF to control access to data sets on system-managed volumes (both DASD and tape) and their associated catalogs. Although MVS and VSAM password protection is ignored for system-managed data sets, non-system-managed VSAM and non-VSAM data sets can be password-protected, RACF-protected, or both. If a data set is both RACF- and password-protected, access is controlled solely through the RACF authorization mechanism.

## Protecting SMS Control Data Sets

You can use RACF to protect your SMS control data sets from deliberate or accidental deletion or alteration. Senior administrators should have ALTER access to SMS control data sets to enable them to create, retrieve, update and delete the SMS control data sets. Allow only the storage administration group to have UPDATE access to SMS control data sets. Give other users READ access to the active SMS configuration so they can browse the SMS configuration using ISMF applications. Give the COMMDS and ACS routine libraries a universal access of NONE.

## Protecting Functions and Commands

You can use RACF FACILITY resource class to control the activation of the SMS configuration and to perform various catalog and DFSMSdss functions against system-managed data sets.

STGADMIN.IGD.ACTIVATE.CONFIGURATION, the FACILITY resource class profile, controls your ability to activate an SMS configuration from ISMF. You must define this RACF profile to use the ACTIVATE command from ISMF. An operator message is issued if the FACILITY resource class is inactive, or if the named profile does not exist. The operator must then confirm the request to activate the new configuration.

The following example shows the RACF commands issued to activate an SMS configuration:

- ```

SETROPTS CLASSACT(FACILITY)

RDEFINE FACILITY STGADMIN.IGD.ACTIVATE.CONFIGURATION UACC(NONE)

PERMIT STGADMIN.IGD.ACTIVATE.CONFIGURATION CLASS(FACILITY) -
      ID(STGADMIN) ACCESS(READ)

```

You can define general resource profiles to protect specialized DFSMSdss and access method services functions that are designed to protect the integrity of system-managed data sets. For example, you can use the BYPASSACS keyword when copying or restoring data sets using DFSMSdss. This overrides SMS class assignments and creates non-system-managed data sets or system-managed data

sets having externally-supplied classes. The BYPASSACS keyword prevents your ACS routines from running. The ability to uncatalog a data set is protected also because of the criticality of cataloging with system-managed storage.

You can create a separate RACF profile to individually authorize each function, keyword, and command for system-managed data sets. Or, using the common high-level qualifier STGADMIN, you can create RACF generic profiles for command or operation authorization.

**Related Reading:** For a list of the profiles you must define to protect catalog and DFSMSdss functions for system-managed data sets, see *z/OS DFSMSdss Storage Administration*.

## Restricting Access to Fields in the RACF Profile

You can use the RACF FIELD resource class with SMS to control the users' ability to specify or update the following fields in a RACF profile:

- Resource owner (RESOWNER) for data set profiles
- Application identifier (DATAAPPL) in user or group profiles
- Default SMS DATACLAS, STORCLAS, and MGMTCLAS values in user or group profiles

You can define default data, storage, and management class names, and an application identifier in RACF user or group profiles. SMS retrieves these defaults and supplies them as input variables to the ACS routines. You can use the application identifier to associate data sets that have different highest-level qualifiers and different resource owners.

To use default SMS classes for a data set, define a resource owner or data set owner in the RACF profile for that data set. RACF uses the resource owner to determine the user or group profiles that contain the default SMS classes.

Having a particular default STORCLAS or MGMTCLAS specified in the user or group profile does not imply that a given user is authorized to use the corresponding class. See "Restricting Access to Classes and Groups" on page 70 for information on authorizing the use of individual management and storage classes.

Be careful when assigning RACF defaults because it is unlikely that a given SMS class is applicable to *all* data sets created by a user or group. However, RACF defaults can be effectively used to supplement your ACS routine logic and handle class assignments for data sets that are difficult to identify using other ACS READ/ONLY variables.

Figure 21 on page 70 shows how you can use the RACF default to control the management class assigned to a data set.

```

PROC MGMTCLAS
:
:
  IF (&ACCT_JOB = 'P' | &ACCT_JOB = 'F' | &ACCT_JOB = 'A')
    THEN SELECT
      WHEN (&ACCT_JOB = 'P' && &STORCLAS = &DEF_STORCLAS)
        SET &MGMTCLAS = 'PAYROLL'

      WHEN (&ACCT_JOB = 'F' && &STORCLAS = &DEF_STORCLAS)
        SET &MGMTCLAS = 'FINANCE'

      WHEN (&ACCT_JOB = 'A' && &STORCLAS = &DEF_STORCLAS)
        SET &MGMTCLAS = 'ADMIN'

      OTHERWISE SET &MGMTCLAS = &DEF_MGMTCLAS

    END /* END OF SELECT */
:
:
END /* END OF PROC */

```

Figure 21. Controlling Management Class Assignments

Figure 22 shows an example of a command sequence you can use to define the SMS-related FIELD resource class profiles. The example enables storage administrators to update the resource owner field, and enables the user to update the set of SMS default classes.

```

SETROPTS CLASSACT(FIELD) RACLIST(FIELD)

RDEFINE FIELD DATASET.DFP.RESOWNER UACC(READ)

PERMIT DATASET.DFP.RESOWNER CLASS(FIELD) -
  ID(STGADMIN) ACCESS(UPDATE)

SETROPTS REFRESH RACLIST(FIELD)

SETROPTS CLASSACT(FIELD) RACLIST(FIELD)

RDEFINE FIELD USER.DFP.* UACC(READ)

RDEFINE FIELD GROUP.DFP.* UACC(READ)

PERMIT USER.DFP.* CLASS(FIELD) -
  ID(&
  RACUID) ACCESS(UPDATE)

PERMIT GROUP.DFP.* CLASS(FIELD) -
  ID(&
  RACUID) ACCESS(UPDATE)

SETROPTS REFRESH RACLIST(FIELD)

```

Figure 22. Defining Resource Class Profiles

## Restricting Access to Classes and Groups

Some SMS configuration elements, such as data class and storage group, do not require protection. No RACF protection for storage groups is needed because your ACS routines control storage group assignment entirely. Data classes do not need RACF protection because inappropriate use of data classes is unlikely to affect your storage resources. However, use of specialized storage classes can cause

high-performance cache facilities or dual copy availability services to be used to support a data set. Management class attributes mandating extra backups can be costly and should only be used for those data sets that require them. Use of these storage and management classes should be controlled by RACF or your ACS routines.

With system-managed storage, data ownership is the basis for determining who can use RACF-protected SMS resources. Previously, checking was based on the user's authority to use a resource. For system-managed data sets, the owner of the data set must be authorized to use the SMS classes. The RACF functions, such as data set protection and authorization control for data, programs, commands, and keywords, apply to databases as well.

RACF contains two resource classes: STORCLAS and MGMTCLAS. Authorize SMS storage and management classes by defining them as RACF profiles to the STORCLAS and MGMTCLAS resource classes. The profile names are the same as the names of the storage or management classes.

The following example shows the command sequence you can use to define a general resource class profile for the storage class, DBCRIT, and the database administrator's ability to use the DBCRIT. In the following example, the storage administration group is the owner of the general resource profile:

```
SETROPTS CLASSACT(STORCLAS) RACLIST(STORCLAS)
RDEFINE STORCLAS DBCRIT OWNER(STGADMIN) UACC(NONE)
PERMIT DBCRIT CLASS(STORCLAS) ID(DBA) ACCESS(READ)
SETROPTS REFRESH RACLIST(STORCLAS)
```

The RACF RESOWNER value, based on the high-level qualifier of the data set name, is the default used to check authorization to use management and storage classes. Another way to check this authorization is to use the user ID that is allocating the data set. This prevents the problems that can occur with restoring or recalling data sets that have a protected storage class and management class, and that are owned by users whose user or group IDs have been revoked.

Certain authorization functions are necessary beyond the data set level, and are outside the scope of RACF. Because of the special nature of these functions, some of them are implemented in particular database products, for example, DB2 and IMS.

## Protecting ISMF Functions

You can use the RACF PROGRAM resource class to control the use of ISMF, including individual ISMF applications, dialog commands, and line operators. With the RACF program control feature, you can set up authorization levels that meet your needs for each of these categories.

You can define several RACF profiles to limit the use of ISMF applications. However, make the list and display dialogs available to all users.

**Related Reading:** For a list of RACF profiles that can be defined to limit the use of ISMF applications, see *z/OS DFSMSdfp Storage Administration*.

Figure 23 shows the RACF commands that enable you to protect corresponding ISMF resources.

```
SETROPTS CLASSACT(PROGRAM) RACLIST(PROGRAM)

RDEFINE PROGRAM DGTFCAT UACC(NONE) -
    ADDMEM(<load library name>/<volser>/NOPADCHK)

PERMIT DGTFCAT CLASS(PROGRAM) -
    ID(stg-administrator-group) ACCESS(READ)

SETROPTS REFRESH WHEN(PROGRAM)
```

Figure 23. Protecting ISMF Functions

## Using ISMF to View the Starter Set

Use ISMF to perform the following tasks:

- Define SMS classes and storage groups
- Develop and test your ACS routines
- Activate your SMS configuration
- Define your tape libraries

You can use the SMS configuration and ACS routines on the product tape and in Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243 to familiarize yourself with the implementation-by-milestone sample configurations.

The base SMS configuration contains all the data classes, storage classes, management classes, and storage groups needed to implement all five milestones. A set of ACS routines accompanies each milestone, beginning with the activating SMS milestone, which is discussed in Chapter 4, “Activating the Storage Management Subsystem,” on page 79. The ACS routine design controls the assignment of classes to ensure that each milestone uses the subset of classes and groups needed during the milestone. You translate the ACS routines for the milestone to fully implement the SMS configuration.

## Viewing the Sample SCDS

The starter set includes a valid SCDS containing:

- Data classes and a data class ACS routine
- Storage classes and a storage class ACS routine
- Management classes and a management class ACS routine
- Storage groups and a storage group ACS routine

The starter set is a machine-readable PDS. Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243, describes sample SMS classes and groups, and contains additional sample ACS routines.

## Viewing the Sample ACS Source Routines

Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243 includes the following sets of ACS source routines as examples for developing your routines:

- Routines for the activating milestone
  - Cause data sets to be non-system-managed.



- Routines for the temporary milestone  
Help you manage temporary data sets.
- Routines for the permanent milestone  
Help you manage permanent data, including TSO, batch, and database data in addition to temporary data.
- Routines for the tape milestone  
Help you manage tape data sets in addition to permanent DASD data sets and temporary data sets.

Use the ISMF Data Class, Storage Class, Management Class, and Storage Group applications to list the attributes of the classes and groups. A list of the sample classes, groups, and ACS routines included in the starter set is contained in Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243.

---

## Using ISMF to Identify Ineligible Data Sets

The ISMF Data Set Application helps you to identify data sets that require special attention when you implement system-managed storage.

Most of your data sets can be system-managed. System-managed data sets must satisfy the following two major requirements:

- They must be cataloged  
DFSMSdss supports the automatic cataloging of data sets during migration to system management. The categories of uncataloged data sets that require additional planning before they can be migrated to system management are:
  - Pattern DSCBs  
Batch jobs typically use these data sets to generate DCB information for the new GDSs. You can defer handling these pattern DSCBs until you migrate the batch data that requires them. This is discussed in Chapter 8, “Managing Batch Data,” on page 135.
  - Uncataloged data sets whose names collide with data set names that are already cataloged in the standard search order
- They must be movable  
SMS is designed to reduce user awareness of device-oriented considerations. System-managed data sets should be able to reside on any volume that can deliver the performance and storage management services required. Unmovable data sets must be identified and, either isolated on non-system-managed volumes, or converted to a format that is supported in the DFSMS environment.  
**Recommendation:** Because there are application dependencies involved when changing the data set format, weigh the effort required to convert these data sets.

## Identifying Unmovable Data Sets and Absolute Track Allocation

All types of unmovable data sets must be identified. Use the ISMF Data Set Application to assist you in identifying these data sets. Follow these steps, in order:

1. Select all the data sets on a set of volumes, using the Data Set Selection Entry Panel, and choose Option 2 to generate a new list from the criteria specified, as shown in Figure 24 on page 74.

```

Panel Defaults Utilities Scroll Help
-----
                        DATA SET SELECTION ENTRY PANEL                Page 1 of 5
Command ==>>>

For a Data Set List, Select Source of Generated List . . 2 (1 or 2)

  1 Generate from a Saved List          Query Name To
    List Name . . .                    Save or Retrieve

  2 Generate a new list from criteria below
    Data Set Name . . . **
    Enter "/" to select option          Generate Exclusive list
    Specify Source of the new list . . 2 (1 - VTOC, 2 - Catalog)
    1 Generate list from VTOC
      Volume Serial Number . . .        (fully or partially specified)
      Storage Group Name . . .          (fully specified)
    2 Generate list from Catalog
      Catalog Name . . .
      Volume Serial Number . . .        (fully or partially specified)
      Acquire Data from Volume . . . . . N (Y or N)
      Acquire Data if DFSMSHsm Migrated . . N (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 24. Selecting Specified Data Sets Using ISMF

2. Enter the DOWN command three times to display page 4 of the Data Set Selection Entry Panel. Select further criteria with the specifications shown in Figure 25.

```

Panel Defaults Utilities Scroll Help
-----
                        DATA SET SELECTION ENTRY PANEL                Page 4 of 5
Command ==>>>

To limit the List, Specify a value or range of values in any of the following:
                                Rel Op  Value   Value   Value   Value
                                -----  -----  -----  -----  -----
Allocation Unit . . . . .
CF Cache Set Name . . . . .
CF Monitor Status . . . . .
CF Status Indicator . . . . .
Change Indicator . . . . .
Compressed Format . . . . .
Data Class Name . . . . .
Data Set Environment . . . . .
Data Set Name Type . . . . .
Data Set Organization  eq    psu     dau     isu     pou
(1 to 8 Values)      ==>>    is
DDM Attributes . . . . .
Device Type . . . . .
(1 to 8 Values)      ==>>
Use ENTER to Perform Selection; Use UP/DOWN Command for other Selection Panel;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 25. Identifying Unmovable Data Sets

When you complete your selection, a list of the data sets that match the selection criteria displays. Figure 26 on page 75 shows a sample list of data sets that match the data set organization criteria.

```

Panel List Dataset Utilities Scroll Help
-----
                                DATA SET LIST
Command ==>                                Scroll ==> HALF
                                           Entries 1-10 of 10
Enter Line Operators below:                Data Columns 6-9 of 39

  LINE          COMPRESSED  % USER DATA  NUM  ALLOC
  OPERATOR      DATA SET NAME  FORMAT  REDUCTION  EXT  UNIT
  ---(1)-----  ---(2)-----  ---(6)---  ---(7)---  (8)  --(9)--
          TEST.SOURCE.DAU1      YES          50      5  BLK
          TEST.SOURCE.DAU2      NO           --      1  TRK
          TEST.SOURCE.DAU3      ---          --      1  TRK
          TEST.SOURCE.DAU4      ---          --      4  TRK
          TEST.SOURCE.DAU5      YES          40      2  TRK
          TEST.SOURCE.ISEQ1     ---          --      1  CYL
          TEST.SOURCE.ISEQ2     ---          --      1  CYL
          TEST.SOURCE.POU       NO           --      1  TRK
          TEST.SOURCE.PSU1      ---          --      1  CYL
          TEST.SOURCE.PSU2      YES          50      1  TRK
-----  -----  -----  -----  -----  -----
                                BOTTOM OF DATA

```

Figure 26. ISMF List of ISAM and Unmovable Data Sets by DSORG

Enter the RIGHT command to display the data set organization column as shown in Figure 27.

```

Panel List Dataset Utilities Scroll Help
-----
                                DATA SET LIST
Command ==>                                Scroll ==> HALF
                                           Entries 1-10 of 10
Enter Line Operators below:                Data Columns 6-9 of 39

  LINE          NUM  ALLOC  SEC  DS  REC
  OPERATOR      EXT  UNIT  ALLOC  ORG  FMT
  ---(1)-----  (8)  --(9)--  --(10)--  (11) (12)
          TEST.SOURCE.DAU1      1  TRK      10  DAU  FB
          TEST.SOURCE.DAU2      1  TRK      10  DAU  FB
          TEST.SOURCE.DAU3      1  TRK      10  DAU  FB
          TEST.SOURCE.DAU4      1  TRK      10  DAU  FB
          TEST.SOURCE.DAU5      1  CYL      0  DAU  FB
          TEST.SOURCE.ISEQ1     1  CYL      0  IS  FB
          TEST.SOURCE.ISEQ2     1  CYL      0  IS  FB
          TEST.SOURCE.POU       1  TRK      10  POU  FB
          TEST.SOURCE.PSU1      1  CYL      1  PSU  FB
          TEST.SOURCE.PSU2      1  TRK      10  PSU  FB
-----  -----  -----  -----  -----
                                BOTTOM OF DATA

```

Figure 27. ISMF Data Set List

3. Repeat the process with the specifications shown in Figure 28 on page 76.

```

Panel Defaults Utilities Scroll Help
-----
                        DATA SET SELECTION ENTRY PANEL                Page 4 of 5
Command ==>

To limit the List, Specify a value or range of values in any of the following:
                                Rel Op  Value      Value      Value      Value
                                -----  -
Allocation Unit . . . .
CF Cache Set Name . . .
CF Monitor Status . . .
CF Status Indicator . .
Change Indicator . . . .
Compressed Format . . .
Data Class Name . . . .
Data Set Environment . .
Data Set Name Type . . .
Data Set Organization   eq      abs
(1 to 8 Values)        ==>     is
DDM Attributes . . . .
Device Type . . . . .
(1 to 8 Values)        ==>

Use ENTER to Perform Selection; Use UP/DOWN Command for other Selection Panel;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 28. Identifying Absolute Track Allocation using ISMF

When you complete your selection, a list of the data sets that match the selection criteria displays. Figure 29 shows a sample list of data sets that match the allocation unit criteria.

```

Panel List Dataset Utilities Scroll Help
-----
                        DATA SET LIST
Command ==>
                                Scroll ==> HALF
                                Entries 1-4 of 4
                                Data Columns 6-9 of 39
Enter Line Operators below:

LINE          COMPRESSED  % USER DATA  NUM  ALLOC
OPERATOR      DATA SET NAME  FORMAT        REDUCTION    EXT  UNIT
---(1)---    -----(2)-----  ---(6)---    ---(7)---    (8)  --(9)--
              MANUF01.SUBS.TCASE1  NO           --      5  ABS
              MANUF01.SUBS.TCASE2  NO           --      1  ABS
              MANUF01.SUBS.TCASE3  ---         --      1  ABS
              MANUF01.SUBS.TCASE4  NO           --      4  ABS
-----  -----  -----  BOTTOM OF DATA  -----

```

Figure 29. ISMF List of Unmovable Data Sets by ALLOC UNIT

## Making Unmovable Data Sets Eligible for System Management

Once you have identified the ineligible data, you can do the following:

- Convert ISAM data sets to VSAM key-sequenced data sets (KSDS). Programs that access the converted ISAM data sets can then do so in VSAM format using the VSAM/ISAM Interface Program (IIP).

**Related Reading:** For more information about converting ISAM data sets to VSAM key-sequenced data sets, see *z/OS DFSMS Using Data Sets*.

- Identify unmovable data sets that can be converted to movable data sets. Move these data sets to system-managed volumes. Pool the remaining non-system-managed volumes containing the unmovable data sets.

---

## Using ISMF to Manage Storage Devices

ISMF helps you perform typical storage administrator tasks interactively. In addition to simplifying these tasks, ISMF interacts with the devices in your storage configuration to prevent errors.

The ISMF Volume Application can help you do the following:

- Set up and manage cache-capable 3990 storage controls with extended functions  
Using ISMF reduces your need to learn access method services syntax and interpret output from access method services functions that control the extended functions, such as DASD fast write and dual copy. The SETCACHE line operator allows you to display and alter the status of all cache-capable 3990 storage control functions, without using operator commands.
- Initialize DASD volumes  
Use the INIT line operator to initialize volumes in online or offline mode.
- Perform problem determination for DASD devices  
Use the ANALYZE line operator to help you diagnose media errors.

---

## Implementing a System-Determined Block Size

During allocation, DFSMSdfp assists you to assign a block size that is optimal for the device. When you allow DFSMSdfp to calculate the block size for the data set, you are using a *system-determined* block size. System-determined block sizes can be calculated for both system-managed and non-system-managed primary storage, VIO, and tape data sets. Use of system-determined block size for tape data sets is discussed in Chapter 11, “Optimizing Tape Usage,” on page 177.

These data sets must have physical sequential, partitioned, or partitioned extended organization and fixed- or variable-length record formats. Unmovable or BDAM data sets are not supported.

Using system-determined block size provides the following benefits:

- Frees data sets from device dependence  
You do not need to know the track capacity to allocate efficiently.
- Eases migration to new devices with different architectures  
No changes to existing JCL are required when installing new devices.
- Optimizes space usage  
The calculated block size attempts to fully use the available track.
- Improves I/O performance  
Although the block size is optimized for space usage, performance benefits can be realized for sequential access.
- Positions you to use sequential data striping  
Striped data sets contain control information in the physical block that might make user-determined block sizes inefficient. You can let the system determine an optimal block size for these data sets.
- Simplifies allocation  
Eliminates the need to specify the BLKSIZE parameter.

Before beginning to implement system-determined block size, you must:

- Understand the external parameters that control its usage
- Determine the space savings

- Understand the planning considerations for its use

## How System-Determined Block Size Works

If you request a system-determined block size, the system calculates the optimal block size, based on the logical record length (LRECL), record format (RECFM), and data set organization (DSORG) requested, and the geometry of the device selected for allocation. Each of these DCB parameters must be available from the program or JCL for the system to determine a block size. You omit the BLKSIZE or set BLKSIZE to zero to request the system to calculate a block size for the data set. Once the block size is calculated, the data set is indicated as *reblockable* to inform programs supporting system-determined block size that the data set's block size can be recalculated if the data set is moved. Also, if the Data Class 'Forced System Determined Blocksize' attribute is specified, then the data set will use a system-determined block size only when there is no block size specified on the JCL DD statement DCB, DCBE and in the DCB OPEN exit routine.

You can cause existing data sets to use a system-determined block size by copying them using DFSMSdss and specifying the REBLOCK parameter or assign a Data Class with 'Force System Determined Blocksize' attribute set to Y. You can modify the DFSMSdss reblock installation exit to have DFSMSdss mark the data set as reblockable. Or, you can use DFSMShsm to allocate the data set using a system-determined block size by migrating and recalling the data set. You must specify SETSYS CONVERSION in DFSMShsm PARMLIB to enable this support. If you do not have these components installed, you can use the DCB OPEN user or installation exit to implement system-determined block size.

## Determining the Space Savings

You can use the ISMF Data Set Application to estimate how much space is being wasted because of inefficient block size usage, by comparing the data set's block size to the optimal block size that DFSMSdfp calculates and ISMF displays. The ISMF column, blocks unused, shows you how much space you can reclaim by using a system-determined block size. ISMF also displays the reblockable indicator for the data set.

Another way is to run the DCOLLECT access method services function.

Before using system-determined block size for your data sets, evaluate the effect on your applications. Applications that specify block size or data sets with very large logical record sizes might present implementation problems. Also, additional virtual storage might be required in the application's address space to store the larger buffers.

---

## Chapter 4. Activating the Storage Management Subsystem

This chapter presents the steps necessary to define and activate a minimal SMS configuration. This establishes an operating environment for the Storage Management Subsystem, but no data sets become system-managed.

---

### Activating a Minimal Configuration

Activating a minimal configuration lets you experience managing an SMS configuration without affecting your JCL or data set allocations. During this phase, you can learn about and gain experience with the following:

- ISMF applications for the storage administrator  
You use ISMF applications to define and activate a minimal SMS configuration.
- The new operator commands that control operation of resources controlled by SMS  
New operating system commands are used to activate an SMS configuration and to display and control storage group and volume status.
- How the SMS base configuration can affect allocations for non-system-managed data sets  
The base configuration for your minimal SMS contains installation defaults for data sets:
  - For non-system-managed data sets, you can specify default device geometry to ease the conversion from device-dependent space calculations to the device-independent method implemented by SMS.
  - For system-managed data sets, you can specify a default management class to be used for data sets that are not assigned a management class by your management class ACS routine.
- Using simplified JCL  
With the minimal configuration active, you can use simplified JCL.
- Implementing allocation standards  
With the minimal configuration active, you can develop a data class ACS routine to enforce your standards.

---

### Managing Data with a Minimal SMS Configuration

The minimal SMS configuration consists of the following elements:

- A base configuration
- A storage class definition
- A storage group containing at least one volume
- A storage class ACS routine
- A storage group ACS routine

All of these elements are required for a valid SMS configuration, except for the storage class ACS routine.

**Recommendation:** Ensure that a storage class ACS routine is part of your minimal configuration. This prevents users from externally specifying a storage class on their DD statements, causing the data set to be system-managed before you are ready.

The storage class ACS routine ensures that the storage class read/write variable is always set to null. The storage group ACS routine is never run because no data sets are system-managed. So, no data sets are allocated as system-managed by the minimal configuration.

Users become familiar with the device-independent space allocation implemented by SMS facilities supported by the minimal configuration. Specifically, the base configuration contains a default unit that corresponds to a DASD esoteric (such as SYSDA). Default geometry for this unit is specified in bytes/track and tracks/cylinder for the predominant device type in the esoteric. If users specify the esoteric, or do not supply the UNIT parameter for new allocations, the default geometry converts space allocation requests into device-independent units, such as KBs and MBs. This quantity is then converted back into device-dependent tracks based on the default geometry.

Follow these steps to activate the minimal configuration:

1. Allocate SMS control data sets.
2. Define Global Resource Serialization (GRS) resource names for active SMS control data sets.
3. Define the system group.
4. Define a minimal SMS configuration.
5. Define the SMS to z/OS.
6. Activate the SMS.
7. Control SMS processing
8. Use simplified JCL to allocate data sets.
9. Enforce standards.

To use SMS effectively, use the information in this chapter and in *z/OS DFSMSdfp Storage Administration*.

---

## Planning to Activate a Minimal SMS Configuration

You can use the starter set along with the sample ACS routines for the Activating the Storage Management Subsystem milestone in Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243 to help you develop a minimal configuration. The examples in this chapter can also help you create your minimal configuration.

Coordinate with your z/OS systems programming group and operations to make z/OS changes required for SMS activation, and schedule an IPL. There are required changes for PARMLIB and GRS definitions.

---

## Allocating SMS Control Data Sets

SMS stores its class and group definitions, translated ACS routines, and system information in the following three control data sets:

- The **SCDS** contains a set of SMS classes and groups and translated ACS routines that implement a specific set of storage management policies.

This is the source control data set, the SMS configuration that you develop and test. You can have several SCDSs. One should correspond to your current policies. Retain at least one prior configuration should you need to regress to it because of error. The SCDS is never used to manage allocations.

- The **ACDS** is the system's active copy of the current SCDS.



**Recommendation:** Ensure that you have extra ACDSs in case a hardware failure causes the loss of your primary ACDS. It must reside on a shared device, accessible to all systems, to ensure that they share a common view of the active configuration. Do not have the ACDS reside on the same device as the COMMDS or SCDS. Both the ACDS and COMMDS are needed for SMS operation across the complex. Separation protects against hardware failure. You should also create a backup ACDS in case of hardware failure or accidental data loss or corruption.

- The **COMMDS** enables communication of configuration changes between SMS systems in a multisystem environment.

It contains the name of the ACDS containing the active storage management policy, and storage group volume statistics. The COMMDS must reside on a shared device accessible to all systems. However, do not allocate it on the same device as the ACDS. Create a spare COMMDS in case of a hardware failure or accidental data loss or corruption. SMS activation fails if the COMMDS is unavailable.

## Calculating the SCDS and ACDS Sizes

The size of constructs within the configuration can support 32 systems. Be sure to allocate sufficient space for the ACDS and SCDS, since insufficient ACDS size can cause errors such as failing SMS activation.

**Related Reading:** For the formula used to calculate the appropriate SMS control data set size, see *z/OS DFSMSdfp Storage Administration*.

## Calculating the COMMDS Size

When you calculate the size of a COMMDS, you have to account for both system and volume information. With SMS 32-name support, the amount of space required for a COMMDS increased. A previously allocated COMMDS might have insufficient space to support the changes. You might need to allocate a new COMMDS prior to activating SMS on a current level system.

**Recommendation:** Always review the COMMDS size when migrating from prior DFSMS releases.

Use the formula described in *z/OS DFSMSdfp Storage Administration* to precisely calculate the COMMDS size.

## Defining the Control Data Sets

After you have calculated their respective sizes, define the control data sets to z/OS using access method services. The job in Figure 30 on page 82, is an example of how to define these data sets. Because these data sets are allocated before SMS is activated, space is allocated in tracks. Allocations in KBs or MBs are only supported when SMS is active.

## Example: Allocating Control Data Sets

---

```
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER(NAME(YOUR.OWN.SCDS) LINEAR VOLUME(D65DM1) -
              TRK(25 5) SHAREOPTIONS(2,3)) -
              DATA(NAME(YOUR.OWN.SCDS.DATA))

DEFINE CLUSTER(NAME(YOUR.OWN.ACDS) LINEAR VOLUME(D65DM2) -
              TRK(25 5) SHAREOPTIONS(3,3)) -
              DATA(NAME(YOUR.ACDS.DATA))

DEFINE CLUSTER(NAME(YOUR.OWN.COMMDS) LINEAR VOLUME(D65DM3) -
              TRK(1 1) SHAREOPTIONS(3,3)) -
              DATA(NAME(YOUR.OWN.COMMDS.DATA))
```

---

Figure 30. Sample Job for Allocating Control Data Sets

Specify SHAREOPTIONS(2,3) only for the SCDS. This lets one update-mode user operate simultaneously with other read-mode users between regions.

Specify SHAREOPTIONS(3,3) for the ACDS and COMMDS. These data sets must be shared between systems that are managing a shared DASD configuration in a DFSMS environment.

---

## Defining GRS Resource Names for SMS Control Data Sets

As you prepare to allocate the SMS control data sets, consider the effects of multiple systems sharing these data sets.

Between systems, the data sets are serialized by RESERVE/RELEASE. If you allocate a data set on a device with other data sets protected by RESERVE/RELEASE, it locks out all activity from other systems to that volume.

**Recommendation:** Ensure that you use global resource serialization (GRS) to convert the reserve into a global ENQ.

A reserve is issued while updating the control data sets for the following reasons:

- The COMMDS is updated with space statistics at the expiration of the time interval specified in the IGDSMSxx member in PARMLIB.
- The ACDS is updated whenever a configuration change occurs, such as when an operator varies a volume offline.

You should place resource name IGDCDSXS in the RESERVE conversion RNL as a generic entry. This minimizes delays due to contention for resources and prevents deadlocks associated with the VARY SMS command.

Make sure to convert the RESERVEs to global ENQs. Using RESERVE/RELEASE can result in deadlocks even if the COMMDS and ACDS data sets are the only ones on separate volumes.

**Requirement:** If you have multiple SMS complexes within a global resource serialization complex, be sure to use unique COMMDS and ACDS data set names to prevent false contention.

## Defining a Minimal SMS Configuration

Figure 31 shows that a valid SMS minimal configuration must contain the following:

- A fully-defined base configuration
- A storage class definition
- A storage group containing at least one volume
- A storage class ACS routine
- A storage group ACS routine

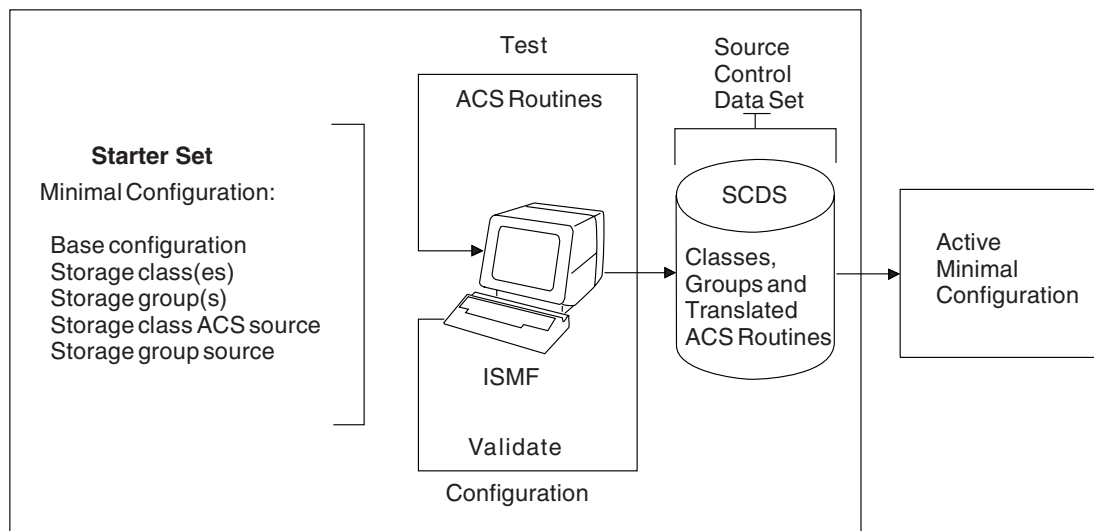


Figure 31. Minimal SMS Configuration

**Tip:** The volume does not have to exist as long as you do not direct allocations to either the storage group or the volume.

Because a storage class can be assigned using a JCL DD statement, a storage class ACS routine is not required in a valid SCDS.

**Recommendation:** Define a storage class ACS routine so that users do not attempt to use the `STORCLAS` keyword in JCL. The base configuration consists of the names for each system or system group in the SMS complex, the default management class, and the default device geometry and unit information.

The following sections describe the steps you should follow to define a minimal configuration. The figures show the sample classes, groups, base configuration, and ACS routines provided in either the starter set or Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243.

## Defining the SMS Base Configuration

Use ISMF to create your SCDS base. The starter set configuration can be used as a model for your own SCDS. The panels displayed in this section reflect data from this sample configuration.

**Related Reading:** For a detailed description of base configuration attributes and how to use ISMF to define its contents, see *z/OS DFSMSdfp Storage Administration*.

Follow these steps to create the SCDS base:

1. Enter an 8 (Control Data Set) on the ISMF Primary Option Menu to view the CDS Application Selection panel shown in Figure 32.

**Requirement:** You must have the correct RACF authorization. If your ISMF menu does not display option 8 (Control Data Set), select option 2 to define yourself as a storage administrator. You might need to exit ISMF and ISPF for the change to take effect.

```

Panel  Utilities  Help
-----
                                CDS APPLICATION SELECTION
Command ==>>

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

Select one of the following Options:
  1. Display      - Display the Base Configuration
  2. Define       - Define the Base Configuration
  3. Alter        - Alter the Base Configuration
  4. Validate     - Validate the SCDS
  5. Activate     - Activate the CDS
  6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
  7. Cache Update - Define/Alter/Delete CF Cache Sets
  8. Lock Display  - Display CF Lock Structure Names for all CF Lock Sets
  9. Lock Update  - Define/Alter/Delete CF Lock Sets
If CACHE Display is chosen, Enter CF Cache Set Name . . *
If LOCK Display is chosen, Enter CF Lock Set Name . . . *
                                (1 to 8 character CF cache set name or * for all)
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 32. CDS Application Selection Panel

2. In the CDS Name field, type in the name of the SCDS that is to contain the base configuration. In this example, the CDS name is YOUR.OWN.SCDS. Enter a 2 (Define) to view the SCDS Base Define panel shown in Figure 33.

```

Panel  Utilities  Scroll  Help
-----
                                SCDS BASE DEFINE                                Page 1 of 2
Command ==>>

SCDS Name . . : YOUR.OWN.SCDS
SCDS Status : VALID

To DEFINE SCDS Base, Specify:

Description ==>> This is the configuration used to implement a
                ==>> minimal configuration

Default Management Class . . STANDEF (1 to 8 characters)
Default Unit . . . . . SYSDA (esoteric or generic device name)
Default Device Geometry
  Bytes/Track . . . . . 56664 (1-999999)
  Tracks/Cylinder . . . . . 15 (1-999999)
DS Separation Profile (Data Set Name)
==> _____

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 33. SCDS Base Define Panel, Page 1 of 2

Use the DOWN command to view the second page as shown in Figure 34.

```
Panel Utilities Scroll Help
-----
                                SCDS BASE DEFINE                                Page 2 of 2
Command ==>>

SCDS Name . . : YOUR.OWN.SCDS
SCDS Status : VALID

Specify one of the following options . . 1          (1 Add, 2 Delete, 3 Rename)

  Specify System Name . . . . . SYSTEM6   or Sys Group Name . .

  New System/Sys Group Name . .          (For option 3, Rename)

System: SYSTEM1 SYSTEM2 SYSTEM3 SYSTEM4 SYSTEM5 SYSTEM6 SYSTEM7 SYSTEM8

Sysgrp: SYSPLX01 SYSPLX02

Use ENTER to Perform Option; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL To Exit.
```

Figure 34. SCDS Base Define Panel, Page 2 of 2

The SCDS name is the same as the value that you specified on the CDS Application Selection panel (see Figure 32 on page 84).

3. Define a default management class and type it in the Default Management Class field. In this example, we have used the STANDEF management class, a management class in the sample configuration.

The default management class is only used when DFSMShsm performs automatic processing for those data sets that do not have management classes assigned to them. When no management class is assigned to a data set, the catalog entry for that data set contains no management class, even though the default management class controls its backup and availability. You should periodically search for data sets that are system-managed and have no management class assigned. DFSMSdss's filtering capabilities can be used to identify system-managed data sets with no management class assigned, and to produce a report containing these management class exceptions.

4. You should set the value in the Default Unit field to your system's primary esoteric name. For Default Device Geometry, specify values for the Bytes/Track and Tracks/Cylinder attributes. The values for the 3380 are 47476 and 15, respectively. For the 3390, the values are 56664 and 15, respectively.

You should indicate the characteristics of your predominant device as the characteristics for the default unit. If your configuration contains 90% 3390-2s and 10% 3380-Ks, then specify the 3390 geometry characteristics as the default device geometry.

The JCL UNIT parameter is optional for new data set allocations for both system-managed and non-system-managed data sets. SMS uses the Default Unit attribute if no unit is specified when allocating non-system-managed data sets. The Default Device Geometry attribute converts an allocation request from tracks or cylinders to KBs or MBs when an esoteric unit is used, or when no unit is given. Through this conversion, uniform space can be allocated on any device type for a given allocation.

The space request is always converted to KBs or MBs according to the following formula:

$$\text{tracks allocated} = \frac{(\# \text{ tracks} \times (\text{bytes}/\text{track}))}{\text{track capacity}}$$

Where:

- *bytes/track* is derived from the Default Device Geometry attribute.
- *track capacity* is the capacity of the device selected, including device overhead.
- The result of the calculation, *tracks allocated*, is rounded up.

This change can affect your user's existing JCL that specifies the UNIT parameter. There are two variations of UNIT coding:

- Users specify a generic name, such as 3380 or 3390:  
These users have allocation converted to bytes, based on the geometry of that device.
- Users specify an esoteric name, such as SYSDA:  
These users have allocation converted to bytes, based on the Default Device Geometry attribute.  
Use an esoteric name for a more consistent amount of allocated space. It provides a transition for users to allocation in the system-managed storage environment.

5. If you have created a data set separation profile, use the optional field DS Separation Profile to provide SMS with the name of the profile. During volume selection for data set allocation, SMS attempts to separate, on the PCU or volume level, the data sets that are listed in the data set separation profile.

You can specify any valid sequential or partitioned member data set name, with a maximum length of 56 characters, with or without quotation marks. For data set names without quotation marks, ISMF will add the TSO user ID prefix of the person who is defining or updating the base configuration.

The default value is **blank**, which indicates that data set separation is not requested for the SMS complex.

**Recommendation:** Use data set separation for a small set of mission critical data only.

**Related Reading:** To learn how to create a data set separation profile, see Using Data Set Separation in *z/OS DFSMSdfp Storage Administration*.

6. Use the System Name or Sys Group Name and New System/Sys Group Name fields to define the SMS system or system group. You can define up to 32 system names and system group names in the SCDS base, depending on whether you are running in compatibility or 32-name mode. When the system is running in compatibility mode, you can only define up to eight system or system group names. When the system is running in 32-name mode, you can define up to 32 system names or system group names, as follows:

- The system name must match the GRS system ID.

**Related Reading:** For a description of the GRS SYSNAME parameter in the IEASYSxx member of the SYS1.PARMLIB, see *z/OS MVS Initialization and Tuning Reference*.

- The system group name must match the name of the Parallel Sysplex. The Parallel Sysplex name is defined in the COUPLExx member of the PARMLIB of each of the systems that belong to the Parallel Sysplex.

**Restriction:** In a JES3 environment, system group name is not supported due to the fact that the system features and resources that are used to determine on which eligible system a job will be run are only applicable to each system, not to a system group.

## Defining the System Group

A system group is a group of systems within an SMS complex that have similar connectivity to storage groups, libraries, and volumes. When a Parallel Sysplex name is specified and used as a system group name, the name applies to all systems in the Parallel Sysplex except for those systems defined as part of the Parallel Sysplex that are explicitly named in the SMS base configuration. For example, suppose your Parallel Sysplex is named SYSPLEX1 and consists of 12 systems, named SY1 through SY12 (see Table 5). All SMS operations carried out on a name of SYSPLEX1 apply to SY1 through SY12. As another example, suppose the SMS base configuration consists of two names, SYSPLEX1 and SY1. All SMS operations carried out on a name of SYSPLEX1 apply only to SY2 through SY12. Operations carried out on a name of SY1 apply to SY1.

Table 5. The SMS View of Parallel Sysplexes - Examples

Example	Parallel Sysplex Name	Explicit System Name	Systems in Complex	SMS View
1	SYSPLEX1	none	SY1 – SY12	SYSPLEX1 (as a system group); no systems individually defined
2	SYSPLEX1	SY1	SY1 – SY12	SYSPLEX1 (as a system group); SY1 is individually defined

## Defining SMS Classes and Storage Group

To define a minimal configuration, define a storage class and storage group, and create their respective ACS routines.

**Recommendation:** Defining a data class and a management class and creating their respective ACS routines are not required for a valid SCDS. However, because the default management class is so important, it should be included in your minimal configuration.

**Related Reading:** For detailed information on defining SMS classes and groups using ISMF, see *z/OS DFSMSdfp Storage Administration*.

### Defining the Storage Class

You must define at least one storage class name to SMS. Because a minimal configuration does not include any system-managed volumes, no performance or availability information need be contained in the minimal configuration's storage class. Specify an artificial storage class, SC1. This class is later used by the storage administrator to create non-system-managed data sets on an exception basis. In the storage class ACS routine, &STORCLAS is set to a null value to prevent users from coding a storage class in JCL before you want to have system-managed data sets.

You can define the class, SC1, in your configuration in one of two ways:



1. Define the class using the define option of the ISMF storage class application.
2. Use the ISMF COPY line operator to copy the definition of SC1 from the starter set's SCDS to your own SCDS.

To define the storage class using the first option:

1. Enter 5 (Storage Class) on the ISMF Primary Option Menu to view the Storage Class Application Selection panel, shown in Figure 35.

```

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

To perform Storage Class Operations, Specify:
CDS Name . . . . . 'USER6.MYCDS'
                                (1 to 44 character data set name or 'Active' )
Storage Class Name . . SC1      (For Storage Class List, fully or
                                partially specified or * for all)

Select one of the following options :
1  1. List      - Generate a list of Storage Classes
2  2. Display   - Display a Storage Class
3  3. Define    - Define a Storage Class
4  4. Alter     - Alter a Storage Class
5  5. Cache Display - Display Storage Classes/Cache Sets
6  6. Lock Display - Display Storage Classes/Lock Sets

If List Option is chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria
If Cache Display is Chosen, Specify Cache Structure Name . .
If Lock Display is Chosen, Specify Lock Structure Name . . .
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 35. Storage Class Application Selection Panel

2. Give values for the CDS Name and a Storage Class Name fields. The CDS Name must be the same name that you gave for the SCDS on the CDS Application Selection panel (see Figure 32 on page 84). In this example, the CDS name is USER6.TEST.SCDS and the storage class name is SC1.  
Enter 3 (Define) to display the Storage Class Define panel, shown in Figure 36 on page 89.

```

Panel Utilities Scroll Help
-----
                                STORAGE CLASS DEFINE                                Page 1 of 2
Command ==>>

SCDS Name . . . . . : USER6.TEST.SCDS
Storage Class Name : SC1
To DEFINE Storage Class, Specify:
  Description ==>
    ==>
Performance Objectives
  Direct Millisecond Response . . . . . (1 to 999 or blank)
  Direct Bias . . . . . (R, W or blank)
  Sequential Millisecond Response . . . . . (1 to 999 or blank)
  Sequential Bias . . . . . (R, W or blank)
  Initial Access Response Seconds . . . . . (0 to 9999 or blank)
  Sustained Data Rate (MB/sec) . . . . . (0 to 999 or blank)
  OAM Sublevel . . . . . (1, 2 or blank)
  Availability . . . . . N (C, P ,S or N)
  Accessibility . . . . . N (C, P ,S or N)
  Backup . . . . . (Y, N or Blank)
  Versioning . . . . . (Y, N or Blank)
Use ENTER to Perform Verification; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 36. Storage Class Define Panel, Page 1 of 2

3. SCDS Name and Storage Class Name are output fields containing the values that you specified on the Storage Class Application Selection panel (see Figure 35 on page 88). Description is an optional field of 120 characters where you can describe the storage class.
4. For the minimal configuration, allow other storage class attributes to default to the ISMF values. Do not specify Y for the Guaranteed Space attribute to avoid allocation failures on specified volumes. You can specify it later for such data sets as IMS online logs, DB2 online logs, or DB2 BSDS.  
Press Enter to verify the attributes. Enter END on the command line or press PF3 to exit this panel.

To define the storage class by copying the definition from the sample base configuration:

1. List the storage classes in the base configuration, as shown in Figure 37 on page 90.

```

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

To perform Storage Class Operations, Specify:
CDS Name . . . . . 'USER6.MYCDS'
                        (1 to 44 character data set name or 'Active' )
Storage Class Name . . NONSMS (For Storage Class List, fully or
                        partially specified or * for all)

Select one of the following options :
1  1. List           - Generate a list of Storage Classes
   2. Display        - Display a Storage Class
   3. Define         - Define a Storage Class
   4. Alter          - Alter a Storage Class
   5. Cache Display  - Display Storage Classes/Cache Sets
   6. Lock Display   - Display Storage Classes/Lock Sets

If List Option is chosen,
Enter "/" to select option      Respecify View Criteria
                                Respecify Sort Criteria

If Cache Display is Chosen, Specify Cache Structure Name . .
If Lock Display is Chosen, Specify Lock Structure Name . . .
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 37. Listing Storage Classes Defined in the Base Configuration

- View the storage classes in the base configuration and enter COPY as a line operator on the line describing the SC1 class, as shown in Figure 38.

```

Panel  List Utilities Scroll Help
-----
                        STORAGE CLASS LIST
Command ==>>>
                                Scroll ==>>> HALF
                                Entries 34-43 of 117
                                Data Columns 3-7 of 18

CDS Name . : USER6.MYCDS

Enter Line Operators below:

LINE   STORCLAS DIR RESP DIR  SEQ RESP SEQ
OPERATOR NAME   (MSEC) BIAS (MSEC) BIAS AVAILABILITY
---(1)--- --(2)--- --(3)--- (4)- --(5)--- (6)- -----(7)-----
          CRITICAL  10 -      10 - CONTINUOUS
          DBCRIT    10 W      10 R CONTINUOUS
          FAST      5 -      5 - STANDARD
          FASTREAD  5 R      5 R STANDARD
          FASTWRIT  5 W      5 W STANDARD
          GSPACE    --- -      --- - STANDARD
          MEDIUM   10 -      10 R STANDARD
COPY    NONSMS     --- -      --- - STANDARD
        NONVIO    --- -      --- - STANDARD
        STANDARD  --- -      --- - STANDARD

```

Figure 38. Initiating a Copy for the SC1 Storage Class

- Copy the SC1 storage class from the base configuration to your own SCDS, as shown in Figure 39 on page 91. Enter the name of your SCDS in the Data Set Name field, and "SC1" in the Construct Name field of the "Copy To" data area. Press Enter. The SC1 SAVED message indicates that the storage class has been successfully copied.

```

Panel  Utilities  Help
-----
                                COPY ENTRY PANEL
Command ==>>

Construct will be Copied from:

Data Set Name . . . : IBMUSER.BASE.SCDS
Construct Name  . . : NONSMS
Construct Type  . . : STORAGE CLASS

Specify "Copy To" Construct:

Data Set Name . . . 'USER6.MYCDS'
                                (1 to 46 Characters)

Construct Name  . . SC1
                                (If Copy Pool - 1 to 23 characters, fully specified)
                                (Others - 1 to 8 characters, fully specified)
Enter "/" to select option  _  Replace like-named Construct
                               _  Perform Alter
                               _  Copy Storage Group Volumes (Pool SG only)

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

```

Figure 39. Copying the SC1 Storage Class Construct

## Defining the Storage Group

You must define at least one pool storage group name to SMS, and at least one volume serial number to this storage group. (A storage group with no volumes defined is not valid.) This volume serial should be for a nonexistent volume to prevent the occurrence of JCL errors from jobs accessing data sets using a specific volume serial number.

Defining a non-existent volume lets you activate SMS without having any system-managed volumes. No data sets are system-managed at this time. This condition provides an opportunity to experiment with SMS without any risk to your data.

Define the storage group in your SCDS:

1. Enter a 6 (Storage Group) on the ISMF Primary Option Menu to view the Storage Group Application Selection panel, shown in Figure 40 on page 92.

```

Panel Utilities Help
-----
                STORAGE GROUP APPLICATION SELECTION
Command ==>>

To perform Storage Group Operations, Specify:
CDS Name . . . . . 'USER6.MYSCDS'
                                (1 to 44 character data set name or 'Active' )
Storage Group Name . . SG2      (For Storage Group List, fully or
                                partially specified or * for all)
Storage Group Type . . POOL      (VIO, POOL, DUMMY, COPY POOL BACKUP
                                OBJECT, OBJECT BACKUP, or TAPE)

Select one of the following options :
 1. List   - Generate a list of Storage Groups
 2. Display - Display a Storage Group (POOL, OBJECT or TAPE only)
 3. Define - Define a Storage Group
 4. Alter  - Alter a Storage Group
 5. Volume - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
  Enter "/" to select option      Respecify View Criteria
                                Respecify Sort Criteria

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

```

Figure 40. Defining a Storage Group for the Minimal Configuration

- Specify values for the CDS Name and Storage Group Name fields. The CDS name must be the same name that you specified for the SCDS on the CDS Application Selection panel (see Figure 32 on page 84). In this example, the CDS name is USER6.MYSCDS and the storage group name is SG2. Enter a 2 (Define) to display the Pool Storage Group Define panels, shown in Figure 41 and Figure 42 on page 93.

```

Panel Utilities Scroll Help
-----
                POOL STORAGE GROUP DEFINE                Page 1 of 2
Command ==>>

SCDS Name . . . . . : USER6.MYSCDS
Storage Group Name : SG2
To DEFINE Storage Group, Specify:
Description ==>
==>
Auto Migrate . . Y (Y, N, I or P)  Migrate Sys/Sys Group Name . .
Auto Backup . . Y (Y or N)         Backup Sys/Sys Group Name . .
Auto Dump . . . N (Y or N)         Dump Sys/Sys Group Name . . .
Overflow . . . . N (Y or N)        Extend SG Name . . . . .
                                Copy Pool Backup SG Name . . .

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

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

Use ENTER to Perform Selection; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 41. Defining Pool Storage Group Attributes Page 1

```

Panel Utilities Scroll Help
-----
                                POOL STORAGE GROUP DEFINE           Page 2 of 2
Command ==>>>

SCDS Name . . . . . : USER6.MYSCDS
Storage Group Name : SG2

To DEFINE Storage Group, Specify:

Allocation/migration Threshold :   High . . 85 (1-99) Low . . 1 (0-99)
Alloc/Migr Threshold Track-Managed: High . . 85 (1-99) Low . . 1 (0-99)
Guaranteed Backup Frequency . . . . . (1 to 9999 or NOLIMIT)
BreakPointValue . . . . . (0-65520 or blank)
Processing Priority . . . . . (1-100)

Use ENTER to Perform Selection; Use UP Command to View previous Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 42. Defining Pool Storage Group Attributes Page 2

SCDS Name and Storage Group Name are output fields containing the values that you specified on the Storage Group Application Selection panel (see Figure 40 on page 92). Description is an optional field of 120 characters that you can use to describe the storage group type.

If you supply the name of a Parallel Sysplex in the Migrate System Name, Backup System Name, or Dump System Name field, make sure you enter it correctly. Otherwise, the expected operation might not occur.

- Let the storage group attributes default to the ISMF values.

Press Enter to verify and display the SMS Storage Group Status Define panel shown in Figure 43.

```

Panel Utilities Scroll Help
-----
                                SMS STORAGE GROUP STATUS DEFINE       Page 1 of 2
Command ==>>>

SCDS Name . . . . . : USER6.MYSCDS
Storage Group Name : SG2
Storage Group Type : POOL
To DEFINE Storage Group System/
Sys Group Status, Specify:

System/Sys      SMS SG   System/Sys      SMS SG
Group Name      Status   Group Name      Status
-----
MYSYS01 ==>> NOTCON  MYSYS02 ==>> NOTCON
MYSYS03 ==>> NOTCON  MYSYS04 ==>> NOTCON
MYSYS05 ==>> NOTCON  MYSYS06 ==>> NOTCON
MYSYS07 ==>> NOTCON  MYSYS08 ==>> NOTCON
*MYSYS09 ==>> DISALL *MYSYS10 ==>> DISALL
*MYSYS11 ==>> DISALL *MYSYS12 ==>> DISALL
*MYSYS13 ==>> DISALL *MYSYS14 ==>> DISALL
*MYSYS15 ==>> DISALL *MYSYS16 ==>> DISALL

( Possible SMS SG
Status for each:
- Pool SG Type
  NOTCON, ENABLE
  DISALL, DISNEW
  QUIALL, QUINEW
- Tape SG Type
  NOTCON, ENABLE,
  DISALL, DISNEW )
* SYS GROUP = sysplex
minus Systems in the
Sysplex explicitly
defined in the SCDS

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 43. Defining Storage Group System Status

SCDS Name, Storage Group Name, and Storage Group Type are output fields containing the values that you specified on the Storage Group Application Selection panel (see Figure 41 on page 92).

4. The names of your systems, system groups, or both are displayed in the System/Sys Group Name columns. Specify ENABLE in the corresponding SMS SG STATUS column.

Press Enter to verify. When the verification is complete, enter the END command twice to return to the Storage Group Application Selection panel (see Figure 40 on page 92).

5. From this panel, enter a 4 (Volume) to view the Storage Group Volume Selection panel, shown in Figure 44.

```

Panel Utilities Help
-----
                                STORAGE GROUP VOLUME SELECTION
Command ==>>

CDS Name . . . . . : USER6.MYSCDS
Storage Group Name : SG2
Storage Group Type : POOL

Select One of the following Options:
- 1. List           - Generate a list of Storage Groups
- 2. Display        - Display SMS Volume Statuses (POOL, OBJECT or TAPE only)
- 3. Define         - Add Volumes to Volume Serial Number List
- 4. Alter          - Alter Volume Statuses (Pool only)
- 5. Delete         - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
Prefix From To Suffix Hex

==>> SG2           _____ - ('X' in HEX field allows
==>>                                     FROM - TO range to include
==>>                                     hex values A through F.)
==>>

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

```

Figure 44. Defining Non-Existent Volume in Storage Group

CDS Name, Storage Group Name, and Storage Group Type are output fields containing the values you specified on previous panels.

6. Enter a 2 (Define) and specify the appropriate volume serial numbers. Each time you press Enter, you display the SMS Volume Status Define panel, shown in Figure 45 on page 95.



```

Panel Utilities Scroll Help
-----
SMS VOLUME STATUS DEFINE                               Page 1 of 2
Command ==>
SCDS Name . . . . . : USER6.MYSCDS
Storage Group Name . : SG2
Volume Serial Numbers : VOL010 - VOL03

To DEFINE SMS Volume Status, Specify:

System/Sys      SMS Vol      System/Sys      SMS Vol      ( Possible SMS Vol
Group Name      Status        Group Name      Status        Status for each:
-----
MYSYS01 ==> ENABLE  MYSYS02 ==> ENABLE  ( NOTCON, ENABLE,
MYSYS03 ==> ENABLE  MYSYS04 ==> ENABLE  DISALL, DISNEW,
MYSYS05 ==> ENABLE  MYSYS06 ==> ENABLE  QUIALL, QUINEW )
*MYSYS09 ==> ENABLE *MYSYS10 ==> ENABLE  * SYS GROUP = sysplex
*MYSYS11 ==> ENABLE *MYSYS12 ==> ENABLE  minus systems in the
*MYSYS13 ==> ENABLE *MYSYS14 ==> ENABLE  sysplex explicitly
*MYSYS15 ==> ENABLE *MYSYS16 ==> ENABLE  defined in the SCDS
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 45. Defining Volume System Status

SCDS Name, Storage Group Name, and Volume Serial Numbers are output fields containing the values that you entered on the Storage Group Volume Selection panel (see Figure 44 on page 94).

7. Define the relationship between the volume and each system or system group by entering DISALL in the SMS VOL STATUS column next to each name in the System/Sys Group Name column.

### Defining the Default Management Class

Define a default management class and name it STANDEF to correspond with the entry in the base configuration.

**Recommendation:** Specifically assign all system-managed data to a management class. If you do not supply a default, DFSMSHsm uses two days on primary storage, and 60 days on migration level 1 storage, as the default.

No management classes are assigned when the minimal configuration is active. Definition of this default is done here to prepare for use in the Managing Permanent Data milestone.

The management class, STANDEF, is defined in the starter set's SCDS. You can copy its definition to your own SCDS in the same way as the storage class was copied. If you choose to define the default management class:

1. Enter a 3 (Management Class) on the ISMF Primary Option Menu to display the Management Class Application Selection panel (see Figure 46 on page 96).

```

Panel Utilities Scroll Help
-----
MANAGEMENT CLASS APPLICATION SELECTION Page 1 of 2
Command ==>

To perform Management Class Operations, Specify:
CDS Name . . . . . 'USER6.MYSCDS'
                                     (1 to 44 character data set name or 'Active' )
Management Class Name . . . MC1      (For Management Class List, fully or
                                     partially specified or * for all)

Select one of the following options :
3 1. List   - Generate a list of Management Classes
   2. Display - Display a Management Class
   3. Define - Define a Management Class
   4. Alter  - Alter a Management Class

If List Option is chosen,
Enter "/" to select option      Respecify View Criteria
                                Respecify Sort Criteria

Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 46. Defining a Management Class for the Minimal Configuration

- Specify values in the CDS Name and Management Class Name fields. The CDS name must be the same name that you specified for the SCDS on the CDS Application Selection panel (see Figure 32 on page 84). In this example, the SCDS name is YOUR.OWN.SCDS and the management class name is STANDEF.

Enter a 3 (Define) to view the first page of the Management Class Define panel (see Figure 47).

```

Panel Utilities Scroll Help
-----
MANAGEMENT CLASS DEFINE Page 1 of 6
Command ==>

SCDS Name . . . . . : USER6.MYSCDS
Management Class Name : MC1

To DEFINE Management Class, Specify:

Description ==> Installation default management class
              ==> assigned when MC ACS routine assigns no class.

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

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

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 47. Management Class Define Panel, Page 1 of 6

SCDS Name and Management Class Name are output fields containing the values that you specified on the Management Class Application Selection panel (see Figure 46). Description is an optional field of 120 characters that you can use to describe the management class.

- Use the default of NOLIMIT for the Expire after Days Non-usage field, the Expire after Date/Days field, and the Retention Limit field. This means the data set is never eligible for expiration.

For allocations not covered by management classes, you can override the expiration date when deletions are done for system-managed DASD data sets. To do this, use the OVRD\_EXPDT keyword in the IGDSMSxx member of the PARMLIB. This should only be done when management class cannot be used, and it is only for use with tape allocations converted to DASD. You should only use this if you never use expiration dates for DASD data sets.

Scroll down to perform verification and to display the second page of the Management Class Define panel, shown in Figure 48.

```

Panel  Utilities  Scroll  Help
-----
                                MANAGEMENT CLASS DEFINE                                Page 2 of 6
Command ==>

SCDS Name . . . . . : USER6.MYSCDS
Management Class Name : MC1

To DEFINE Management Class, Specify:

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

Migration Attributes
Primary Days Non-usage . . . . 2          (0 to 9999 or blank)
Level 1 Days Non-usage . . . . 15         (0 to 9999, NOLIMIT or blank)
Command or Auto Migrate . . . . BOTH      (BOTH, COMMAND or NONE)

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

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 48. Management Class Define Panel, Page 2 of 6

SCDS Name and Management Class Name are output fields containing the values that you specified on the Management Class Application Selection panel (see Figure 46 on page 96).

- Specify N in the Partial Release field to inhibit DFSMSHsm's space management from releasing allocated but unused space. We specified a short life on primary and migration level 1 for these data sets, to prevent over-commitment of primary and migration level 1 storage. These data sets should be re-assigned a management class that is more appropriate than the default. Specify BOTH in the Command or Auto Migrate field to permit DFSMSHsm, the storage administrator, or the user to manage the data set.

Scroll down to perform the verification and to display the third page of the Management Class Define panel shown in Figure 49 on page 98.

```

Panel  Utilities  Scroll  Help
-----
                                MANAGEMENT CLASS DEFINE                                Page 3 of 6
Command ==>>>

SCDS Name . . . . . : USER6.MYCDS
Management Class Name : MC1

To DEFINE Management Class, Specify:
Backup Attributes
Backup Frequency . . . . . 1          (0 to 9999 or blank)
Number of Backup Vers . . . . . 2      (1 to 100 or blank)
(Data Set Exists)
Number of Backup Vers . . . . . 1      (0 to 100 or blank)
(Data Set Deleted)
Retain days only Backup Ver . . . 60   (1 to 9999, NOLIMIT or blank)
(Data Set Deleted)
Retain days extra Backup Vers . . 30   (1 to 9999, NOLIMIT or blank)
Admin or User command Backup . . BOTH (BOTH, ADMIN or NONE)
Auto Backup . . . . . Y               (Y or N)
Backup Copy Technique . . . . . S      (P, R, S, VP, VR, CP or CR)

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; Cancel to Exit.

```

Figure 49. Management Class Define Panel, Page 3 of 6

SCDS Name and Management Class Name are output fields containing the values that you entered on the Management Class Application Selection panel (see Figure 46 on page 96).

5. In the Backup Attributes fields, specify that a minimal number of backup versions be retained for the data set. Specify BOTH for Admin or User command Backup, Y for Auto Backup to ensure that the data set is retained on both DFSMSHsmbackup volumes and migration level 2, and let Backup Copy Technique default. Leave the remaining fields blank.

## Creating ACS Routines

After you define the SMS classes and group, develop your ACS routines. In the minimal configuration, you assign a null storage class in the storage class ACS routine. The storage group ACS routine is not run if a null storage class is assigned. However, you must code a trivial one to satisfy the SMS requirements for a valid SCDS. After you have written the ACS routines, use ISMF to translate them into executable form.

### Writing the ACS Routines

1. If you do not have the starter set, allocate a fixed-block PDS or PDSE with LRECL=80 to contain your ACS routines. Otherwise, go to the next step.
2. Enter a 7 (AUTOMATIC CLASS SELECTION) on the ISMF Primary Option Menu to display the ACS Application Selection panel shown in Figure 50 on page 99.

```

Panel  Utilities  Help
-----
                                ACS APPLICATION SELECTION
Command ==>

Select one of the following options:
1  1. Edit           - Edit ACS Routine source code
   2. Translate      - Translate ACS Routines to ACS Object Form
   3. Validate       - Validate ACS Routines Against Storage Constructs
   4. Test           - Define/Alter Test Cases and Test ACS Routines
   5. Display        - Display ACS Object Information
   6. Delete         - Delete an ACS Object from a Source Control Data Set

If Display Option is Chosen, Specify:

CDS Name  . . 'YOUR.OWN.SCDS'
                                (1 to 44 Character Data Set Name or 'Active')

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

```

Figure 50. Writing an ACS Routine

3. Select option 1 (Edit), and press Enter to display the Edit-Entry panel shown in Figure 51.

```

Menu  RefList  RefMode  Utilities  Workstation  Help
-----
                                Edit Entry Panel
Command ==>

ISPF Library:
Project . . . IBMUSER
Group . . . . IMPLACTV . . . . .
Type . . . . ACSLIB
Member . . . STORCLAS      (Blank or pattern for member selection list)

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . .
Volume Serial . . .      (If not cataloged)

Workstation File:
File Name . . . . .

Initial Macro . . . . .      Options
Profile Name . . . . .      / Confirm Cancel/Move/Replace
Format Name . . . . .      Mixed Mode
Data Set Password . . . . .  Edit on Workstation
                                Preserve VB record length

```

Figure 51. Accessing the ISPF/PDF Editor

4. Type in the appropriate data set name for the ACS routines. We have shown the name of the PDS or PDSE corresponding to the sample ACS routines for this milestone. The storage class ACS routine is allocated in the STORCLAS member.  
Press Enter to access the ISPF/PDF editor.
5. On this screen, enter the source code for the storage class ACS routine, as shown in Figure 52 on page 100. This routine sets a null storage class and exits. No data is system-managed.

```

Menu  Functions  Utilities  Help
-----
EDIT ---- IBMUSER.IMPLACTV.ACSLIB(STORCLAS) - 01.00 ----- COLUMNS 001 072
COMMAND ==>                                     SCROLL ==> PAGE
***** TOP OF DATA *****
PROC STORCLAS

/*****/
/*          C H A N G E   H I S T O R Y          */
/*          =====                          */
/*  DATE    RESP  DESCRIPTION OF CHANGE          */
/* -----  ----  -----                          */
/* 91/03/24  EG    Initial routine created.        */
/* PURPOSE:  This routine assigns a null storage class to all data sets. */
/* INPUT:    The following ACS variables are referenced: NONE */
/* OUTPUT:   Null storage class.                   */
/* RETURN    Zero is the only return code. Data set allocations are */
/* CODES:    not failed in this routine.           */
/*****/

SET &STORCLAS = ''

END /* END OF STORAGE CLASS ROUTINE PROC */
***** BOTTOM OF DATA *****

```

Figure 52. Sample Storage Class ACS Routine for the Minimal Configuration

6. Enter the END command to save the routine and return to the Edit-Entry panel shown in Figure 51 on page 99.
7. Enter the name of your storage group ACS routine as the new member name in the Member field. The sample routine uses the name STORGRP. The ISPF/PDF editor appears again.
8. On this screen, enter the source code for the storage group ACS routine, as shown in Figure 53 on page 101. This source code assigns the previously defined storage group. Because this particular storage group contains a non-existent volume, no volumes are system-managed.

```

Menu Functions Utilities Help
-----
EDIT ---- IBMUSER.IMPLACTV.ACSLIB(STORGRP) - 01.00 ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** TOP OF DATA *****
PROC STORGRP

/*****/
/*          C H A N G E   H I S T O R Y          */
/*          =====                          */
/*          DATE    RESP  DESCRIPTION OF CHANGE          */
/*          -----  ---  -----                          */
/*          91/03/24  EG   Initial routine created.          */
/*          PURPOSE: This routine is never run for the minimal */
/*                   SMS configuration. It only exists to satisfy the */
/*                   requirement for storage group ACS routine */
/*                   for every valid SMS configuration. A storage */
/*                   group containing no real DASD volumes is assigned, */
/*                   NOVOLS. */
/*          INPUT:   The following ACS variables are referenced: NONE */
/*          OUTPUT:  The NOVOLS storage group is assigned. */

```

```

Menu Functions Utilities Help
-----
EDIT ---- IBMUSER.IMPLACTV.ACSLIB(STORGRP) - 01.00 ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
/*          RETURN    Zero is the only return code. Allocations are not */
/*          CODES:    failed in this routine. */
/*****/
/* Start of FILTLIST Statements */
/*****/
/* End of FILTLIST Statements */
/*****/
/* Start of Mainline SELECT */
/*****/
SET &STORGRP = 'NOVOLS'          /* Assign to storage group */
                                /* with no 'real' volumes */
/*****/
/* End of Mainline SELECT */
/*****/
END                               /* End of storage group proc*/
***** BOTTOM OF DATA *****

```

Figure 53. Sample Storage Group ACS Routine for the Minimal Configuration

9. Enter the END command to save the routine and return to the Edit-Entry panel (see Figure 51 on page 99). From this panel, enter the END command again to return to the ACS Application Selection panel (see Figure 50 on page 99).

**Translating the ACS Routines**

The following translation process checks the routines for syntax errors and converts the code into an ACS object. If the code translates without any syntax errors, then the ACS object is stored in the SCDS.

1. Select option 2 (Translate) from the ACS Application Selection panel (see Figure 50 on page 99), and press Enter to display the Translate ACS Routines panel shown in Figure 54 on page 102.



```

Panel Utilities Help
-----
                                TRANSLATE ACS ROUTINES
Command ==>

To Perform ACS Translation, Specify:

SCDS Name . . . . . 'YOUR.OWN.SCDS'
                                (1 to 44 Character Data Set Name)

ACS Source Data Set . . 'IBMUSER.IMPLACT.ACSLIB'
                                (1 to 44 Character Data Set Name)

ACS Source Member . . . 'STORCLAS'(1 to 8 characters)

Listing Data Set . . . 'STORCLAS.ERROR.LISTING'
                                (1 to 44 Character Data Set Name)

Use ENTER to Perform ACS Translation;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 54. Translating an ACS Routine

2. Enter the appropriate values for the storage class ACS routine. Press Enter to perform the translation. If errors are found, the listing contents are displayed. If the translation is successful, the message ACS OBJECT SAVED is displayed.
3. Enter the END command to return to the Translate ACS Routines panel. Next, translate the storage group ACS routine.

## Validating the SCDS

When you validate your SCDS, you verify that all classes and groups assigned by your ACS routines are defined in the SCDS. To validate the SCDS:

1. Enter an 8 (Control Data Set) on the ISMF Primary Option Menu to display the CDS Application Selection panel, shown in Figure 55.

```

Panel Utilities Help
-----
                                CDS APPLICATION SELECTION
Command ==>

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

Select one of the following Options:
1. Display      - Display the Base Configuration
2. Define      - Define the Base Configuration
3. Alter       - Alter the Base Configuration
4. Validate    - Validate the SCDS
5. Activate    - Activate the CDS
6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
7. Cache Update - Define/Alter/Delete CF Cache Sets
8. Lock Display - Display CF Lock Structure Names for all CF Lock Sets
9. Lock Update  - Define/Alter/Delete CF Lock Sets
If CACHE Display is chosen, Enter CF Cache Set Name . . *
If LOCK Display is chosen, Enter CF Lock Set Name . . . *
                                (1 to 8 character CF cache set name or * for all)

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

```

Figure 55. Specifying the SCDS Base Configuration

- In the CDS Name field, type in the name of the SCDS that is to contain the base configuration. The CDS name must be the same name that you previously used for the SCDS. In this example, the CDS name is YOUR.OWN.SCDS.  
Enter a 4 (Validate) to view the Validate ACS Routines or Entire SCDS panel, shown in Figure 56.

```

Panel Utilities Help
-----
                VALIDATE ACS ROUTINES OR ENTIRE SCDS
Command ==>

To Perform Validation, Specify:

SCDS Name . . . . . 'YOUR.OWN.SCDS'
                    (1 to 44 Character Data Set Name)

ACS Routine Type . . * (DC=Data Class, MC=Management Class, SC=Storage
                    Class, SG=Storage Group, *=Entire SCDS)

Listing Data Set . . VALIDATE.ERROR.LISTING
                    (1 to 44 Character Data Set Name)

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

```

Figure 56. Validating the SCDS

- In the SCDS Name field, specify the name of your SCDS. Enter an asterisk in the ACS Routine Type field to validate the entire SCDS.  
You can save a listing of the validation results by specifying a sequential data set name or partitioned data set member in the Listing Data Set field. If you leave this field blank, no listing is generated, so you do not see possible errors.  
You receive warning messages from the VALIDATE command if there are classes defined in your SCDS that are not assigned in your ACS routines.  
If you have storage groups defined in your SCDS that are not assigned in your ACS routines, you receive messages from VALIDATE and your SCDS is marked invalid.

---

## Defining the Storage Management Subsystem

In preparation for starting SMS, update the IEFSSNxx, IEASYSyy, and IGDSMSxx members of PARMLIB to define SMS to z/OS.

### Defining How to Activate SMS

You can activate SMS only after you define it to z/OS as a valid subsystem.

To define SMS to z/OS, you must place a record for SMS in an IEFSSNxx member. IEFSSNxx defines how z/OS activates the SMS address space. You can code an IEFSSNxx member with keyword or positional parameters, but not both.

**Recommendation:** Use keyword parameters.

Use the following keyword syntax to define SMS in IEFSSNxx:

```
SUBSYS SUBNAME(SMS)
[INITRN(IGDSSIIN) [INITPARM('ID=yy,PROMPT=NO')]
                    YES]]
                    DISPLAY)]]
```

If you choose to use positional instead of keyword parameters, use the following positional format to define SMS in IEFSSNxx:

```
SMS[, [IGDSSIIN] [, 'ID=yy[, PROMPT=NO']]
                    YES]]
                    DISPLAY]]
```

Where:

**ID=yy**

yy is the two-character suffix for the SMS initialization member, IGDSMSxx.

**PROMPT=DISPLAY**

This option displays the contents of the IGDSMSxx member, but you cannot change the contents.

During initial testing, you probably want to be able to start SMS manually. Omit IGDSSIIN in the SMS record to do this. Once you are comfortable with SMS operation, add IGDSSIIN to cause SMS to start automatically during IPL.

**Recommendation:** Place the SMS record before the JES2 record in IEFSSNxx to start SMS before starting the JES2 subsystem.

**Related Reading:** For complete descriptions of these parameters, see *z/OS DFSMSdfp Storage Administration*.

## Specifying Members to be Used During IPL

You must also update the IEASYSyy member of PARMLIB to identify the IGDSMSxx member used during SMS initialization. Add the line SMS=xx to this PARMLIB member.

If you are defining additional subsystems, include their records in members following the xx member in the SSN= parameter, in addition to SMS=yy, as follows:

```
SSN=(xx,01,...)   where xx is the suffix of the IEFSSNxx member
                  containing the SMS record
SMS=yy           where yy is the same value as on ID=yy in the
                  IEFSSNxx member
```

## Defining the ACDS and COMMDS

IGDSMSxx contains the information that is used to initialize the SMS address space and identify the COMMDS. The data sets that you specify for the ACDS and COMMDS pair must be the same for every system that shares DASD in your SMS configuration.

IGDSMSxx also sets the synchronization time interval between systems. This interval represents how many seconds SMS lets elapse before it checks the COMMDS for status from other systems.

If you plan to use the RACF default data class, storage class, and management class for the data set owner, you must specify ACSDEFAULTS=YES. The following command shows a basic format for the SMS record you submit:

```
SMS ACDS(YOUR.OWN.ACDS)
    COMMDS(YOUR.OWN.COMMDS)
```

**Related Reading:** For descriptions of the IGDSMSxx parameters, see *z/OS MVS Initialization and Tuning Reference* or *z/OS DFSMSdfp Storage Administration*.

## Activating a New SMS Configuration

You can manually activate a new SMS configuration in two ways. SMS must be active before you use one of these methods. To start SMS, IPL the system with SMS defined as a valid subsystem, and start SMS automatically at IPL. Or, IPL the system with SMS defined as a valid subsystem, and start SMS later using the SET SMS=yy operator command.

**Related Reading:** For more information about activating a new SMS configuration, see *z/OS DFSMSdfp Storage Administration*.

### Activating an SMS Configuration from ISMF

Select the ACTIVATE option, or enter the ACTIVATE command on the command line of the ISMF CDS Application Selection panel (see Figure 55 on page 102). The information from the SCDS is copied into the ACDS. The SCDS itself is never considered active. Attempting to activate an ACDS that is not valid results in an error message.

### Activating an SMS Configuration from the Operator Console

From the operator console, enter the following command:

```
SETSMS {ACDS(YOUR.OWN.ACDS)} {SCDS(YOUR.OWN.SCDS)}
```

YOUR.OWN.ACDS specifies a data set that has been defined as an ACDS. To activate the configuration, information is brought into the SMS address space from the ACDS. To update the current ACDS with the contents of an SCDS, specify the name of the SCDS only. Otherwise, if you want to both specify a new ACDS and update it with the contents of an SCDS, enter the SETSMS command with both the ACDS and SCDS specified.

**Tip:** The ACTIVATE command, run from the ISMF CDS application, is equivalent to the SETSMS operator command with the SCDS keyword specified. If you use RACF, you can enable storage administrators to activate SMS configurations from ISMF by defining the facility, STGADMIN.IGD.ACTIVATE.CONFIGURATION, and issuing permit commands for each storage administrator.

Display configuration status information by entering ACTIVE in the CDS NAME field on the CDS Application Selection panel (see Figure 55 on page 102).

z/OS operator commands complement your ability to monitor and control SMS operation. You can use the DISPLAY operator command to show information about the active configuration. The following sample command displays the status of the storage group PRIME80 and all the volumes defined in the storage group:

```
DISPLAY SMS,STORGRP(PRIME80),LISTVOL
```

Figure 57 on page 106 shows output from this command.

```

00- 20.47.08          D SMS,STORGRP(PRIME80),LISTVOL
    20.47.08          IGD002I 20:47:08 DISPLAY SMS 825      C
    STORGRP TYPE          SYSTEM= 1
    PRIME80 POOL          +
    VOLUME UNIT SYSTEM= 1          STORGRP NAME
    PRM801          +          PRIME80
    PRM802          +          PRIME80
    PRM803          +          PRIME80
***** LEGEND *****
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
SYSTEM 1 = SYSTEM1
IEE612I CN=01          DEVNUM=040 SYS=SYSTEM1

```

Figure 57. Storage Group and Volume Status for PRIME80

## Activating SMS

To activate SMS, perform the following steps:

1. Define SMS to the operating system.
 

Update the PARMLIB members on all the systems where SMS is intended to run. If you are sharing DASD between systems, you only need to activate the SMS configuration on one system; the COMMDS activates the other systems. If you want to prevent initial activation on other systems, the ACDS and COMMDS should reside on non-shared volumes.
2. Initialize the SMS address space.

**Recommendation:** To simplify recovery, leave empty one of the eight system or system group name slots for 8-name mode, and one of the 32 system or system group name (or a combination or both) slots for 32-name mode. If a problem occurs (such as a hardware failure), you might need to specify a system that was in the configuration as part of a system group. If there is an empty name slot, you can add the system name without disrupting the other systems in the SMS complex.

These are the steps for recovery:

1. Use the ISMF CDS Application (option 3) to add the system name to the SMS configuration.
2. Activate the configuration, either from ISMF or from the operator console.
3. Use the VARY SMS operator command to update the configuration.
4. When the problem is corrected, remove the system name from the configuration.
5. Reactivate the configuration.

## Controlling SMS Processing with MVS Operator Commands

The DFSMS environment provides a set of MVS commands the operator can use to control SMS processing. The VARY, DISPLAY, DEVSERV, and SET commands are MVS operator commands that support SMS operation.

## SETSMS

This changes SMS options from the operator console. You can use this command to activate a new configuration from an SCDS. SETSMS supports SMS and is modeled after the SETSMF command, which controls SMF processing. The MVS operator must use SETSMS to recover from ACDS and COMMDS failures. You can also use the SETSMS SAVESCDS command to save the active ACDS as an SCDS, if the current SCDS and its backup are lost.

**Related Reading:** For an explanation of how to recover from ACDS and COMMDS failures, see *z/OS DFSMSdfp Storage Administration*.

## SET SMS

- Changes options set from PARMLIB for SMS
- Restarts SMS if it has terminated
- Updates SMS configuration
- Starts SMS, if it has not already been started and is defined as a valid z/OS subsystem

## VARY SMS

Changes storage group, volume, library, or drive status. Use this command to:

- Limit new allocations to a volume or storage group
- Enable a newly-installed volume for allocations

## DISPLAY SMS

This shows volumes, storage groups, libraries, drives, SMS configuration information, SMS trace parameters, SMS operational options, OAM information, OSMC information, and cache information. Use this command to:

- Confirm that the system-managed volume status is correct
- Confirm that SMS starts with the proper parameters

## DEVSERV

This displays information for a device. Use it to display the status of extended functions in operation for a given volume that is attached to a cache-capable 3990 storage control. Figure 58 shows the output from a typical DEVSERV request.

```
DEVSERV P,1000,4
IEE459I 10.45.49 DEVSERV PATHS 780
UNIT DTYPE M CNT VOLSER  CHPID=PATH STATUS
      TC DFW PIN DC-STATE CCA DDC ALT CU-TYPE
1000,3380K,0,000,D65DM1,2E=+ 2F=+
      YY YY N SIMPLEX C0 00 3990-3
1001,3380K,0,000,D65DM2,2E=+ 2F=+
      YY NY N SIMPLEX C1 01 3990-3
1002,3380K,0,000,D65DM3,2E=+ 2F=+
      NY YY N SIMPLEX C2 02 3990-3
1003,3380K,0,000,D65DM4,2E=+ 2F=+
      YY NY N SIMPLEX C3 03 3990-3
***** SYMBOL DEFINITIONS *****
0 = ONLINE + = PATH AVAILABLE
```

Figure 58. Output from the DEVSERV command

**Related Reading:** For more information about operator commands, see *z/OS MVS System Commands*.

---

## Enforcing Standards

You can use data class ACS routine facilities to automate or simplify storage allocation standards if you:

- Use manual techniques to enforce standards
- Plan to enforce standards before implementing DFSMS
- Use DFSMSdfp or MVS installation exits to enforce storage allocation standards

The data class ACS routine provides an automatic method for enforcing standards, because it is called for system-managed and non-system-managed data set allocations. Standards are enforced automatically at allocation time, rather than through manual techniques after allocation.

Enforcing standards optimizes data processing resources, improves service to users, and positions you for implementing system-managed storage. You can fail requests or issue warning messages to users who do not conform to standards. Consider enforcing the following standards in your DFSMS environment:

- Prevent extended retention or expiration periods.
- Prevent specific volume allocations, unless authorized.  
For example, you can control allocations to spare, system, database, or other volumes.
- Require valid naming conventions for permanent data sets.

Appendix C, “Installation and User Exits,” on page 259, describes the installation exits available in the DFSMS environment. Use the information to evaluate if your installation exit usage continues to apply to system-managed data sets.

---

## Chapter 5. Managing Temporary Data

**Recommendation:** After you activate SMS, select temporary data sets for migration to system-managed storage. Managing temporary data sets during this phase of the implementation enables you to gain experience with DFSMS functions. Temporary data sets are easy to identify and recreate, and do not require availability management. The following are benefits of system-managed temporary data sets:

- Both temporary VSAM and non-VSAM data sets are supported.
- Temporary data sets are automatically deleted by DFSMSHsm, thus supporting efficient space management.

Temporary data sets left by the system because of abnormal end-of-job are deleted by DFSMSHsm during space management.

- Temporary data sets can be allocated on VIO or system-managed storage volumes.

Using system-managed temporary data eliminates the need for dedicated public or temporary volumes, and frees volumes for permanent data sets.

- Temporary data sets can be directed to VIO based on data set size and usage, improving performance by directing I/O to your processor's central or expanded storage.

VIO data sets are backed by auxiliary storage as paging data sets.

- Temporary data set access to VIO can be restricted, based on the RACF variables, &USER and &GROUP.

RACF-related ACS READ/WRITE variables, &USER and &GROUP, can be interrogated in your ACS routine to limit access to VIO.

Temporary data sets are created and deleted within the same job, job step, or terminal session. No entries are made in the basic catalog structure (BCS) for these data sets, but system-managed VSAM data sets do have VVDS entries. Both VSAM and non-VSAM data sets have VTOC entries. The data set name for temporary data is either omitted or is a single qualifier with && or && at the beginning. When the DSNNAME is omitted, the system generates a name that begins with SYS and includes the Julian date and time.

When you code request temporary data set allocation, the ACS read-only variable data set type, &DSTYPE, is set to TEMP. The storage class ACS routine determines whether to allocate these data sets to VIO or to volumes in a pool storage group category depending on the data set usage and size. During automatic space management, DFSMSHsm automatically deletes system-managed temporary data sets that remain on a volume after an abnormal end of job or system failure. Figure 59 on page 110 shows how DFSMSHsm allocates and manages temporary data sets.



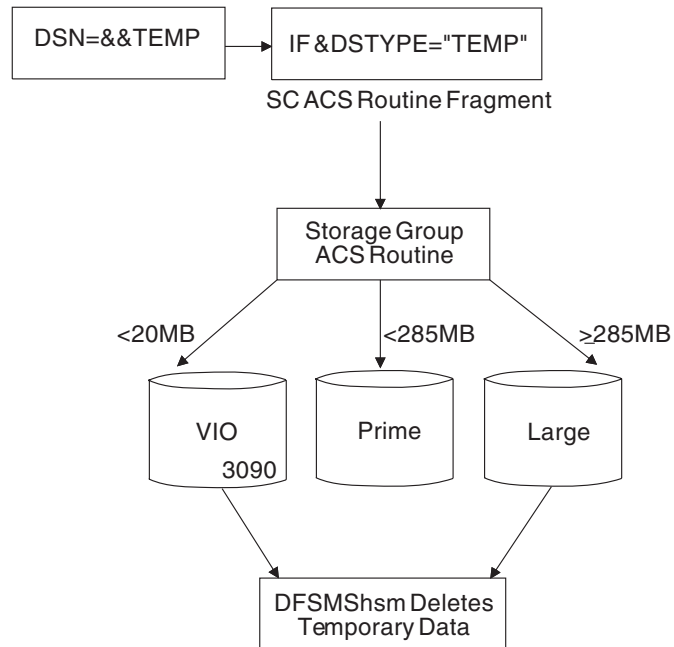


Figure 59. System-Managed Temporary Data

The following major tasks are required for system-managed temporary data sets:

- Review the planning considerations
- Define SMS storage classes and groups
- Create ACS routines
- Test the ACS routines
- Initialize DASD volumes for LARGE<sub>xx</sub> and PRIME<sub>xx</sub> storage groups
- Reactivate the configuration

---

## Planning to Manage Temporary Data

Properly managing temporary data involves determining which data sets can be directed to VIO or to the primary or large storage groups. This helps improve performance and reduce I/O controller and device workload. VIO simulates I/O to the VIO UNIT in processor and auxiliary storage that you specify in the VIO storage group.

Review the following advice on VIO usage:

- Do not direct DFSORT temporary SORTWK<sub>nn</sub> data sets to VIO.  
SORTIN and SORTOUT data sets might be good candidates, especially during the offshift processing periods. However, we do not recommend allocating DFSORT temporary work data sets as VIO data sets. Instead, consider using the Hipersorting function of DFSORT. Hipersorting improves DFSORT elapsed time and reduces execute channel programs (EXCPs) by using Hiperspace™ in place of some or all of the SORTWK<sub>nn</sub> space. DFSORT queries the system about paging activity and, based on the information returned, allocates Hiperspace as long as it does not adversely affect the performance of other programs using expanded storage. For more information about Hipersorting, refer to *z/OS DFSORT Application Programming Guide* or to *z/OS DFSORT Tuning Guide*.
- Do not direct OS COBOL II Interactive Debug for SYSUT5 to system-managed storage. An unmovable data set is dynamically allocated. This data set cannot be system-managed.

- Assess your systems' paging activity and the size of central and expanded storage to determine the maximum data set size to be directed to VIO:
  - Temporary data sets between 2 MB and 6 MB are potential candidates for VIO. IBM recommends 2 MB as an initial threshold.
  - Evaluate the amount of expanded and central storage on all CPUs if you have multiple processors.
  - Create a specific VIO storage group for each CPU based on its storage size. You can tailor the VIO unit and maximum size in each storage group definition and set the status to DISALL for the other systems in the multi-processor complex.

---

## Defining SMS Classes and Groups

This section describes how to define SMS storage classes and groups for system-managed temporary data. To allocate temporary data sets on system-managed volumes, you must define one or more storage classes and groups. Temporary data does not need a management class because it does not require backup and migration. Allocation bypasses the management class ACS routine.

### Defining the Storage Classes

Define two additional storage classes, NONVIO and STANDARD, to support system-managed temporary data set processing, using the ISMF Storage Class Application. The NONVIO storage class is an artificial class that is set in the storage class ACS routine, and is tested in the storage group ACS routine to inhibit DFSORT temporary SORTWKnn data sets from being directed to VIO. Assign all temporary data sets other than DFSORT work data sets to the STANDARD storage class. The NONVIO and STANDARD storage classes are defined in the starter set's SCDS. Copy these storage classes into your SCDS and tailor the STANDARD storage class:

1. Enter 5 (Storage Class) on the ISMF Primary Option Menu to view the Storage Class Application Selection panel.
2. Type your source control data set name in the CDS Name field and STANDARD in the Storage Class Name field.  
Enter 4 (Alter) to view the Storage Class Alter panel.
3. Tailor the Performance Objectives fields to meet your performance requirements for data sets. Use this storage class as the default for system-managed data sets.
4. Tailor the Description field.
5. Press Enter to verify the information entered. Then press PF3 to save your changes.
6. Press PF3 to return to the ISMF Primary Option Menu.

### Defining the Storage Groups

Add the definitions for the VIO, PRIMExx and LARGExx storage groups to the minimal configuration. In the starter set, xx is used to indicate the type of device in the storage group. A value of 80 indicates that the storage group contains 3380s, and 90 indicates that 3390 devices comprise the storage group. Modify these suffixes to match your DASD configuration. You can copy the storage groups from the starter set's SCDS to create your storage groups. In the process, you can change the storage group names. These storage groups are used for temporary data sets, as follows:

- VIO contains data sets less than 20 MB.

- PRIMExx contains data sets greater than, or equal to, 20 MB and less than 285 MB.
- LARGExx contains data sets greater than, or equal to, 285 MB.

The PRIMExx and LARGExx storage groups support temporary data sets that are too large for VIO support. To tailor the storage groups:

1. Enter 6 (Storage Group) on the ISMF Primary Option Menu to view the Storage Group Application Selection panel.
2. Type in values for the CDS Name, Storage Group Name and Storage Group Type fields. In this example, the CDS name is YOUR.OWN.SCDS, the storage group name is VIO, and the storage group type is VIO.

Enter 3 (Alter) to view the VIO Storage Group Alter panel. The VIO Maxsize attribute in the storage group determines the largest temporary data set that can be written to VIO. You determine the size by examining your use of central and expanded storage and your paging activity.

Type in your threshold as the VIO Maxsize attribute if you need to change the 20 MB value. This is the primary space size plus 15 times the secondary space size.

Reasonable limits for VIO depend far more on the sizes of paging data sets than they do on the amount of central storage.

Type in a device type as the VIO UNIT attribute. The VIO device type is virtual and is unrelated to actual devices on your system. Update the Description field to reflect your changes.

Press Enter to verify your changes. Then press PF3 to save the updated storage group. Press PF3 again to return to the Storage Group Application Selection panel.

3. Now tailor the PRIMExx storage group.
4. Enter a 4 (Volume) on the Storage Group Application Selection panel. Then, specify a volume or range of volumes and enter a 2 (Define) on the Storage Group Volume Selection panel. Define the relationship between the volume and each of your systems or system groups by typing in ENABLE in the SMS VOL STATUS column next to the appropriate system or system group names in the System/Sys Group Name column.
5. Optionally, define the LARGExx storage group in the same way as you did for the PRIMExx storage group.

**Restriction:** If you use VOL=REF processing to refer to a temporary data set, you might get different results in storage group assignments than expected. This is because temporary data sets are assigned a storage group by the system, based on a list of eligible storage groups, such as: VIO, PRIME, STANDARD, etc. Data sets that use VOL=REF are assigned a storage group based on this list of eligible storage groups, not on the name of the storage group used to successfully allocate the first data set being referenced. This might result in the data sets being allocated in different storage groups.

---

## Creating ACS Routines

After you define the SMS storage classes and groups, design the ACS routines. Use ACS routines to automatically assign the storage classes and groups to all new temporary data set allocations. To manage temporary data sets in ACS routine logic, direct small, temporary data sets to VIO, medium data sets to the PRIMExx storage group, and large data sets to the LARGExx storage group. Assign all DFSORT temporary work data sets to a non-VIO storage class.

## Writing the ACS Routines

Use the ISPF/PDF editor, called from the ISMF Automatic Class Selection Application, to develop your ACS routines based on the starter set or the examples for the Temporary milestone in Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243. Perform the following steps:

- Tailor the storage class ACS routine, based on any changes in storage class names, valid device names, or DDNAMEs.
- Tailor the storage group ACS routine, based on any changes in storage group names or large data set size threshold.

### Tailoring the Storage Class ACS Routine

The data class and storage class ACS routines process all new allocations, not just ones for direct access storage. You must ensure that non-DASD requests for allocations are not assigned a non-null storage class. Define your esoteric names for DASD, such as SYSDA, in a FILTLIST that includes all valid unit types. Check the UNIT against this FILTLIST. If the UNIT is not valid, assign a null storage class.

### Tailoring the Storage Group ACS Routine

Assign data sets with a storage class name of NONVIO to a PRIMExx storage group if they are smaller than 285 MB. Otherwise, assign them to a LARGExx storage group.

## Translating the ACS Routines

After writing the ACS routines, use ISMF to translate them into executable form. The translation process checks the routines for syntax errors and converts the code into valid ACS routines. If the code translates without any syntax errors, then the SCDS is updated with the new ACS routine. To translate the ACS routines:

1. Select option 2 (Translate) from the ACS Application Selection panel, and press Enter to view the Translate ACS Routine panel.
2. Enter the appropriate data set name for the storage class ACS routine. Press Enter to perform the translation.
3. When the translation is complete, enter the END command to return to the Translate ACS Routine panel, and complete the fields for the storage group ACS routine.

---

## Testing the ACS Routines

After completing your ACS routines, you can use ISMF to write and run test cases to verify that the routines are properly assigning the SMS storage classes and groups. Testing the individual routines of a new or modified configuration lets you activate SMS with greater confidence.

**Restriction:** ACS installation exits are not called during ACS routine testing.

## Creating ACS Test Cases

**Recommendation:** Allocate a PDS for the ACS test cases.

1. Enter a 7 (Automatic Class Selection) on the ISMF Primary Option Menu to view the ACS Application Selection panel, shown in Figure 60 on page 114.

```

Panel  Utilities  Help
-----
                                ACS APPLICATION SELECTION
Command ==>>

Select one of the following options:
4 1. Edit           - Edit ACS Routine source code
  2. Translate      - Translate ACS Routines to ACS Object Form
  3. Validate       - Validate ACS Routines Against Storage Constructs
  4. Test           - Define/Alter Test Cases and Test ACS Routines
  5. Display        - Display ACS Object Information
  6. Delete         - Delete an ACS Object from a Source Control Data Set

If Display Option is Chosen, Specify:

CDS Name  . . 'YOUR.OWN.SCDS'
                                (1 to 44 Character Data Set Name or 'Active')

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

```

Figure 60. Creating ACS Test Cases

2. Enter a 4 (Test) to view the ACS Test Selection panel, shown in Figure 61.

```

Panel  Utilities  Help
-----
                                ACS TEST SELECTION
Command ==>>

Select one of the following Options:

1 1. DEFINE        - Define an ACS Test Case
  2. ALTER         - Alter an ACS Test Case
  3. TEST          - Test ACS Routines

If DEFINE or ALTER Option is Chosen, Specify:

ACS Test Library . . USER6.TEST.DATA
ACS Test Member  . . ONCE1

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

```

Figure 61. Defining ACS Test Cases

3. Type in the name of the PDS containing the ACS test case data in the ACS Test Library field. In this example, the data set name is USER6.TEST.DATA. Type in the name of the particular library member containing the test case in the ACS Test Member field. You can type in one test case per member.  
Enter a 1 to view the first page of the ACS Test Case Define panel, as shown in Figure 62 on page 115.

```

Panel Utilities Scroll Help
-----
                        ACS TEST CASE DEFINE                        Page 1 of 4
Command ==>>

ACS Test Library : USER6.TEST.DATA
ACS Test Member  : ONCE1

To DEFINE ACS Test Case, Specify:
Description ==>
Expected Result
DSN (DSN/Collection Name) . .
MEMN (Object Name) . . . . .
Sysname . . . . . Xmode . . . . . Def_dataclas . .
Sysplex . . . . . ACSenvir . . . . . Def_mgmtclas . .
DD . . . . . Dataclas . . . . . Def_storclas . .
Dsorg . . . . . Mgmtclas . . . . . Dsntype . . . . .
Recorg . . . . . Storclas . . . . . If Ext . . . . .
Dstype . . . . . Storgrp . . . . . Seclabel . . . . .
Downer . . . . . Size . . . . . Space_Type . . . . .
Expdt . . . . . Maxsize . . . . . Second_Qty . . . . .
Retpd . . . . . Blksize . . . . .

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 62. ACS Test Case Define Panel, Page 1 of 4

ACS Test Library and ACS Test Member are output fields containing the values that you specified on the ACS Test Selection panel (see Figure 61 on page 114). Description is an optional field of 120 characters that you can use to describe the test case.

4. Specify the appropriate values. The following are sample values for your use:
  - DSN: STGADMIN.TEST.TEMPDATA
  - DD: SORTWK1
  - Dsorg: PS
  - Dstype: TEMP
  - Xmode: BATCH
  - ACSenvir: ALLOC
  - MAXSIZE: 400000

Leave the remaining fields blank and scroll down to view the second page of the ACS Test Case Define panel, shown in Figure 63 on page 116.

```

Panel Utilities Scroll Help
-----
                        ACS TEST CASE DEFINE                        Page 2 of 4
Command ==>

ACS Test Library : USER6.TEST.DATA
ACS Test Member . : ONCE1

To DEFINE ACS Test Case, Specify:
Job . . . . . STGADM01 Pgm . . . . . Vol . . . . .
Group . . . . . SYSADMIN  Applic . . . . . Unit . . . . . 3390
User . . . . . Nvol . . . . .
Acct_job ==>
          ==>
          ==>
Acct_step ==>
          ==>
          ==>
Label . . . . .
Filenum . . . . .
Libname . . . . .

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 63. ACS Test Case Panel, Page 2 of 4

ACS Test Library and ACS Test Member are output fields containing the values that you specified on the ACS Test Selection panel (see Figure 61 on page 114).

5. Specify STGADM01 in the JOB field, SYSADMIN in the GROUP field, and 3390 in the UNIT field. Leave the remaining fields blank.

Press Enter to perform the verification. Enter the END command to return to the ACS Test Selection panel (see Figure 61 on page 114).

## Running ACS Test Cases

To run the test cases developed in “Creating ACS Test Cases” on page 113:

1. Enter a 3 (Test) on the ACS Test Selection panel (see Figure 61 on page 114) to view the Test ACS Routines panel, shown in Figure 64.

```

Panel Utilities Help
-----
                        TEST ACS ROUTINES
Command ==>

To Perform ACS Testing, Specify:

CDS Name . . . . . 'YOUR.OWN.SCDS'
                                     (1 to 44 Character Data Set Name or 'Active')
ACS Test Library . . 'USER6.TEST.DATA'
ACS Test Member . . ONCE1      (fully or partially specified or * for all
                               members)
Listing Data Set . . 'TEMP.TEST01.LISTING'
                                     (1 to 44 Character Data Set Name or Blank)

Select which ACS Routines to Test:

DC ==> N (Y/N) SC ==> Y (Y/N) MC ==> N (Y/N) SG ==> Y (Y/N)

Use ENTER to Perform Verification and Testing;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 64. Test ACS Routines Panel

2. Type in the CDS name. The CDS name must be the same as the one you used for the SCDS on the CDS Application Selection panel. The ACS test library must be the same as the one you used on page 1 of the ACS Test Case Define panel (see Figure 63 on page 116). In this example, the CDS name is YOUR.OWN.SCDS and the ACS test library name is STGADMIN.ACSTEST.LIB. Type in an asterisk in the ACS Test Member field to run all the test cases in the PDS library. To create a list of the test results, type in a data set name in the Listing Data Set field.  
Press Enter to test the ACS routines. If you entered a name in the Listing Data Set field, the results are displayed on the PDF Browse panel, shown in Figure 65.

```

Menu Utilities Compilers Help
-----
BROWSE   TEST01.LISTING                               Line 00000000 Col 001 080
Command ==>                                         Scroll ==> DATA
***** Top of Data *****
                        ACS TESTING RESULTS

CDS NAME       : YOUR.OWN.SCDS
ACS ROUTINE TYPES: SC SG
ACS TEST LIBRARY : USER6.TEST.DATA

ACS TEST
MEMBER        EXIT CODE  RESULTS
-----
DESCRIPTION: TEST CASE CREATED 99/11/07 AT 10:24
TEMPT01      0 SC = NORMAL
              0 SG = NORMAL

ACS TESTING RC: 00
***** Bottom of Data *****

```

Figure 65. Creating an ACS Output Listing

3. After examining the results, enter the END command to view the ACS Output Listing Disposition panel, on which you can specify whether to keep the output listing.

---

## Initializing DASD Volumes for Large and Prime Storage Groups

If you have dedicated temporary volumes, you can reinitialize them as system-managed, and add them to one of the two storage groups after deallocating them. Volumes that are allocated cannot be reinitialized until all active tasks using data sets on the volume terminate. An IPL might be necessary to free volumes allocated by the system.

---

## Reactivating the Configuration

After you make changes to the ACS routines, validate and reactivate the SMS configuration.

### Validating the Configuration

Validation checks that you only assign classes and groups that are defined in your SCDS. Validation also ensures that all storage groups defined in your SCDS are assigned in your storage group ACS routine. To validate your SMS configuration:

1. On the ISMF Primary Option Menu, enter an 8 (Control Data Set) to view the CDS Application Selection panel.



2. The CDS Name field displays the name of the SCDS that has been previously defined.  
Enter a 4 (Validate) to view the Validate ACS Routines or Entire SCDS panel.
3. The CDS Name field displays the name of the SCDS in use. Enter an asterisk in the ACS Routine Type field to validate the entire SCDS. Type in a meaningful data set name in the Listing Data Set field.  
Press Enter to perform the validation. After it completes, a browse screen displays, showing whether the validation is successful.
4. From the browse screen, enter the END command to view the Output Listing Disposition panel.  
Fill in the fields as appropriate and press Enter to return to the Validate ACS Routines or Entire SCDS panel. Enter the END command to return to the CDS Application Selection panel.

## Activating the Configuration

The activate function lets you change the active SMS configuration. This function causes the MVS operator command, SETSMS, to be run using the name of the SCDS supplied in the CDS Name field. Ensure that you are authorized to use this command, otherwise the operator must decide whether to allow the configuration change. Follow these steps to activate your SMS configuration:

1. On the CDS Application Selection panel, enter a 5 (Activate) to view the Confirm Activate Request panel.
2. The CDS Name field displays the name of the previously defined SCDS. This name must be the same as that used during the configuration validation.  
Enter a Y in the Perform Activation field to activate the configuration.

---

## Chapter 6. Managing Permanent Data

This chapter describes the SMS configuration enhancements required to implement system management by data type. Special considerations for each data type are noted.

---

### Identifying Types of Permanent Data

The following are potential types of data in your environment:

- TSO
- Batch
- Tape mount management candidates
- Database data: CICS/VSAM, IMS, DB2
- Tape volumes
- Objects

TSO, batch, and database data are the usual candidates for migration during the Managing Permanent Data phase. Tape data sets and volumes are migrated in the Managing Tape Data milestone. You can migrate database data sets and tape volumes under SMS in any order.

Most data set types can benefit from SMS, with some types benefiting more than others.

Figure 66 on page 120 shows how application-owned volumes are migrated to system-managed storage groups using DFSMSdss. Sample ACS routines to migrate permanent data sets to the DFSMS environment are provided in Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243.

**Recommendation:** Implement these design and migration activities jointly with user group representatives to share experience and gain credibility. The order in which you migrate data types can vary, based on your objectives for the migration and the readiness of your users to participate in the effort.

**Related Reading:** For specific information about objects, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

---

### Planning the Migration of Permanent Data

**Recommendation:** Use the following steps to migrate permanent data:

1. Review the benefits for managing the data type.
2. Understand how this data is currently allocated and managed.
3. Develop a strategy to identify the data using data set name, RACF ownership, or job accounting information.
4. Determine performance service-level requirements and specify them in storage classes.
5. Determine space management and backup requirements and specify them in management classes.
6. Decide on a migration strategy, and simulate the migration using DFSMSdss facilities.

7. Define any required storage groups or add volumes to accommodate the new data type.
8. Develop and test ACS routines using ISMF facilities.
9. Supplement DFSMSHsm migration level 1, migration level 2, and backup volumes to support the data.
10. Activate the new configuration, and verify that new allocations are system-managed.
11. Migrate existing data to system-managed storage.
12. Optionally, evaluate the usefulness of SMS data classes for the data type and implement as appropriate.

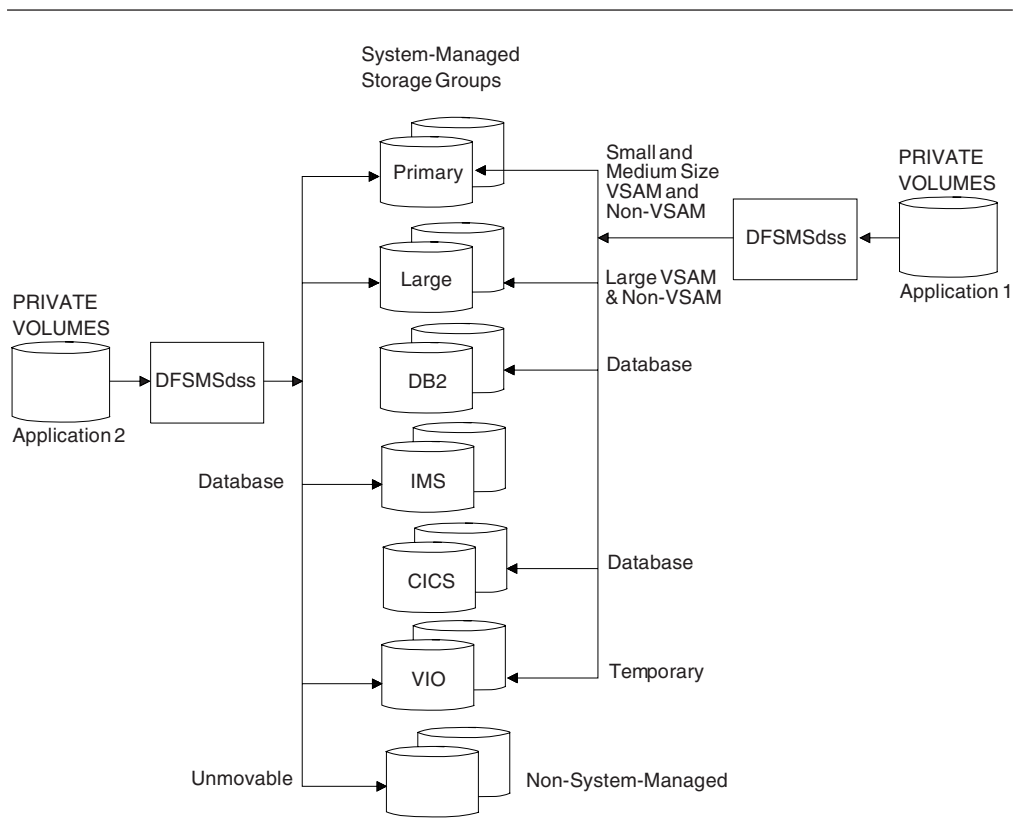


Figure 66. Migrating Permanent Data

---

## Chapter 7. Managing TSO and HFS Data

This chapter describes a process for placing your TSO data under system management. Data accessed using TSO can include database data if you have Query Management Facility™ (QMF\*) or an equivalent product installed. However, the TSO data that this chapter addresses is data owned by application development.

These data sets are relatively easy to place under system management because allocation requests for TSO data typically use esoteric unit names, and the data sets are already in storage pools. You can readily convert these pools to the primary storage groups, PRIME80 and PRIME90.

If you now use DFSMSHsm, you are probably already providing management services for TSO data. Implementing DFSMS-based space management and availability services requires translating the volume-oriented management parameters documented in your DFSMSHsm PARMLIB member to the data-set-oriented SMS management classes.

HFS data sets are similar to TSO data sets. HFS files, like PDSE members, cannot be individually managed. These files cannot be converted. They are created in HFS data sets as hierarchical files using z/OS UNIX System Services. HFS data sets must be system-managed.

Data set-level DFSMSdss and DFSMSHsm functions can be performed on HFS data sets. However, file level backups can only be performed using Tivoli Storage Manager clients.

---

### Managing TSO Data

The following are the main benefits of managing TSO data:

- TSO data sets benefit from DFSMSHsm's fully automatic availability and space management for these data sets.

These data sets are usually smaller and have lower I/O activity than production data sets, and typically do not have predictable access or update patterns. Automation of these management services relieves application developers from time-consuming storage management tasks.

- You can use DASD for TSO data sets more efficiently by specifying immediate release of unused space through management class.

Unused space is released at data set CLOSE time if you specify Partial Release=YES IMMED (or COND IMMED, to ensure that space is not released unless secondary space is specified) in the management class assigned to the TSO data set. This causes unused space to be released immediately if a secondary allocation is specified.

- You can simplify JCL development requirements for your application developers and improve space usage for data sets with SMS data classes.

Using the DATACLAS parameter to supply DD statement parameters for frequently allocated data types benefits both you and application developers.

Some benefits of data classes are:

- Standards enforcement: The JCL simplification is attractive to application developers, and provides an incentive to conform to data set naming standards to fully automate data class assignment.
- Optimal block size usage: When allocation is assisted by data class specification, the system determines the optimal block size for the data set based on the device selected by SMS.
- Using DFSMS's program management, you can store source and load libraries currently organized as PDSs in PDSE format.

The PDSE format is supported only by the program management binder and loader, not by the linkage editor. Unlike PDSs, PDSEs do not require periodic compression to consolidate fragmented space for reuse. You do not have to recreate PDSEs when the number of members expands beyond the PDS's available directory blocks. PDSEs are supported by many of the utilities that currently support PDSs. Members in PDSEs can be read and written concurrently by multiple systems.

SMS allocates TSO data in the PRIMExx and LARGExx storage groups, based on data set size. Data sets larger than 285 MB are directed to the LARGExx storage group. Listing data sets, SYSPRINT from compilers and linkage editors, are automatically deleted by DFSMSHsm after a short life on primary storage. Active source, object, and load libraries exist on primary storage indefinitely. If these data sets are not used, they are migrated by DFSMSHsm to migration level 1, but are recalled automatically when accessed by a user.

Multiple versions of backups of TSO data sets are maintained to minimize the effect of accidental deletion of application programmer data, such as source or JCL libraries. These data sets receive better-than-average availability service.

TSO data sets do not have high performance requirements in comparison to other data categories, and are assigned standard performance services.

The major tasks for managing TSO data include:

- Review the planning considerations for data migration to system management, including fallback contingencies.
- Design and test performance and availability services for TSO data.
- Design and test backup and space management services for TSO data.
- Determine your physical space requirements for TSO data and add volumes to the PRIMExx and LARGExx storage groups.
- Determine any additional resources required for DFSMSHsm space and availability management.
- Activate new configuration.
- Migrate TSO data.
- Design and test automated data allocation, using data classes.

---

## Choosing a Strategy for Placing Your Volumes under System Management

You can either convert in-place or place data under system management by moving the data to system-managed volumes. The technique you use depends on the way your TSO data sets are grouped today, and on your objectives for managing TSO data in the DFSMS environment:

- Do your TSO volumes have indexed VTOCs, adequately-sized VVDS?

System-managed volumes must have indexed VTOCs. Also, system-managed non-VSAM data sets have a Non-VSAM Volume Record (NVR) containing SMS class names for the data set in the VVDS.

**Related Reading:** For information on estimating the VVDS size, see *z/OS DFSMSdss Storage Administration*.

- Are you satisfied with your TSO data set performance?

If you are not, consider converting the data sets *with movement* to achieve performance based on the performance objectives that you set in the storage classes for TSO data sets.

- Are you satisfied with the efficiency of TSO data set space usage?

You can evaluate space usage on the volume or data set level. You might have some volumes that are used much more than others. If you place the TSO data under system management by moving it to other volumes, you can specify target thresholds in the SMS storage group for TSO data sets to better balance the volume use. At the data set level, your application programmers might be over-allocating space for data sets or using inefficient block sizes for their data sets. Converting with movement improves the efficiency of space use at the data set level, too.

- Is your TSO data currently mixed with non-TSO data in a storage pool?

If your TSO data is not isolated, you must move the TSO data sets, move the non-TSO data sets, or use an approach that combines both to form a group of data sets that can be placed under system management.

- Are you managing your TSO data with DFSMSHsm today?

If so, you can convert in-place or use DFSMSHsm to migrate the data to migration level 1 and have the TSO data converted on demand when a user accesses the data. The DFSMSHsm recall runs your storage class ACS routine, and if your routine determines that a data set should be system-managed, the data set is allocated on a system-managed volume. The management class ACS routine is also run to assign a management class for the data set.

**Recommendation:** If many of your TSO data sets are inactive and you are implementing a more aggressive policy to keep only active data sets on primary, use the on-demand technique.

---

## Determining a Fallback Strategy

Plan how you can remove selected data sets from system management, in case of system or application problems.

1. Determine the DFSMSdss filters for the data sets to be selected for migration to non-system-managed volumes.
2. Select a spare volume to temporarily contain these data sets.
3. Update your storage class ACS routine's data set name FILTLIST to define the masks for the data sets. Then, test for this FILTLIST variable in your storage class ACS routine and assign a null storage class to ensure the data sets are created as non-system-managed.
4. Activate the new configuration.
5. Submit a DFSMSdss job to copy the data sets to the spare volume using parameters shown in Figure 67 on page 124. This job does not execute your ACS routines. Instead, a null (non-system-managed) storage class is assigned and the data sets are copied to the volume indicated by the OUTDYNAM parameter. BYPASSACS is a protected RACF facility. Ensure that you are authorized to use it.

```

COPY DS(INC(TSOHLQ.**)) -
OUTDYNAM(D65DM1) -
NULLSTORCLAS -
BYPASSACS(**)

```

Figure 67. Sample Fallback Procedure Using DFSMSdss

## Designing for TSO Performance and Availability Services

All TSO data sets are assigned the STANDARD storage class. If TSO data sets are allocated on volumes behind a cache-capable 3990 storage control, these data sets become may-cache data sets. If the cache is not overloaded from must-cache data sets, TSO data sets that are good cache candidates can use cache and DASD fast write.

No additional logic in the storage class ACS routine is required to assign TSO data sets to the STANDARD class. The OTHERWISE statement associated with the first SELECT statement is run for TSO data set allocations, setting the storage class to STANDARD. Refer to the storage class ACS routine displayed in Appendix B, “Sample Classes, Groups, and ACS Routines,” on page 243 for a coding example.

## Designing for TSO Data Backup and Space Management

The following two management classes must be defined for TSO data:

- EXTBAK
- INTERIM

EXTBAK defines the availability and space management attributes for long-term libraries that contain JCL, source program, CLIST, object, and load modules. Because the Partial Release attribute is set to CI (conditional immediate), any primary space that is allocated, but not used, is released immediately if a secondary space request is specified. EXTBAK retains these data sets on primary storage indefinitely if they are referenced at least once every 15 days, as dictated by the value of MIGRATE PRIMARY DAYS NON-USAGE. If 15 days elapse without the data set being referenced, DFSMSHsm moves the data set to migration level 1 where the data set exists for 60 days before being written to migration level 2. The attribute, Number Backup Versions, Data Set Exists, indicates that the five most recent versions of the data set are retained by DFSMSHsm on backup volumes as long as the data set exists.

INTERIM defines the space management attributes for short-term programmer listings. Because Partial Release is set to I (immediate), any primary space that is allocated, but not used, is released immediately, because these are transient data sets. DFSMSHsm deletes them after three days, regardless of their usage, based on the value of the Expire after Date/Days attribute. No backup versions are initiated, because Auto Backup is set to NO.

Table 6 shows the attributes for these two management classes.

Table 6. Management Classes for TSO Data

Attributes	Management Classes
------------	--------------------

Table 6. Management Classes for TSO Data (continued)

Name	EXTBAK	INTERIM
Expire after Days Non-usage	NOLIMIT	3
Expire after Date/Days	NOLIMIT	3
Retention Limit	NOLIMIT	3
Partial Release	COND IMMED	YES IMMED
Migrate Primary Days Non-usage	15	3
Level 1 Days Non-usage	60	60
Command or Auto Migrate	BOTH	BOTH
# GDG Elements on Primary	—	—
Rolled-off GDS Action	—	—
Backup Frequency	0	—
Number Backup Versions, Data Set Exists	5	—
Number Backup Versions, Data Set Deleted	1	—
Retain Days Only Backup Version	60	—
Retain Days Extra Backup Versions	30	—
Admin or User Command Backup	BOTH	—
Auto Backup	YES	NO
Backup Copy Technique	STANDARD	STANDARD

FILTLISTs define long-term programmer data and SYSPRINT output from programmer's jobs. These data types are identified by the value of the data set's LLQ. Figure 68 shows the FILTLIST section needed to identify the TSO data sets.

```

FILTLIST PGMRLIST          INCLUDE('SYSOUT',LIST*,OUT*)
FILTLIST PGMRDATA         INCLUDE('ASM',COB*,FOR*, 'CNTL', 'JCL', 'PLI',
                              'PL1', 'SCRIPT', 'TEXT', 'CLIST', 'SRC',
                              'SOURCE',LOAD*, 'PGMLIB', 'RESLIB')
    
```

Figure 68. FILTLISTs for TSO Data Used in Management Class ACS Routine

Figure 69 on page 126 shows the mainline logic needed to assign the management classes. When the LLQ of the data set being allocated satisfies one of literals or masks listed in the FILTLIST statement, the management class, EXTBAK or INTERIM, is assigned to the system-managed data set. Any TSO data sets that do not have LLQs matching either PGMRDATA or PGMRLIST are assigned the STANDARD management class. This is done by the last OTHERWISE clause in the management class ACS routine. See the management class ACS routine in Appendix B, "Sample Classes, Groups, and ACS Routines," on page 243 for an example.



```

WHEN (&LLQ = &PGMRDATA)                /* Backup application */
DO                                        /* development data */
  SET &MGMTCLAS = 'EXTBAK'              /* frequently */
  EXIT
END
WHEN (&LLQ = &PGMRLIST)                 /* Give 'short life' to */
DO                                        /* programmer listings */
  SET &MGMTCLAS = 'INTERIM'
  EXIT
END

```

Figure 69. Management Class ACS Routine for TSO Data

## Designing the Physical Storage Environment

No storage groups are defined or modified to manage TSO data. TSO data is assigned to PRIME<sub>xx</sub> or LARGE<sub>xx</sub> storage groups, based on the size of the data set. Data sets larger than 285MB are directed to the LARGE<sub>xx</sub> storage group.

No additions to the storage group ACS routine logic are required to support TSO data.

## Placing TSO Data under System Management

To convert TSO data to system management, you can choose either in-place conversion or movement of the TSO data to system-managed volumes.

### In-Place Conversion

If your TSO data is currently pooled and you are satisfied with the overall performance of the TSO workload, you can convert the volumes in-place using DFSMSdss CONVERTV. The CONVERTV command processes volumes and evaluates the eligibility of the volumes and the data sets on the volume to be system-managed. If the volume and data sets are eligible for system management, your storage class and management class ACS routines are run to assign storage classes and management classes to all the data sets on the volume. Figure 70 shows a sample CONVERTV operation.

```

//CONVV EXEC PGM=ADDRSSU,REGION=4096K
//*-----*
/* JOB : DSS CONVERTV *
/* NOTE: Sample DSS Job to Convert TSO *
/* data in place *
/*-----*
//SYSPRINT DD SYSOUT=*
//D65DM1 DD VOL=SER=D65DM1,UNIT=3390,DISP=SHR
//SYSIN DD *

CONVERTV -
SMS -
DDNAME(D65DM1)

```

Figure 70. Sample TSO Data Conversion In-Place

## Using Data Movement

DFSMSDss performs conversion with data movement using the COPY or DUMP/RESTORE operations. The commands call your storage and management class ACS routines to determine the SMS classes to assign to your data sets. Those that are assigned storage classes are moved to volumes in the PRIMExx or LARGExx storage groups, based on the size of the data sets.

Figure 71 is a sample of a job that calls DFSMSDss to move data sets from a non-system-managed volume to system-managed volumes determined by your storage group ACS routine. In this example, the source data resides in TSO pools so the TSO source volume, D65DM1, is specified as the primary DFSMSDss filter in the LOGINDYNAM parameter. All data sets on the source volume are moved, excluding a group of system programmer data sets having the high-level qualifier, SYS1. If your ACS routines determine that any data set should not be system-managed, the data set is moved to the non-managed volumes that you specify in the OUTDYNAM parameter. In this example, D65DM2 is the target volume for all non-system-managed data sets.

```
/*-----*
/* JOB : DSS COPY *
/* NOTE: Sample DSS Job to Convert a TSO *
/* data to system-managed storage *
/*-----*
//COPY EXEC PGM=ADDRSSU,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

COPY DATASET(INCLUDE(**) EXCLUDE(SYS1.**)) -
LOGINDYNAM(D65DM1) -
OUTDYNAM(D65DM2) -
ALLDATA(*) -
ALLEXCP -
CATALOG -
SPHERE -
DELETE -
PURGE -
TGTALLOC(SOURCE) -
TOLERATE(IOERROR) -
WAIT(2,2)
```

Figure 71. Sample TSO Data Conversion with Movement

If you want DFSMSDss to allocate the TSO data sets with the minimum space required to contain the data, omit ALLEXCP and ALLDATA. Also, use TGTALLOC(TRACK) to ensure that the minimum space is allocated for data sets that were originally allocated in cylinders.

Review the following parameters to determine which are applicable to placing your TSO data sets under system management:

- ALLEXCP should be coded only if you want to copy all the space allocated for a data set created by EXCP.
- ALLDATA should be coded only if you want to copy all the space allocated from the source to the target data set.
- TGTALLOC(SOURCE) should be coded if you want to preserve the original allocation unit of the source data set, such as cylinders or tracks.

If you code ALLDATA and ALLEXCP, review the effect of these parameters, based on the data set organization of the data sets being moved and the type of target device.

**Related Reading:** For more information about using ALLDATA and ALLEXCP, see the DFSMSdss section of *z/OS DFSMSdfp Storage Administration*.

---

## Automating Data Allocation Using Data Classes

TSO data set naming conventions are based on the high-level qualifier, typically the TSO user ID. You can use the data set's LLQ to identify the type of data. Some examples of LLQs are:

**PLI** for PL/1 source programs

**CLIST** for CLIST libraries

**CNTL** for JCL libraries

**TEXT** for TEXT data

**LOAD**  
for load libraries

**LIST** for listing data sets

The starter set assumes that you have a data set naming convention based on LLQ. Sample data classes are included in the starter set to help you define the following:

- VSAM data sets based on the RECORG parameter

With these data classes, users can create key-sequenced, relative record, entry-sequenced, or linear VSAM data sets using batch JCL run in the background, or TSO ALLOCATE commands run in the foreground. These data classes create VSAM data sets with a primary allocation of 400 KB. The user supplies the information about key offset and length for VSAM key-sequenced data sets. Any performance-related options are the user's responsibility to provide.

Table 7 on page 129 shows the attributes for these sample VSAM classes.

- Simple physical sequential data sets including test data sets and output listings.

You can create 80-byte, fixed-block data sets by using the class DATAF. The primary space requested is 400KB.

Data sets having variable-blocked records with a average record size of 255 bytes are defined using DATAV. Based on the primary space request, 1.275 MB are allocated.

Listing data sets having a primary space of 90 MB are allocated using the LISTING data class.

Table 8 on page 129 shows the attributes for these physical sequential data classes.

- Load and source libraries in both PDS and PDSE format

The data class, LOADLIB, is reserved for load libraries that you intend to allocate as partitioned data sets. SRCFLIB and SRCVLIB are data classes that allocate PDSEs, based on the value of the DATA SET NAME TYPE attribute.

Table 9 on page 129 shows the attributes for these model libraries.

## Sample Data Classes for TSO Data

Table 7 on page 129 shows the data class attributes for VSAM data sets based on the RECORG parameter:

Table 7. Data Classes for VSAM Data Sets

Attributes	VSAM Data Set Data Classes			
Name	DIRECT	ENTRY	KEYED	LINEAR
Recorg	RR	ES	KS	LS
Keyoff	—	—	0	—
Space Avgrec	U	U	U	U
Space Avg Value	4096	4096	4096	4096
Space Primary	100	100	100	100
Space Secondary	100	100	100	100
Volume Count	1	1	1	1

Table 8 shows the data class attributes for simple physical sequential data sets including test data sets and output listings:

Table 8. Data Classes for Physical Sequential Data Sets

Attributes	DASD Physical Sequential Data Set Data Classes		
Name	DATAF	DATAV	LISTING
Recfm	FB	VB	VBA
Lrecl	80	255	137
Space Avgrec	U	U	U
Space Avg Value	80	255	137
Space Primary	5000	5000	2000
Space Secondary	5000	5000	2000
Volume Count	1	1	1

Table 9 shows data class attributes for load and source libraries in both PDS and PDSE format:

Table 9. Data Classes for Libraries

Attributes	Partitioned Data Set Data Classes		
Name	LOADLIB	SRCFLIB	SRCVLIB
Recfm	U	FB	VB
Lrecl	—	80	255
Space Avgrec	U	U	U
Space Avg Value	23476	80	255
Space Primary	50	5000	5000
Space Secondary	50	5000	5000
Space Directory	62	62	62
Data Set Name Type	PDS	LIBRARY	LIBRARY
Volume Count	1	1	1

**Recommendation:** Allow users to assign a data class externally through batch JCL, TSO ALLOCATE commands, or using the ISPF/PDF enhanced allocation

application. The sample data class routine for the Managing Permanent Data milestone lets users assign a data class if it is valid.

The data class ACS routine for DASD data sets has three segments: one for externally-requested data classes, one for VSAM data sets and the other for non-VSAM data sets. The data class ACS routine performs the following tasks:

- Checks for externally-requested data classes and assigns it to the data set if the data class is part of your active SMS configuration. If it is not, no data class is assigned.
- Assigns the appropriate VSAM data class, if the DCB characteristic, RECORG, for the data set indicates that it is a VSAM data set. Otherwise, the LLQ of the data set name for any non-VSAM data set allocations is compared with the FILTLIST variables for each of the data types. If it matches, the data class associated with the variable is assigned.

### **Example: ACS Routine Fragments for Assigning Data Classes**

Figure 72 on page 131 shows the data class ACS routine fragment necessary to assign the data classes for TSO data sets.

---

```

FILTLIST DASD          INCLUDE('3380','3390','SYSDA','')
FILTLIST DATAF       INCLUDE('DATA','FDATA','SYSGO','SYSLIN',OBJ*)
FILTLIST DATAV       INCLUDE('TEXT','VDATA')
FILTLIST LIST         INCLUDE('SYSOUT',LIST*)
FILTLIST LOAD         INCLUDE('RESLIB','PGMLIB',LOAD*)
FILTLIST SRCF         INCLUDE(COB*,FOR*,CNTL*,JCL*)
                       EXCLUDE(**.SPFTEMP%.CNTL)
FILTLIST SRCV         INCLUDE('PLI','PL1','SCRIPT','CLIST')
FILTLIST VALID_DATA_CLASS
                       INCLUDE('DATAF','DATAV','DBARCLOG','DBICOPY',
                                'DIRECT','ENTRY','GDGF80','GDGV104',
                                'KEYED','LINEAR','LISTING','LOADLIB',
                                'SRCFLIB','SRCVLIB')
/***** End of FILTLIST Definitions *****/
/***** Start of Mainline SELECT *****/
SELECT                /* Start of mainline Select*/
  WHEN (&DATACLAS EQ &VALID_DATA_CLASS &&
        &UNIT EQ DASD) /* Allow users to specify */
    DO                /* data class for any */
      SET &DATACLAS = &DATACLAS /* DASD allocation */
    EXIT
  END
/* This code segment handles DASD allocations. First, VSAM data */
/* sets are assigned a data class based on type of organization. */
/* Then, selected non-VSAM data sets are assigned a data class based */
/* on data set name. Finally, data sets that are not identified */
/* and associated with a specific data class are assigned a "null" */
/* data class. These data sets rely on external specifications */
/* for successful allocation. */
/***** Start DASD Data Set Mainline *****/
WHEN (&UNIT = &DASD)
  DO
    SELECT (&RECORDG) /* Start of VSAM SELECT */
      WHEN ('ES')      /* "ENTRY" for a VSAM ESDS */
        DO
          SET &DATACLAS = 'ENTRY'
          EXIT
        END
      WHEN ('RR')      /* "DIRECT" for a VSAM RRDS*/
        DO
          SET &DATACLAS = 'DIRECT'
          EXIT
        END
      WHEN ('KS')      /* "KEYED" for a VSAM KSDS */
        DO
          SET &DATACLAS = 'KEYED'
          EXIT
        END
      WHEN ('LS')      /* "LINEAR" for a VSAM LDS */
        DO
          SET &DATACLAS = 'LINEAR'
          EXIT
        END
      OTHERWISE
        DO

```

---

Figure 72. Assigning a Data Class for VSAM Processing

Figure 73 on page 132 shows the data class ACS routine fragment necessary to assign the data classes, based on the data set name's LLQ.

**Related Reading:** For a full description of data set naming conventions, see *z/OS MVS JCL Reference*.

---

```

/***** Assign a data class based on the DSN's low-level qualifier *****/
SELECT (&LLQ)          /* Start of non-VSAM SELECT */
  WHEN (&LOAD)         /* "LOADLIB" to load libraries */
  DO                   /* that must be in PDS format */
    SET &DATACLAS = 'LOADLIB'
    EXIT
  END
  WHEN (&SRCF)         /* "SRCFLIB" to source libraries */
  DO                   /* with fixed-length records */
    SET &DATACLAS = 'SRCFLIB'
    EXIT
  END
  WHEN (&SRCV)         /* "SRCVLIB" to source libraries */
  DO                   /* with variable-length records */
    SET &DATACLAS = 'SRCVLIB'
    EXIT
  END
  WHEN (&LIST)         /* "LISTING" to output listings */
  DO
    SET &DATACLAS = 'LISTING'
    EXIT
  END
  WHEN (&DATAF)        /* "DATAF" to data sets with */
                        /* length records and program */
  DO                   /* records and program object */
    SET &DATACLAS = 'DATAF'
    EXIT
  END
  WHEN (&DATAV)        /* "DATAV" to data sets with */
                        /* variable-length records */
  DO
    SET &DATACLAS = 'DATAV'
    EXIT
  END
  OTHERWISE
  DO
    SET &DATACLAS = ' '
    EXIT
  END
  END
  END /* End of non-VSAM SELECT */
  END /* End of DASD Select */
/***** End of DASD Data Set Mainline *****/
OTHERWISE /* Set data class to null */
  DO /* value for all other */
    SET &DATACLAS = ' ' /* data sets and devices */
    EXIT
  END
  END /* End of Mainline Select */
/***** End of Mainline SELECT *****/
END /* End of data class routine */

```

---

Figure 73. Assigning a Data Class Based on the Low-Level Qualifier

---

## Defining SMS Constructs for HFS Data

You can specify a value of HFS as the Data Set Name Type attribute on the ISMF Data Class Define and Data Class Alter panels. The Data Class List panel indicates whether selected data sets are HFS data sets.

HFS data sets should have a separate data class, and should be placed in a primary (PRIMExx) storage group.

ACS routines allow the HFS value as long as the &DSNTYPE read-only variable is also provided.





---

## Chapter 8. Managing Batch Data

This chapter describes a process for migrating your batch data to system management. You can migrate your batch data to system management to improve the performance of jobs running in your batch window, and to simplify the backup and recovery procedures for data sets used by batch jobs.

---

### Understanding the Benefits of Placing Batch Data under System Management

Batch data is data that is processed on a regular basis, usually as part of a production cycle. A majority of these data sets are sequential data sets that are members of generation data groups.

You can readily migrate batch data to system management, but you need to understand application cycles to properly define the availability requirements of the data sets.

You can realize these benefits by migrating batch data to system management, as follows:

- You can improve batch job performance for I/O-bound jobs with sequential and VSAM data striping.

Jobs that process large, sequential or VSAM data sets can improve performance if you convert these data sets to extended-format data sets that are striped. Sequential and VSAM data striping causes data sets to be written across multiple DASD volumes on unique storage paths and then read in parallel on each volume.

Channel load and storage space can also be reduced if you use host-based data compression.

- You can improve batch job performance by moving tape data sets to DASD. Batch jobs constrained by tape access speed or drive availability can benefit from system management. Sequential access speed of striped data sets on DASD is faster than that of tape, and you spend no time with tape mounts. Production data sets and backups of production data sets both benefit from system management:

- Production data sets

Migrating this data to system management gives you an opportunity to improve performance for batch applications by moving selected data sets to DASD.

- Backup copies of production data sets

Data sets that are application *point-in-time* backups can be written to a system-managed DASD buffer and managed with DFSMSHsm according to the application's space and availability requirements. This strategy is discussed in Chapter 11, "Optimizing Tape Usage," on page 177.

- Critical batch data sets benefit from dual copy and from RAID technology.

Unrecoverable I/O errors can cause reruns. A cache-capable 3990 model storage control's dual copy capability can diminish the effect of hardware outages by maintaining a secondary copy of the data set on a separate volume. An I/O error on the primary volume causes the system to switch automatically to the secondary copy without any disruption to the application. If you use DASD fast

write with dual copy, there is no performance penalty for this increased availability because the I/O is complete when written to the cache-capable 3990 storage control's non-volatile storage. The update is then destaged to the primary and secondary volumes.

- Batch data sets benefit from DFSMSHsm's fully automatic availability management.

Most applications' availability requirements are met with the STANDARD management class. Applications with specialized cycles, such as batch data sets, might require unique management classes. Using data set-oriented management classes lets you customize services, based on each application's requirements. Later, you can simplify or eliminate application-initiated backup and recovery procedures.

SMS allocates batch data in the PRIMExx and LARGExx storage groups, based on data set size. Data sets larger than 285 MB are directed to the LARGExx storage group.

Most data sets are allocated using system-determined block size to optimize space usage. SMS allocates large, sequential batch data sets having high-performance requirements in extended format. Critical data sets are maintained on dual copy volumes.

For sequential data sets, SMS writes a hardware EOF at the beginning of the data set at initial allocation. This prevents data integrity problems when applications try to read the data before data is written in the data set.

You can manage the majority of your data with the STANDARD management class. Data sets having unique management requirements are identified by data set name or RACF &APPLIC and managed using a specialized management class. Generation data sets are identified by the ACS variable, &DSTYPE, and receive special management, based on the nature of the generation data group. If the generation data group contains backups of data sets, the current copy migrates quickly to migration level 1. In contrast, if it represents the current version of the production data set, the current version is retained on primary storage until the next generation is created. Output data sets containing reports are early candidates for movement to migration level 1.

Batch data sets can vary greatly from cycle to cycle. Certain data sets should not have space released automatically because of this variability. However, batch data sets that are generation data sets are assigned a management class causing unused space to be released automatically when the data set is closed.

---

## Planning Your Migration Strategy

Consider the following issues before you migrate batch data:

- The applicability of sequential data striping for your batch data
- The use of pattern DSCBs by your batch jobs
- The technique that you use to migrate your batch data
- Restart considerations in jobs that use GDGs

---

## Improving Batch Performance by Using Sequential Data Striping

You should select the data sets that benefit from sequential data striping based on your knowledge of how batch jobs use these data sets. Use the following criteria to select your candidates for sequential data striping:

- Select large, physical sequential data sets  
The VTOC must be accessed for each volume that contains a part of the striped data set. There is a slight performance penalty because of this additional processing that is not offset for small data sets.
- Select data sets processed using BSAM or QSAM only (no EXCP)
- Select data sets for jobs that are I/O-bound  
Read the RMF™ Device Activity Report statistics to find how long the jobs were executing. Volumes with high device CONNECT times are candidates for sequential data striping. Use caution in interpreting these statistics. If the high CONNECT time results from I/O contention because of other data sets on the volume, sequential data striping cannot improve performance. Therefore, it is beneficial to have a separate storage group for striping. It is also beneficial to have volumes in this storage group from as many different storage controls and serially connected channels as possible. A single storage control supports up to four paths. Therefore, you can have up to four volumes per data set per storage control.
- Select data sets used by applications that can coexist with a system-determined block size  
Blocks for striped data sets contain additional control information. Applications and JCL might require changes to block size and buffer size specifications to ensure that striped data sets are efficiently stored on DASD.

**Tip:** You can use the DFSMS Optimizer Function to help you select data sets that can benefit from striping.

---

## Eliminating Dependencies on Pattern GDGs

Pattern DSCBs must be replaced by data classes before batch data sets that are generation data sets are migrated to system management. Pattern DSCBs violate the SMS data set catalog requirement.

To replace pattern DSCBs, create data classes in your SMS configuration to describe the DCB characteristics. Use the parameter, DATACLAS, on the DD statement to generate the proper DCB characteristics for the generation data sets.

---

## Deciding on a Migration Strategy

Migrating batch data to system management usually requires migrating with data movement. If you want to improve performance through the use of sequential data striping, use DFSMSdss to move the data sets to the PRIMExx and LARGExx storage groups, and allocate those that benefit from sequential data striping on volumes behind a cache-capable 3990 storage control that is serially attached to your processor. Migrating the data to system management with movement enables the data sets to be distributed over the storage groups, based on size, performance, and availability needs.

If you must temporarily migrate back on an application basis, consider any volume-specific JCL that exists for the application. Either update the JCL to eliminate this dependency, or retain a spare volume initialized with the old volume serial number.

## Designing for Batch Performance and Availability Services

Review your batch workload for critical I/O-bound jobs that use physical sequential data sets. These are long-running jobs that are increasingly difficult to complete within your batch window. Consider using sequential data striping for the data sets that cause bottlenecks for these applications. You must estimate the rate that data must be read or written by the application and express this rate in MB per second. Place the rate in the SUSTAINED DATA RATE storage class attribute. If there is great variability in required I/O rates, you need multiple storage classes.

The FASTSEQ storage class is an example of a storage class that allocates a striped data set. For FASTSEQ, the value of SUSTAINED DATA RATE, 9 MB/sec, causes the data set to be spread over two 3390 volumes because the data transfer rate for a 3390 volume is 4.2 MB/sec. When you use the SUSTAINED DATA RATE to create striped data sets automatically, you must also create a data class for striped data sets. An example of a data class describing a striped data set is shown in Table 12 on page 142.

Data sets that must be available for critical applications should be assigned the CRITICAL storage class. The dual copy feature then maintains multiple copies of the data set. If an error occurs in the primary data set, the secondary copy is automatically used by the application.

Backup generation data sets usually do not have special performance or availability requirements. These are assigned the STANDARD storage class.

Table 10 shows the storage classes that are useful for batch data sets.

*Table 10. Storage Classes for Batch Data*

Attributes	Storage Classes			
	STANDARD	MEDIUM	FASTSEQ	CRITICAL
Name	STANDARD	MEDIUM	FASTSEQ	CRITICAL
Direct Millisecond Response	—	10	—	10
Sequential Millisecond Response	—	10	—	10
Availability	STANDARD	STANDARD	STANDARD	CONTINUOUS
Accessibility	STANDARD	STANDARD	CONTINUOUS	CONTINUOUS
Guaranteed Space	NO	NO	NO	NO
Guaranteed Synchronous Write	NO	NO	NO	NO
Sustained Data Rate	—	—	9.0	—

Generation data sets can be identified by testing the ACS variable, &DSTYPE. It is set to GDS for all generation data sets. The backup generation data sets are identified by the second-level qualifier.

Batch data is migrated on an application-by-application basis. To control the migration, define a FILTLIST in the storage class ACS routine to identify the data sets that should be excluded from system management. You could also use RACF &APPLIC to control this list of eligible applications if data set names are too cumbersome to summarize in filters. As you successfully migrate data sets for an application, remove one or more other applications from the list until the list is empty and all batch data has been migrated.

---

## Designing for Batch Data Backup and Space Management

Generation data sets have predictable backup and availability requirements in comparison to other batch data sets. We provide guidelines for developing management classes for both sets of batch data.

### Managing GDGs

Manage GDG data sets according to their type, either backup or production. Migrate backup generation data sets to migration level 1 and the older versions to migration level 2. The backup GDS can be recalled if required. Keep the current generation of production generation data sets on primary storage until the next is created.

There are two management classes you can assign to generation data sets. GDGPROD and GDGBKUP are applicable to production and backup GDSs, respectively. Table 11 lists the attributes for the GDG data classes.

*Table 11. Management Classes for Batch Data*

Attributes	Management Classes			
	GDGBKUP	GDGPROD	STANDARD	MONTHMIG
Name	GDGBKUP	GDGPROD	STANDARD	MONTHMIG
Expire after Days Non-usage	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
Expire after Date/Days	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
Retention Limit	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
Partial Release	YES IMMED	YES IMMED	NO	NO
Migrate Primary Days Non-usage	2	15	15	35
Level 1 Days Non-usage	0	60	60	70
Command or Auto Migrate	BOTH	BOTH	BOTH	BOTH
# GDG Elements on Primary	1	1	—	1
Rolled-off GDS Action	EXPIRE	EXPIRE	EXPIRE	EXPIRE
Backup Frequency	—	0	—	0

Table 11. Management Classes for Batch Data (continued)

Attributes	Management Classes			
Number Backup Versions, Data Set Exists	—	1	—	1
Number Backup Versions, Data Set Deleted	—	1	—	1
Retain Days Only Backup Version	—	60	60	120
Retain Days Extra Backup Versions	—	30	30	60
Admin or User Command Backup	NONE	BOTH	BOTH	BOTH
Auto Backup	NO	YES	YES	YES
Backup Copy Technique	STANDARD	CONCURRENT PREFERRED	STANDARD	CONCURRENT PREFERRED

## Managing Non-GDG Batch Data Sets

Because production cycles vary greatly, consider their characteristics carefully when you plan for automatic management by DFSMSHsm. The MONTHMIG management class handles batch application data sets generated during a monthly cycle. Table 11 on page 139 lists its attributes.

Few characteristics of batch data sets can be generalized. Consider these variances before you automate space and availability management with DFSMSHsm:

- Batch data sets might be inactive for a long time, but when needed, must be available immediately. This type of data set should not be migrated to tape.
- Batch data set sizes might vary greatly, based on the production cycle. These data sets should be assigned a management class with Partial Release=NO to inhibit DFSMSHsm from releasing space.
- Batch data sets might only be needed for a short period. These data sets might consist of reports or error listings, and be eligible for early deletion.

You should supplement the basic set of management classes with others that reflect the data set requirements for specialized applications.

If you have a large, complex batch workload and want to use SMS performance services without analyzing the management requirements of the data, you can migrate these data sets to system management and assign a management class that does not cause any automatic space or availability management actions to occur. Always assign a management class, because if one is not assigned, the default management class attributes or DFSMSHsm defaults are used. Once you have migrated your batch data, you can design the management classes and the management class ACS routines to accommodate batch data. You can assign the management classes for batch by executing DFSMSDss's CONVERTV, using the REDETERMINE option for volumes containing batch data.

The sample management class routine detects generation data sets using the DSTYPE read/only variable. It then assigns a management class, depending on whether the generation data set is production or backup. A data set naming standard that uses the second-level qualifier to convey whether a data set is a backup or production GDS is assumed in the sample ACS routine. Figure 74 shows the management class ACS routine logic for batch data sets. As with other data types, a large percentage of batch data is assigned the STANDARD management class.

### Example: Management class ACS routine fragment for batch data

```

WHEN (&DSTYPE = 'GDS' &&
      &DSN(2) = *BK*)
  DO
    SET &MGMTCLAS = 'GDGBKUP'
    EXIT
  END
WHEN (&DSTYPE = 'GDS' &&
      &DSN(2) = *PR*)
  DO
    SET &MGMTCLAS = 'GDGPROD'
    EXIT
  END
OTHERWISE
  DO
    SET &MGMTCLAS = 'STANDARD'
    EXIT
  END

```

/\* Manage generation data \*/  
 /\* putting rolled-off \*/  
 /\* versions on MLI; do not\*/  
 /\* backup \*/

/\* Give normal data sets \*/  
 /\* medium migration and \*/  
 /\* backup services \*/

Figure 74. Management Class ACS Routine Fragment for Batch Data

## Designing the Physical Storage Environment

No new storage groups are defined to manage batch data. You should add volumes to LARGE<sub>xx</sub> and PRIMExx storage groups to contain the batch data sets. If some volumes are reserved to contain data sets assigned the CRITICAL class, you must prepare volumes to be primary and secondary dual copy volumes. Chapter 3, “Enabling the Software Base for System-Managed Storage,” on page 67, describes how you can use ISMF to do this.

You do not need to add logic to the storage group ACS routine for batch data.

## Designing Data Classes for Batch Data

Data classes assist you in replacing generation data sets and defining striped data sets.

### Replacing Pattern DSCBs

Data classes help you migrate generation data sets to the system-managed environment. They represent one way of replacing the unsupported pattern DSCBs. One class, GDGV104, is defined for data sets having variable length records and another class, GDGF80, supports data sets with fixed length records. An average primary space allocation of 800 KB is requested for data sets assigned the GDGF80 data class, and 520 MB is requested for GDGV104. As with other data classes, the



user can supply other DCB parameters to supplement or override the ones in the assigned data class. The attributes for GDGF80 and GDGV104 are listed in Table 12.

Users can assign the data classes created to replace the GDG pattern DSCBs by specifying the data class, using the DATACLAS parameter on the DD statement for the GDS. Or, you can automate assignment of data classes by testing the &DSTYPE variable for the value GDS and the data set name.

## Defining Sequential Data Striping

The data class determines if the data set should be allocated as striped. Use the Data Set Name Type attribute to specify the requirement for this performance service. You can require data set striping by setting Data Set Name Type to (Extended=R), or conditionally request data set striping by setting the attribute to (Extended=C). In the first case, allocation for the data set fails if the required space behind a cache-capable 3990 storage control is not available. In the second case, if space is not available, the data set is allocated as a physical sequential data set in non-striped sequential format. Table 12 shows the attributes for the FASTBAT data class, which sets a mandatory performance requirement for data sets that are assigned to it.

Table 12. Sample Data Classes for Striped and Generation Data Sets

Attributes	Generation Data Group Striped		
Name	GDGF80	GDGV104	FASTBAT
Recfm	F	V	V
Lrecl	80	104	—
Space Avgrec	K	M	M
Space Avg Value	80	104	250
Space Primary	10	5	60
Space Secondary	20	2	2
Data Set Name Type	—	—	(EXTENDED,R)
Retpd or Expdt	—	—	—
Volume Count	1	1	—

---

## Migrating Batch Data

The migration with data movement strategy depends on being able to identify the data sets for a specific application by the data set name. In the sample job, the HLQ is used to identify the data set.

The data is moved to the PRIME<sub>xx</sub> and LARGE<sub>xx</sub> storage groups, based on the size of the data sets using DFSMSdss COPY and the source volumes that the application's data resides on as another filter. Figure 75 on page 143 shows sample JCL that you can use to migrate the first application's data to system management. The sample job uses the following parameters:

- The LOGINDYNAM(D65DM2,D65DM3) parameter tells DFSMSdss to move all data sets on volumes D65DM2 and D65DM3.
- The DATASET(INCLUDE(1STAPPL.\*\*)) indicates that, for the volumes selected, only data sets having the 1STAPPL HLQ are copied.

After you successfully migrate the first, modify the data set and volume filters to migrate other applications.

You cannot migrate to striped data sets with DFSMSdss. You can exclude these data sets from the initial COPY and then recreate them as striped data sets through the use of standard z/OS utilities, such as access method services or IEBGENER.

```
COPY DATASET(INCLUDE(1STAPPL.**)) -  
  LOGINDYNAM(D65DM2,D65DM3)      -  
  ALLDATA(*)                       -  
  ALLEXCP                          -  
  CATALOG                           -  
  SPHERE                             -  
  DELETE                             -  
  PURGE                              -  
  TGTALLOC(SOURCE)                 -  
  TOLERATE(IOERROR)                -  
  WAIT(2,2)
```

*Figure 75. DFSMSdss Job Migrating Batch Data to System Management*

If you temporarily remove data sets for a batch application from system management, you can still use the DATACLAS parameter to supply the data set characteristics. Data classes apply to both system-managed and non-system-managed data sets.



---

## Chapter 9. Managing Database Data

This chapter describes how you can migrate your CICS, IMS, or DB2 database data to system management. Each database management system has unique data sets and facilities to support its online environment. These differences will change the recommended storage management procedures. Database data has diverse space, performance, and availability requirements; however, dividing your database data into categories helps you identify the required SMS services and implement a staged migration to system-managed storage.

There are five major categories of database data:

- Production databases
- End-user databases
- Recovery data sets
- System data sets
- Test data bases

---

### Understanding the Benefits of Placing Your Database Data under System Management

Database data benefits from system management in the following ways:

- Database data can benefit from data isolation achieved through the use of SMS storage groups.

SMS services help you automate allocation of production database data sets (as well as production data bases from end user and test data bases) on separately managed volumes. You should isolate database data on separate volumes because of its high performance and availability requirements, its specialized backup and recovery management procedures, and its migration and expiration criteria.

- Both production and end-user databases can benefit from the improved performance of enhanced dynamic cache management.

Using SMS dynamic cache management lets you specify a hierarchy of performance for database data, based on your knowledge of the online applications' use of databases and the requirements of database data types, such as recovery or system data sets. If you are already using volume-level caching and the extended functions of a cache-capable 3990 storage control, you can use the data set-level caching provided by SMS dynamic cache management to improve performance and increase automation of performance management.

- Production databases can benefit from improved SMS allocation algorithms.

SMS tries to balance allocations in (and across) storage groups that have large numbers of volumes. It also uses a randomization technique for volume selection from the secondary volume list in order to avoid skews resulting from the addition of new volumes, or owing to differences in workload between the time of allocation and the time of actual use. With SMS allocation, users do not have to worry about specifying volume serial numbers or ensuring that the specified volumes have adequate free space. Additionally, with SMS allocation algorithms, volumes are not fragmented too much for the request for space to be satisfied. While some production databases might require special placement of data for critical performance, or for separation of software striped data (as in DB2), all other data should benefit from SMS allocation algorithms.

- Database data backup processes can benefit from point-in-time copy using either concurrent copy or virtual concurrent copy.

For concurrent copy, virtual concurrent copy, or flash copy, implement this high-performance backup with DFSMSDss and a cache-capable 3990 storage control. It is supported by CICS, IMS, and DB2. With concurrent copy, you are only required to quiesce the affected databases briefly, instead of shutting down the entire online system during the backup process.

Virtual concurrent copy support is provided through IBM RAMAC Virtual Array devices with the SnapShot feature. With virtual concurrent copy, you create a "fast" point-in-time version of a data set so that it is unavailable for normal application processing for a very minimal period of time. The version is then available for application testing, reporting, or backup operations.

- End-user and test databases can benefit from the automatic availability management provided by DFSMSHsm.

Database management systems, except for CICS VSAM file control systems, typically maintain their own catalogs that contain allocation-related information about the data sets that support the online environment. Catalog accuracy is most critical when a database must be recovered. Because there is no automated facility to keep the database management system informed of backups done by DFSMSHsm, you should not use DFSMSHsm to back up production databases. Because end-user databases usually have less stringent recovery requirements, DFSMSHsm can effectively manage them. DFSMSHsm's automatic space management can migrate database data after a period of disuse and automatically recall it when needed by the user.

- Management of database data is improved by implementing standards enforcement through the use of data classes and the data class ACS routine.

Storage and database administrators can jointly develop data classes for commonly used types of database data. With data classes, you can standardize the allocation parameters used to create new database data sets. You can also enforce data set naming standards through the use of data class ACS routines.

- Use dual copy and RAID architecture to increase availability of critical data sets.

Some recovery data sets are duplexed by IMS and DB2 to provide greater reliability. You can use a cache-capable 3990 storage control's dual copy capability to provide protection against hardware outages for CICS, and extend availability to other IMS and DB2 data sets.

You can also use virtual concurrent copy support, through an IBM RAMAC Virtual Array device with the SnapShot feature, to create test data bases from production data bases instantaneously, without using additional storage resources. Virtual concurrent copy support lets you restart/rerun interrupted backups, which cannot be done using concurrent copy.

- Database applications using non-shared resources (NSR) can benefit from improved performance and elapsed times in applications that access the data directly and sequentially.
- Database applications that perform sequential data access can benefit from striping.
- Applications, such as data warehousing that uses large sizes, can benefit from extended addressability.

Table 13 on page 147 summarizes the SMS services for database data.

Table 13. SMS Services for Database Data

Data Type	Space Management	Performance Management <sup>1</sup>	Availability Management	Volume Selection
Production Databases	None	Cache, DFW	Concurrent copy for point-of-consistency backup	Yes
End-user Databases	Automatic migration	Cache, DFW	Automatic backup using concurrent copy	Yes
Recovery Data Sets	Automatic Migration	Cache, DFW	Dual copy	Yes
System Data Sets	None	Cache, DFW	Automatic backup using concurrent copy	Yes

1. DFW refers to the DASD fast write extended function of a cache-capable 3990 storage control.

You can design your ACS routines so that SMS restricts the allocation of data sets in CICS, IMS, and DB2 storage groups to production databases and selected system data sets. Only specially-identified users, such as the database or storage administrator can allocate data in these storage groups. Most data sets that support the database environment, including the recovery and system data sets, are directed to the PRIMExx storage group. The storage and database administrators have special SMS authority to assign data sets with critical performance and availability requirements to specific volumes. Dual copy and RAID technology provide high availability for selected data sets that are not duplexed by the database management system. Use DASD fast write and cache to provide superior performance for databases and recovery data sets.

DFSMS supplements the backup and recovery utilities provided by the database management system as follows:

- DFSMSdss uses concurrent copy capability and virtual concurrent copy support to create point-in-time backups.
- Data base utilities (except for CICS) invoke DFSMSdss for concurrent copy and virtual concurrent copy support for point-in-time backups and backup-while-open.
- DFSMShsm backs up system data sets and end-user database data that is less critical than production database data. You can use the backup-while-open function with concurrent copy or virtual concurrent copy support to back up CICS VSAM data sets while they are open for update.
- DFSMShsm carries out direct migration to migration level 2 for archived recovery data sets on DASD.
- End-user and test database data is migrated by DFSMShsm through the storage hierarchy, based on database data usage.

---

## Planning Your Migration

There are several major tasks for migrating database data to system management:

- Designing the storage classes and ACS routine
- Designing the management classes and ACS routine
- Designing the storage groups and ACS routine
- Testing your SMS configuration
- Activating the new SMS configuration
- Migrating the end-user databases and related database data sets
- Testing migrate/recall performance for end-user databases
- Migrating the production databases and related database data sets

For CICS, IMS, and DB2, you must ensure that any database data sets you plan to system-manage are cataloged using the standard search order. In particular, check image copies and logs to ensure that they are cataloged.

CICS data can benefit from compression, extended format, extended addressability, secondary volume space amount, and dynamic cache management enhancements when the data sets are KSDSs. Batch programs accessing this data can benefit from system-managed buffering.

For IMS data, consider converting any OSAM data sets to VSAM. By converting to VSAM, you can benefit from enhanced dynamic cache management. IMS Version 5 supports enhanced dynamic cache management for OSAM data sets. KSDSs being used by IMS can be extended format but cannot be compressed because IMS uses its own form of compression and cannot tolerate compression performed by DFSMS.

For DB2 data, complete the following tasks:

- Verify consistency between DB2 STOGROUPs and SMS storage groups. See “Relating DB2 STOGROUPs to SMS Storage Groups” on page 166 for more information.
- Ensure that SMS management class expiration attributes are synchronized with DB2’s expiration information:
  - Expiration of logs must be consistent with the value of ARCRETN. You should update the BSDS with the DB2 change log inventory utility.
  - Expiration of any DB2 index spaces or table spaces requires a proper SQL DROP.
  - Expiration of any DB2 image copies requires running the MODIFY utility to update SYSCOPY.
- Set DSNZPARM to have DFSMSshsm automatically recall DB2 data sets during DB2 access. Set RECALL to Y. Set RECALLD, the maximum wait for DFSMSshsm to complete re-creation of data sets on primary storage, based on testing with typical end-user database data sets.

---

## Designing for Database Performance and Availability

The following sections identify the performance and availability recommendations, by data type, for each database management system. Database data set name qualifiers, especially the low-level qualifier (LLQ), identify the type of data set in a FILTLIST variable that is later tested in the ACS routine to determine which SMS class should be assigned. The LLQs used in this chapter are the same as the ones in the sample start-up procedures for CICS Version 3 Release 3, IMS Version 4 Release 1, and DB2 Version 3 Release 1.

## Designing for CICS Data

Recovery data sets have high availability requirements because they must be available during restart if CICS fails. Some of these data sets also benefit from SMS services that deliver good response time. For example:

- Fault tolerant devices, such as dual copy and RAMAC, can be effectively used to protect recovery data sets from being lost because of a hardware outage.
- Some recovery data sets, such as system logs, have both stringent availability requirements and above average performance requirements because each CICS transaction updates the system logs. The logs can benefit from DASD fast write. Additionally, if two system logs are allocated, they should be placed on separate volumes. This requirement can be satisfied by assigning a storage class with guaranteed space.

System data sets, such as the CICS availability manager data sets, must also be available during restart and are good candidates for allocation on fault-tolerant devices.

- The CAVM message and control data sets require a storage class with guaranteed space, because they should be placed on different volumes.
- Use DASD fast write for intrapartition transient data to improve transaction response time.
- Transactions using auxiliary temporary data can benefit from using cache and DASD fast write.

Databases must be evaluated independently, based on their criticality, read, and write activity, to determine the required storage services.

Three tables have been developed to identify the types of CICS database data sets having high availability and performance requirements. Table 14 shows the relationship between the data set, data type, and LLQ that are used to identify the data sets having high availability requirements in the storage class ACS routine.

*Table 14. CICS Data Sets Requiring High Availability*

CICS data set	Data set type	Low-level qualifier
System logs	Recovery	DFHJ01A/B
Restart data set	Recovery	DFHRSD
CICS Availability Manager (CAVM)	Recovery	DFHXRMSG (message) DFHXRCTL (control)
CICS system definition	System	DFHCSD
Production databases	Databases	—

Table 15 shows the relationship between the data set, data type, and LLQ that identify data sets having high write activity in the storage class ACS routine.

*Table 15. CICS Data Sets Having High Write Activity*

CICS data set	Data set type	Low-level qualifier
System logs	Recovery	DFHJ01A/B
Intrapartition transient data	System	DFHINTRA
Auxiliary temporary storage	System	DFHTEMP
Production databases	Databases	—



Table 16 shows the relationship between the data set, data type, and LLQ that identify data sets having high read activity in the storage class ACS routine.

*Table 16. CICS Data Sets Having High Read Activity*

CICS data set	Data set type	Low-level qualifier
Program libraries	System	LOADLIB COB2CICS COB2LIB PLILINK
Auxiliary temporary storage	System	DFHTEMP
User databases	Databases	—

In the sample configuration, these are the storage class assignments:

**Class name**

**CICS data sets**

**DBCRT**

Restart data set, system logs, CICS system definition, CAVM data sets

**MEDIUM**

User databases

**FAST** Program libraries

**FASTWRIT**

Intrapartition transient data, auxiliary temporary storage, production databases

All other CICS data sets are assigned the STANDARD storage class. SMS storage classes assigned to these database data sets contain performance objectives and performance and availability requirements.

**Using the DBCRIT Storage Class**

SMS attempts to ensure that data sets assigned to the DBCRIT storage class are:

- Mandatory cache users, because direct and sequential millisecond response is 5 milliseconds
- Mandatory users of DASD fast write, because direct bias is W and cache use is required

SMS ensures that data sets assigned to the DBCRIT storage class are:

- Placed on fault-tolerant devices, because availability is CONTINUOUS
- Using point-in-time copy, because accessibility is CONTINUOUS

Using the Guaranteed Space attribute to specify volumes is not recommended for most of the data sets for the following reasons:

- SMS uses randomizing techniques to select volumes, which should satisfy most, if not all, allocations. The randomizing techniques tend to spread data sets across the available volumes in a storage group.
- With the IBM RVA and the ESS, multiple logical volumes can be mapped to a physical volume due to their RAID architecture, volume capacity, and, if applicable, their log structured array architecture.
- The IBM ESS has large cache structures and sophisticated caching algorithms. It is capable of providing a much larger throughput. Its capabilities of parallel access volume and multiple allegiance allow many concurrent accesses to the same data. Therefore, specific volume placement and data set separation used for performance reasons should no longer be required.

## Using the MEDIUM Storage Class

SMS attempts to give data sets assigned to the MEDIUM storage class above average performance, because direct and sequential millisecond response is 10 milliseconds.

SMS attempts to allocate these data sets behind a cache storage control and use dynamic cache management to deliver good I/O service.

## Using the FASTWRIT Storage Class

SMS attempts to give data sets assigned to the FASTWRIT storage class mandatory DASD fast write and cache services, because direct and sequential millisecond response is 5 milliseconds, and direct and sequential bias is W.

SMS ensures that data sets assigned to the FASTWRIT storage class have concurrent copy, because accessibility is CONTINUOUS.

## Using the FAST Storage Class

SMS ensures that data sets assigned to the FAST storage class have mandatory cache services, because direct and sequential millisecond response is 5 milliseconds and direct and sequential bias is R.

Refer to Table 23 on page 161 for the list of attributes associated with these storage classes.

## Designing the Storage Class Routine for CICS Data

The high-level qualifier (HLQ) identifies CICS data sets and the LLQ identifies data types requiring specialized performance and availability services. Each of the special service database data types are identified by a FILTLIST statement.

Figure 76 shows the FILTLIST section of the storage class ACS routine for CICS database data sets.

```
FILTLIST SPECIAL_USERS      INCLUDE('SYSPROG','STGADMIN','DBA')
FILTLIST VALID_STORAGE_CLASS INCLUDE('BACKUP','CRITICAL','FAST','FASTREAD',
                                     'FASTWRIT','GSPACE','MEDIUM','NONVIO',
                                     'STANDARD','DBCRT')

/*****
/*          Start of CICS-related FILTLISTS          */
/*****
FILTLIST CICS                INCLUDE(%CICS.***)
FILTLIST CICS_PROD_CAVM      INCLUDE(PCICS**.*.DFHXRMSG,
                                     PCICS**.*.DFHXRCTL)
FILTLIST CICS_PROD_DB        INCLUDE(PCICS**.*.EMPLOYEE.DB.*)
FILTLIST CICS_PROD_LIB       INCLUDE(PCICS**.*.LOADLIB,
                                     PCICS**.*.COB2CICS,
                                     PCICS**.*.COB2LIB,
                                     PCICS**.*.PLILINK)
FILTLIST CICS_USER_DB        INCLUDE(TCICS**.*.PAYEXTRT.DB.*)
FILTLIST CICS_PROD_LOG       INCLUDE(PCICS**.*.DFHJ01%)
FILTLIST CICS_PROD_RESTART   INCLUDE(PCICS**.*.DFHRSD)
FILTLIST CICS_PROD_CSD       INCLUDE(PCICS**.*.DFHCSD)
FILTLIST CICS_PROD_INTRA     INCLUDE(PCICS**.*.DFHINTRA)
FILTLIST CICS_PROD_TEMP      INCLUDE(PCICS**.*.DFHTEMP)

/*****
/*          End of CICS-related FILTLISTS          */
/*****
```

Figure 76. FILTLIST Section for CICS from Storage Class ACS Routine

Figure 77 shows the coding required to let database and storage administrators and system programmers assign a storage class externally. Additionally, this restricted set of users can use the artificial class, NONSMS, to exclude data sets from system management.

```

WHEN (&GROUP = &SPECIAL_USERS &&          /* Permit storage admin., */
      &STORCLAS = &VALID_STORAGE_CLASS) /* system programmers, */
DO                                           /* and DBAs to specify */
  SET &STORCLAS = &STORCLAS                /* a storage class      */
EXIT
END
WHEN (&GROUP = &SPECIAL_USERS &&          /* Permit storage admin. */
      &STORCLAS = 'NONSMS')                /* or data base admin. */
DO                                           /* to create             */
  SET &STORCLAS = ''                       /* non-system-managed   */
EXIT                                         /* data sets            */
END

```

*Figure 77. Segment to Permit Special Users to Override SMS allocation*

CICS database data sets are first identified by the HLQ, PCICS, UCICS, or TCICS. Selected system and recovery database data sets are identified and assigned a storage class having attributes set to provide the required services. In this sample routine, production databases are given DASD fast write and cache services. User databases receive better than average service and, if allocated behind a cache-capable 3990 storage control, become may-cache candidates. In keeping with our ACS coding recommendations, once a data type has been identified, it is assigned an appropriate storage class and then the routine is exited. Figure 78 on page 153 shows the coding for the CICS database data sets.

**Example: SELECT section for CICS from storage Cclass ACS routine:**

```

/*****
/*   Start of CICS Select
/*****
  WHEN (&DSN = &CICS)
  DO
    SELECT
      WHEN (&DSN = &CICS_PROD_CAVM OR
            &DSN = &CICS_PROD_RESTART OR
            &DSN = &CICS_PROD_CSD)
      DO
        SET &STORCLAS = 'DBCRT'
        EXIT
      END
      WHEN (&DSN = &CICS_PROD_TEMP OR
            &DSN = &CICS_PROD_LIB)
      DO
        SET &STORCLAS = 'FAST'
        EXIT
      END
      WHEN (&DSN = &CICS_PROD_INTRA OR
            &DSN = &CICS_PROD_DB)
      DO
        SET &STORCLAS = 'FASTWRIT'
        EXIT
      END
      WHEN (&DSN = &CICS_USER_DB)
      DO
        SET &STORCLAS = 'MEDIUM'
        EXIT
      END
    OTHERWISE
    DO
      SET &STORCLAS = 'STANDARD'
      EXIT
    END
  END

  END

/*****
/*   End of CICS Select
/*****
```

Figure 78. SELECT Section for CICS from Storage Class ACS Routine

## Designing for IMS Data

IMS database data sets benefit from the following availability services provided by SMS:

- Use of fault-tolerant devices, such as dual copy or RAMAC, for the message queue data sets can increase availability, because read or write errors for this data set cause IMS failure.
- Data sets that can be duplexed by IMS, such as the OLDS, WADS, and RECON, can use fault-tolerant devices for added protection.

Use of SMS guaranteed space supports IMS duplexing for system-managed data sets. The primary and secondary copies of the OLDS data sets can be placed on

separate volumes by using a storage class that has the Guaranteed Space attribute set to YES. Also, active logs and archive logs can be placed on different volumes using the same SMS service.

IMS supports two types of databases: DL/1 and DEDB. IMS can duplex DEDB databases; however, DL/1 databases are not duplexed. Use fault-tolerant devices to protect critical DL/1 databases.

Table 17 shows the relationship between the data set, data type, and LLQ that identify the data sets having high availability requirements in the storage class ACS routine. The data sets listed in Table 17 are not duplexed by IMS.

*Table 17. IMS Data Sets Requiring High Availability*

IMS data set	Data set type	Low-level qualifier
Restart Data Set (RDS)	Recovery	RDS
Message Queues Data Set (MSG Q)	Recovery	QBLKS SHMSG LGMSG
DL/1 databases	Databases	—

Several IMS data types benefit from DASD fast write. For example, Extended Recovery Facility (XRF) users, with the RECON and OLDS, benefit from DASD fast write because multiple systems use these database data sets. Also, consider using DASD fast write for the IMS scratchpad area (SPA), for IMS users with heavy conversational workloads.

The access reference pattern for the database can affect the caching or DASD fast write benefits for production databases. Databases having high update activity are candidates for DASD fast write.

**Recommendation:** Leave the direct and sequential millisecond storage class attributes blank. This lets DFSMS determine whether to cache or use DASD fast write for the database, based on how it is used. If you are assigning storage service attributes for a particularly critical database, you can ensure that it uses cache and DASD fast write by assigning low direct and sequential millisecond response times to override dynamic cache management.

Table 18 shows the relationship between the data set, data type, and LLQ that identify the data set in the storage class ACS routine.

*Table 18. IMS Data Sets Having High Write Activity*

IMS data set	Data set type	Low-level qualifier
Write Ahead Data Set (WADS)	Recovery	WADS0/1
Online Log Data Set (OLDS)	Recovery	OLPxx/OLSxx
Recovery Control Data Set (RECON)	Recovery	RECON
Scratchpad area (SPA)	System	SPA

IMS databases that are predominantly read can also benefit from cache services. Table 19 on page 155 summarizes this potential benefit for both DEDB and DL/1 databases.

Table 19. IMS Data Sets Having High Read Activity

IMS data set	Data set type	Low-level qualifier
Production databases	Databases	—

The following storage class assignments are in the sample configuration:

**Class name**

IMS data sets

**DBCRT**

Restart data set, message queue data set, and DL/1 databases

**FASTWRIT**

WADS, OLDS, RECON, SPA

**FAST DEDB**

All other IMS data sets are assigned the STANDARD storage class.

Refer to Table 23 on page 161 for the list of attributes associated with these storage classes.

### Designing the Storage Class Routine for IMS Data

The HLQ identifies IMS data sets and the LLQ identifies data types requiring specialized performance and availability services. Figure 79 shows the FILTLIST section of the storage class ACS routine that identifies IMS database data sets for special services.

```

FILTLIST SPECIAL_USERS      INCLUDE('SYSPROG','STGADMIN','DBA')
FILTLIST VALID_STORAGE_CLASS INCLUDE('BACKUP','CRITICAL','FAST','FASTREAD',
                                     'FASTWRIT','GSPACE','MEDIUM','NONVIO',
                                     'STANDARD','DBCRT')
/*****
/*      Start of IMS-related FILTLISTS      */
/*****
FILTLIST IMS                INCLUDE(%IMS*,**)
FILTLIST IMS_PROD_DL1      INCLUDE(PIMS*.PAYROLL.DB.*)
FILTLIST IMS_PROD_LMSG     INCLUDE(PIMS*.*.LGMSG)
FILTLIST IMS_PROD_OLDS     INCLUDE(PIMS*.*.OL*)
FILTLIST IMS_PROD_QUEUE    INCLUDE(PIMS*.*.QBLKS)
FILTLIST IMS_PROD_RECON    INCLUDE(PIMS*.*.RECON)
FILTLIST IMS_PROD_RESTART  INCLUDE(PIMS*.*.RDS)
FILTLIST IMS_PROD_SMSG     INCLUDE(PIMS*.*.SHMSG)
FILTLIST IMS_PROD_SPA      INCLUDE(PIMS*.*.SPA)
FILTLIST IMS_PROD_WADS     INCLUDE(PIMS*.*.WADS*)
/*****
/*      End of IMS-related FILTLISTS      */
/*****

```

Figure 79. FILTLIST Section for IMS from Storage Class ACS Routine

This is a common section run before processing IMS database data sets by type. It lets system programmers, database and storage administrators assign storage classes externally, and even allocate these data sets as non-system-managed. Figure 80 on page 156 shows the ACS code.

---

```

WHEN (&GROUP = &SPECIAL_USERS &&          /* Permit storage admin., */
      &STORCLAS = &VALID_STORAGE_CLASS) /* system programmers, */
DO                                           /* and DBAs to specify */
  SET &STORCLAS = &STORCLAS              /* a storage class      */
EXIT
END
WHEN (&GROUP = &SPECIAL_USERS &&          /* Permit storage admin. */
      &STORCLAS = 'NONSMS')              /* or data base admin. */
DO                                           /* to create */
  SET &STORCLAS = ''                      /* non-system-managed */
EXIT                                         /* data sets */
END

```

---

*Figure 80. ACS Code to Permit Special Users to Override SMS Allocation*

Figure 81 on page 157 shows the logic to process IMS database data sets by type. Only production database data sets are assigned specialized services in this routine. The first character of the data set's HLQ defines a production (P) or test (T) database data set. The three low-order characters of the HLQ are set to IMS. The DSN mask defined in the FILTLIST section for the data set type describes the data set type in the LLQ. Consider the following when deciding on your naming conventions:

- The first WHEN clause verifies that the data set belongs to an IMS system, by checking the FILTLIST variable, IMS
- Each uniquely named data set type is assigned the recommended storage class
- Any IMS production database data set not specifically identified, or any test IMS database data set, is assigned STANDARD storage services.

**Example: SELECT section for IMS from storage class ACS routine:**

```

/*****
/*   Start of IMS Select
/*****
  WHEN (&DSN = &IMS)
    DO
      SELECT
        WHEN (&DSN = &IMS_PROD_RESTART OR
              &DSN = &IMS_PROD_QUEUE OR
              &DSN = &IMS_PROD_SMSG OR
              &DSN = &IMS_PROD_LMSG)
          DO
            SET &STORCLAS = 'DBCRT'
            EXIT
          END
        WHEN (&DSN = &IMS_PROD_WADS OR
              &DSN = &IMS_PROD_OLDS OR
              &DSN = &IMS_PROD_SPA)
          DO
            SET &STORCLAS = 'FASTWRIT'
            EXIT
          END
        WHEN (&DSN = &IMS_PROD_DL1)
          DO
            SET &STORCLAS = 'FAST'
            EXIT
          END
        OTHERWISE
          DO
            SET &STORCLAS = 'STANDARD'
            EXIT
          END
      END
    END
  END
/*****
/*   End of IMS Select
/*****
```

Figure 81. SELECT Section for IMS from Storage Class ACS Routine

## Designing for DB2 Data

DB2 system and recovery data sets can benefit from dual copy service for the following reasons:

- The directory and DB2 catalog are critical and cannot be duplexed by DB2.
- The boot strap data set (BSDS) and the active logs can be duplexed by DB2. Fault-tolerant devices can provide additional protection from hardware outages.
- If you currently place the BSDS and dual active logs on separate volumes, you can continue in SMS by setting the storage class Guaranteed Space attribute to YES.

**Recommendation:** Do not use the guaranteed space attribute to specify volumes to separate the partitions of a DB2 table space. You can avoid using the guaranteed space attribute in the following situations:

- SMS volume selection algorithms normally spread data set allocations if you have a large number of volumes in a storage group.
- You can use DFSMS striping for linear data sets.



- With the IBM RVA and the ESS, multiple logical volumes can be mapped to a physical volume due to their RAID architecture, volume capacity, and, if applicable, their log structured array architecture.
- The IBM ESS has large cache structures and sophisticated caching algorithms. It is capable of providing a much larger throughput. Its capabilities of parallel access volume and multiple allegiance allow many concurrent accesses to the same data. Therefore, specific volume placement and data set separation used for performance reasons should no longer be required.

Table 20 shows the relationship between the data set, data type, and LLQ that identifies the data set in the storage class ACS routine.

*Table 20. DB2 Data Sets Requiring High Availability*

DB2 data set	Data set type	3rd-level qualifier
DB2 catalog	System	DSNDB06
DB2 directory	System	DSNDB01

The active logs can benefit from DASD fast write and cache services, because DB2 transactions can wait for logging before completing. The access reference pattern for the database can affect the caching or DASD fast write benefits for production databases. The DFSMS I/O statistics provide long term measurement of data set accesses, response components, and cache statistics. They can be used in application tuning or batch window reduction. You can benefit significantly from I/O priority scheduling in a mixed workload environment. For example, to achieve consistent response times for transaction processing, you can prioritize transaction processing reads above query reads and DB2 asynchronous writes.

Table 21 shows the relationship between the data set, data type, and LLQ that identifies the data set in the storage class ACS routine.

*Table 21. DB2 Data Sets Having High Write Activity*

DB2 data set	Data set type	Low-level qualifier
Active Log	Recovery	LOGX/Y
Boot Strap	System	BSDSX/Y

The access reference pattern for the database can affect the caching benefits for production DB2 databases. Table 22 summarizes this result.

*Table 22. DB2 Data Sets Having High Read Activity*

DB2 data set	Data set type	2nd-level qualifier
Production databases	Databases	DSNDBC/D

The sample configuration has the following storage class assignments:

```

Class name
  DB2 data sets
DBCRT
  Directory, Catalog, BSDS
FASTWRIT
  Active logs
FAST DB2 databases

```

All other DB2 data sets are assigned the STANDARD storage class. Refer to Table 23 on page 161 for the list of attributes associated with these storage classes.

## Preventing Inadvertent Migration of DB2 Data

Database data sets may stay open for a number of days. To prevent them from being inadvertently migrated during the DFSMSHsm automatic space management cycle, DFSMS updates the last-referenced date when data sets are closed or when databases are shut down. However, this update and close process is bypassed for DB2 data sets if the system chooses to use the DB2 fast shutdown option. As a result, DB2 data sets can be unintentionally migrated, and if so, it can take a long time for the unintended migrations to be recalled during the database start-ups.

You can prevent inadvertent migration of DB2 data in one of the following ways:

- Assign a management class with Command or Auto Migrate = None.
- Assign a management class with Command or Auto Migrate = Command. This specification prevents DFSMSHsm from automatically migrating DB2 data during the primary space management cycle, but it allows you to migrate test databases on command.
- Assign a management class with Primary Days non Usage > *nm* days, where *nm* is greater than the number of days when these data sets stay open.
- Modify the relevant DB2 database settings. Contact your IBM DB2 Service Support Representative for further information.

## Designing the Storage Class Routine for DB2 Data

DB2 database data sets are identified because of the naming conventions of the HLQs and LLQs. Describe the masks for the data sets requiring above average performance and availability in a FILTLIST statement. Assign them a variable name that is tested in the ACS routine logic. Figure 82 shows the filter list for DB2 storage class ACS routine.

```

FILTLIST section for DB2 database data sets.
FILTLIST SPECIAL_USERS      INCLUDE('SYSPROG','STGADMIN','DBA')
FILTLIST VALID_STORAGE_CLASS INCLUDE('BACKUP','CRITICAL','FAST','FASTREAD',
                                     'FASTWRIT','GSPACE','MEDIUM','NONVIO',
                                     'STANDARD','DBCRT')
/*****
/*          Start of DB2-related FILTLISTS          */
/*****
FILTLIST DB2                INCLUDE(%DB*.**)
FILTLIST DB2_PROD_DIRECTRY  INCLUDE(PDB*.*.DSNDB01.***)
FILTLIST DB2_PROD_CATALOG  INCLUDE(PDB*.*.DSNDB06.***)
FILTLIST DB2_PROD_LOG       INCLUDE(PDB*.*.LOG*)
FILTLIST DB2_PROD_BSDDS     INCLUDE(PDB*.*.BSDDS*)
/*****
/*          End of DB2-related FILTLISTS          */
/*****

```

Figure 82. FILTLIST Section for DB2 from Storage Class ACS Routine

Figure 83 on page 160 shows the common logic that lets privileged users, such as the database administrator, override SMS allocation decisions or even to allocate the data set as a non-system-managed data set.

```

WHEN (&GROUP = &SPECIAL_USERS &&          /* Permit storage admin., */
      &STORCLAS = &VALID_STORAGE_CLASS) /* system programmers, */
DO                                           /* and DBAs to specify */
  SET &STORCLAS = &STORCLAS                /* a storage class      */
EXIT
END
WHEN (&GROUP = &SPECIAL_USERS &&          /* Permit storage admin. */
      &STORCLAS = 'NONSMS')                /* or data base admin. */
DO                                           /* to create             */
  SET &STORCLAS = ''                        /* non-system-managed   */
EXIT                                         /* data sets             */
END

```

*Figure 83. Logic to Permit Special Users to Override SMS allocation*

Figure 85 on page 162 shows the logic to assign storage classes to DB2 database data sets. Only production database data sets are assigned specialized services in this routine. The first character of the data set's HLQ denotes whether the data set is production (P), test (T), or end-user (U). The three low-order characters of the HLQ are set to DB2. The DSN mask defined in the FILTLIST section for the data set type describes it in the LLQ.

Based upon the naming conventions, the following is also true:

- The first WHEN clause verifies that the data set belongs to a DB2 system by checking the FILTLIST variable, DB2
- Each uniquely named data set type is assigned the recommended storage class
- Any DB2 production database data set not specifically identified or any test or end-user DB2 database data set, is assigned STANDARD storage services

## Example: SELECT section for DB2 from storage class ACS routine

```

/*****
/*   Start of DB2 Select
/*****
WHEN (&DSN = &DB2)
DO
  SELECT
    WHEN (&DSN = &DB2_PROD_LOG)
    DO
      SET &STORCLAS = 'FASTWRIT'
      EXIT
    END
    WHEN (&DSN = &DB2_PROD_CATALOG OR
          &DSN = &DB2_PROD_DIRECTRY OR
          &DSN = &DB2_PROD_BSDS)
    DO
      SET &STORCLAS = 'DBCRTIT'
      EXIT
    END
  OTHERWISE
  DO
    SET &STORCLAS = 'STANDARD'
    EXIT
  END
END
END
END

```

Figure 84. SELECT Section for DB2 from Storage Class ACS Routine

Table 23 shows the attributes for the storage classes assigned to database data sets.

Table 23. Storage Classes for Database Data

Attributes	Storage Classes			
Name	MEDIUM	FAST	DBCRTIT	FASTWRIT
Direct Millisecond Response	10	5	10	5
Direct Bias	—	—	W	W
Sequential Millisecond Response	10	5	10	5
Sequential Bias	—	—	—	W
Availability	STANDARD	STANDARD	CONTINUOUS	STANDARD
Accessibility	STANDARD	CONTINUOUS	CONTINUOUS	CONTINUOUS
Guaranteed Space	NO	NO	NO	NO
Guaranteed Synchronous Write	NO	NO	NO	NO

### Example: SELECT section for DB2 from storage class ACS routine:

```
/******  
/* Start of DB2 Select */  
/******  
WHEN (&DSN = &DB2) /* Select DB2 data sets */  
DO  
  SELECT  
    WHEN (&DSN = &DB2_PROD_LOG) /* Use fast write for */  
    DO /* active logs */  
      SET &STORCLAS = 'FASTWRIT'  
      EXIT  
    END  
    WHEN (&DSN = &DB2_PROD_CATALOG OR /* Dual copy for catalog */  
          &DSN = &DB2_PROD_DIRECTRY OR /* directory and boot */  
          &DSN = &DB2_PROD_BSDS) /* strap data set */  
    DO  
      SET &STORCLAS = 'DBCRT'  
      EXIT  
    END  
  OTHERWISE /* Give all other DB2 data*/  
  DO /* average performance */  
    SET &STORCLAS = 'STANDARD'  
    EXIT  
  END  
END  
END
```

Figure 85. SELECT Section for DB2 from Storage Class ACS Routine

### Designing for Extended Addressability

Data warehousing projects that require table spaces larger than one terabyte can use extended addressability for linear data sets. To do this, assign a data class with the extended addressability attribute to the data set when it is defined, and ensure that the following attributes are specified for the data class:

```
Recorg = LS  
Data Set Name Type = Extended  
IF Extended = Required  
Extended Addressability = Yes
```

Then, ensure that your data class ACS routine for DB2 includes the following statement:

```
FILTLST DB2WH INCLUDE(DWH*.**) /* assuming these have a high level*/  
/* qualifier of DWH...*/  
SELECT  
  WHEN (&DSN = &DB2WH)  
  DO  
    &DATACLAS = 'LDSEA'  
  END  
  OTHERWISE  
  ...
```

### Designing for Database Data Backup and Recovery

**Recommendation:** Use your database management system's utilities to perform backup and recovery, because most IMS and DB2 production database data has specialized backup and recovery requirements. Or, consider using DFSMSdss or DFSMSshm's automatic backup services, supported by concurrent copy and virtual concurrent copy support, to help you do point-in-time backups. IMS and DB2

utilities support and invoke DFSMSdss for point-in-time copy support. DFSMSShsm and DFSMSdss can be used for all CICS VSAM File Control data sets.

You should not use DFSMSShsm to manage most production database data. Instead, assign the NOACT management class to these data sets. NOACT inhibits DFSMSShsm space and availability management. Specifically, Auto Backup is set to NO so that DFSMSShsm does not back up the data set, Admin or User Command Backup is set to NONE to prohibit manual backup commands, and expiration attributes are set to NOLIMIT to prevent data set deletion.

Although production database data does receive automatic backup service, you can use DFSMSdss to run point-in-time for production database data sets. Accessibility is set to CONTINUOUS for all storage classes assigned to production database data sets to ensure that the data set is allocated to a point-in-time capable volume.

Database data that has less critical availability requirements, typically test or end-user databases, benefit from system management using DFSMSShsm. Additionally, selected data types for production systems can be effectively managed using SMS facilities.

For CICS/VSAM systems, extrapartition transient data, test and end-user database data can be managed with DFSMSShsm. Extrapartition transient data is directed to DFSMSShsm's migration level 2 by assigning the DBML2 management class to these data types. The attributes for DBML2 keep data sets on primary storage for two days (Migrate Primary Days Non-usage=2) and, if not used, they are migrated to tape (Level 1 Days Non-usage=0).

End-user and test data sets are assigned the DBSTAN management class. This class is different from the STANDARD management class because backup copies for data sets assigned to it are retained much longer than average data sets (Retain Days Only Backup Version=400).

DFSMSShsm uses high-speed management service using concurrent copy for these data sets, because Backup Copy Technique is set to CONCURRENT REQUIRED.

After doing a data set restore using DFSMSdss (either directly or driven by DFSMSShsm), the databases need to be brought to the point of failure or to a point of consistency using forward recovery logs. This can be achieved using CICSVR for CICS VSAM File control data sets. CICS and CICSVR support backup-while-open using either concurrent copy or virtual concurrent copy support.

For IMS systems, DFSMSShsm can manage change accumulation logs and image copies. These data sets can stay on primary storage for a short time and then migrate directly to tape. The DBML2 management class is assigned.

DFSMSShsm uses high-performance backup service using concurrent copy for these data sets, because Backup Copy Technique is set to CONCURRENT REQUIRED.

For DB2 systems, you can manage archive logs and image copies with DFSMSShsm. These data sets can be retained on primary storage for a short period of time and then migrated directly to tape. DBML2 management class is provided for these data set types. End-user database data can also be managed. These data sets are assigned the DBSTAN management class.

DFSMSHsm uses high-performance management service using concurrent copy for these data sets, because Backup Copy Technique is set to CONCURRENT REQUIRED.

Table 24 shows the attributes for the management classes assigned to database data sets.

*Table 24. Management Classes for Database Data*

Attributes	Management Classes		
Name	DBML2	DBSTAN	NOACT
Expire after Days Non-usage	NOLIMIT	NOLIMIT	NOLIMIT
Expire after Date/Days	NOLIMIT	NOLIMIT	NOLIMIT
Retention Limit	NOLIMIT	NOLIMIT	NOLIMIT
Partial Release	COND IMMED	NO	NO
Migrate Primary Days Non-usage	2	15	—
Level 1 Days Non-usage	0	60	—
Command or Auto Migrate	BOTH	BOTH	NONE
# GDG Elements on Primary	1	1	—
Rolled-off GDS Action	EXPIRE	EXPIRE	—
Backup Frequency	1	0	—
Number Backup Versions, Data Set Exists	2	3	—
Number Backup Versions, Data Set Deleted	1	1	—
Retain Days only Backup Version	60	400	—
Retain Days Extra Backup Versions	30	100	—
Admin or User Command Backup	BOTH	BOTH	NONE
Auto Backup	YES	YES	NO
Backup Copy Technique	CONCURRENT REQUIRED	CONCURRENT REQUIRED	STANDARD

## Designing the Management Class ACS Routine

The management class ACS routine handles database data. The data set masks for the data set types managed by DFSMSHsm are identified, and placed either in the FILTLIST DBML2 or DBSTAN, depending on the management requirements of the data type. Figure 86 on page 165 shows the FILTLIST statements required for the sample management class routine.

```

FILTLIST DB2          INCLUDE(PDB*.***)
FILTLIST IMS         INCLUDE(PIMS*.***)
FILTLIST CICS        INCLUDE(PCICS*.***)
FILTLIST DBSTAN      INCLUDE(UDB*.***,UCICS*.***)
FILTLIST DBML2       INCLUDE(*.ICOPY*.***,*.ARCHLOG*.***,
                          ***.CHGACCUM,***.IMACOPY,
                          ***.DFHEXTRA)
FILTLIST SPECIAL_USERS  INCLUDE('SYSPROG','STGADMIN','DBA')
FILTLIST VALID_STORAGE_CLASS INCLUDE('BACKUP','CRITICAL','FAST','DBCRIT',
                          'FASTREAD','FASTWRT','GSPACE',
                          'MEDIUM','NONVIO','STANDARD')
FILTLIST VALID_MGMT_CLASS  INCLUDE('DBML2','DBSTAN','EXTBAK',
                          'GDGBKUP','GDGPROD','INTERIM',
                          'NOACT','STANDARD')

```

*Figure 86. FILTLIST Section for Database from Management Class ACS Routine*

In the database logic section of the management class routine, the data set name is matched with the two FILTLIST variables and, if there is a match, the corresponding management class is assigned. In this routine, any production database data sets not specifically identified as managed data types are assigned the NOACT class. Figure 87 on page 166 shows the ACS coding segment for database data in the management class ACS routine.



## Example: Management class ACS routine sections for database data

```

/*****
/*      Start of Mainline SELECT
/*****
SELECT
  WHEN (&GROUP = &SPECIAL_USERS &&
        &MGMTCLAS = &VALID_MGMT_CLASS) /* Let system programmers */
    DO /* assign externally- */
      SET &MGMTCLAS = &MGMTCLAS /* specified management */
      EXIT /* class */
    END
  WHEN (&DSN = &DBML2) /* Send CICS extra- */
    DO /* partition, DB2 image */
      SET &MGMTCLAS = 'DBML2' /* copies and archive logs,*/
      EXIT /* IMS change accumulation */
    END /* and image copies to ML2 */

  WHEN (&DSN = &DBSTAN) /* Send CICS and DB2 */
    DO /* end-user database data */
      SET &MGMTCLAS = 'DBSTAN' /* to ML1 */
      EXIT
    END

  WHEN (&DSN = &CICS OR
        &DSN = &IMS OR
        &DSN = &DB2) /* Don't manage any other */
    DO /* production database data*/
      SET &MGMTCLAS = 'NOACT'
      EXIT
    END
  OTHERWISE
    DO
      SET &MGMTCLAS = 'STANDARD' /* Give normal datasets */
      EXIT /* medium migration and */
    END /* backup service */

END
/*****
/*      End of Mainline SELECT
/*****
END /* End of Management Class Procedure */
```

Figure 87. Management Class ACS Routine Sections for Database Data

## Relating DB2 STOGROUPs to SMS Storage Groups

There is a unique storage group for production databases and selected system data sets, but end-user, recovery database data, and most system data sets are directed to the primary storage groups, PRIME80 or PRIME90. If you are migrating DB2 database data to system management, and you use DB2 STOGROUPs to manage DB2 database data, design your SMS storage groups to be compatible with your database administrator's DB2 STOGROUP definitions.

**Recommendation:** After you have successfully migrated some database data to system management, use SMS, rather than DB2, to allocate database data.

DB2 lets the database administrator define a collection of volumes that DB2 uses to find space for new data set allocations. This collection is known as a DB2 STOGROUP. If you use DB2 STOGROUPs to manage DB2 allocation, you must ensure that your strategy for DB2 database data migration does not conflict with

your database administrator's allocation procedures. To have SMS select the volumes for database allocation without assistance from DB2, define DB2 STOGROUPs with VOLUMES(\*). Do *not* use specific volume serial numbers and storage class with guaranteed space for these allocations.

Table 25 shows the attributes for the storage groups defined for CICS/VSAM, IMS, and DB2 database data. No space management or backup services are performed by DFSMSHsm for these storage groups. However, volumes in the database storage groups are dumped by DFSMSHsm for local recovery required because of a hardware failure. These volumes are also dumped and stored offsite in preparation for a site disaster.

*Table 25. Storage Groups for Database Data*

Attributes	Storage Groups		
	CICS	DB2	IMS
Name	CICS	DB2	IMS
Type	POOL	POOL	POOL
Auto Migrate	NO	NO	NO
Auto Backup	NO	NO	NO
Auto Dump	YES	YES	YES
Dump Class	DBONSITE, DBOFFS	DBONSITE, DBOFFS	DBONSITE, DBOFFS
High Threshold	75	75	75
Low Threshold	60	60	60
Guaranteed Backup Frequency	NOLIMIT	NOLIMIT	NOLIMIT

Figure 88 on page 168 shows the filters that identify the production database data sets using the data set name's HLQ and isolate them in their own storage group. You or the database administrator allocate the production database data using a storage class to specify the data's performance and availability requirements. The storage class also indicates that the database is being allocated by an administrator authorized to place data in this storage group. Figure 89 on page 168 shows the ACS routine statements that identify the production database and database storage class, and assign the storage group.

Only selected system, recovery, and production database data is selected in the storage group ACS routine to be allocated in the database storage groups. All other database data is allocated on volumes in the PRIME<sub>xx</sub> or LARGE<sub>xx</sub> storage group.

## Example: FILTLIST section for database from storage group ACS routine

```
/* Start of FILTLIST Statements */
/* *****/
FILTLIST CICS          INCLUDE(PCICS*.**)
                      EXCLUDE(**.DFHXRMSG,**.DFHXRCTL,**.LOADLIB,
                                **.COB2CICS,**.COB2LIB,**.PLILINK,
                                **.DFHJ01%,**.DFHRSD,**.DFHCSD,
                                **.DFHINTRA,**.DFHTEMP)
FILTLIST DB2          INCLUDE(PDB*.**)
                      EXCLUDE(*.*.DSNDB01.**,**.LOG*)
FILTLIST IMS          INCLUDE(PIMS*.**)
                      EXCLUDE(**.LGMSG,**.OL*,**.QBLKS,
                                **.RDS,**.SHMSG,
                                **.SPA,**.WADS*,**.RECON)
FILTLIST SPECIAL_USERS INCLUDE('SYSPROG','STGADMIN','DBA')
/* *****/
/* End of FILTLIST Statements */
/* *****/
```

Figure 88. FILTLIST Section for Database from Storage Group ACS Routine

## Example: SELECT section for database from storage group ACS routine

```
WHEN (&DSN = &CICS AND          /* Isolate CICS databases */
      &GROUP = &SPECIAL_USERS)
DO
  SET &STORGRP = 'CICS'
  EXIT
END
WHEN (&DSN = &DB2 AND          /* Isolate DB2 databases */
      &GROUP = &SPECIAL_USERS)
DO
  SET &STORGRP = 'DB2'
  EXIT
END
WHEN (&DSN = &IMS AND          /* Isolate IMS databases */
      &GROUP = &SPECIAL_USERS)
DO
  SET &STORGRP = 'IMS'
  EXIT
END
OTHERWISE                       /* Normal and medium-sized */
DO                               /* data sets go to the */
  SET &STORGRP = 'PRIME80','PRIME90' /* PRIME storage group */
  EXIT
END
```

Figure 89. SELECT Section for Database from Storage Group ACS Routine

---

## Allocating DB2 Partitioned Table Spaces

DB2 does software striping on partitioned table spaces, where each of the partitions is a separate linear data set. Striping is used so that parallel I/O can be performed concurrently against more than one of these partitions. The benefit of striping is only achievable if multiple partitions do not get allocated on the same volume. You can achieve volume separation without resorting to the storage class guaranteed space allocations on SMS volumes.

To do this, allocate all of the partitions in a single IEFBR14 job step, using JCL. As long as there are adequate number of volumes in the storage groups and the volumes are not above allocation threshold, the SMS allocation algorithms with SRM ensure each partitions is allocated on a separate volume.

Do not use guaranteed space with volume serial numbers, as it is time-consuming and might result in problems during extensions to new volumes if the specified volume does not have enough space. Instead, use '\*' as the volume serial number in JCL, or specify a unit or volume count of more than 1 to ensure that data sets can be extended to new volumes. If you run out of space on the specified number of volumes, use an AMS ALTER ADDVOL command to add additional volumes and reaccess the table.



---

## Chapter 10. Planning for System-Managed Tape

Converting to system-managed tape can help you optimize your installation's tape subsystem operation and media utilization in the following ways:

- Allowing the system to manage the placement of tape data
- Automating tape mount and demount operations
- Using the full capacity of the tape media
- Taking advantage of hardware and software compaction
- Exploiting new technologies for removable media

System-managed tape allows you to define requirements for your tape data in logical, rather than physical, terms. Without requiring changes to programs or JCL, you can define the policies that the system uses to map those logical requirements to physical tape resources, which can include mixed device types (for example, IBM 3490 or 3490E transports), mixed media (for example, cartridge system tapes or enhanced capacity cartridge system tapes), and tape library dataservers.

This chapter describes the tasks you need to complete as you plan for system-managed tape. It shows you how to optimize your current tape environment and convert your tape volumes to system management. See Chapter 11, "Optimizing Tape Usage," on page 177 and Chapter 12, "Managing Tape Volumes," on page 219 for more information on implementation considerations and sample plans for managing tape data and volumes under SMS.

---

### Optimizing Your Current Tape Environment

Depending on your installation's current tape practices, you could maximize the use of your tape resources and even improve batch job throughput by implementing tape mount management (TMM).

**Tip:** With TMM, you need to extensively analyze tape mounts, modify ACS routines to redirect allocations intended for tape to a DASD pool, and then migrate them to tape with the DFSMSHsm interval migration. To avoid this complicated process, you can use the IBM Virtual Tape Server (VTS) to fill tape media, reduce tape mounts, and save system resources. See "Using the Virtual Tape Server (VTS) to Optimize Tape Media" on page 179 for more information.

Tape mount management is a methodology for managing tape data sets within the DFSMS storage hierarchy:

1. Tape data sets are categorized according to size, pattern of usage, and other criteria, so that appropriate DFSMS policies can be assigned to tape mount management candidates.
2. Data sets written to tape are intercepted at allocation and, if eligible for tape mount management, redirected to a system-managed DASD buffer. The buffer serves as a staging area for these data sets until they are written to tape. The location of the data is transparent to the application program.
3. DFSMSHsm periodically checks the occupancy of the DASD buffer storage group to ensure that space is available when needed and migrates data sets to a lower level of the storage hierarchy when they are no longer required on primary DASD volumes.
4. DFSMSHsm eventually moves the data to tape, using single-file format and data compaction to create full tape cartridges.

5. If an application later requests a data set, DFSMSHsm automatically recalls it from where it resides in the storage hierarchy and allocates it on primary DASD for access.

This process can significantly reduce tape mounts and the number of cartridges required to store the data. Operations can benefit from a decreased number of random tape mounts, while applications benefit from improved job throughput because the jobs are no longer queued up on tape drives.

**Restriction:** You cannot use tape mount management for OAM objects that are written to tape.

Converting tape data sets to system-managed storage requires careful planning. For tape mount management, you should include the following major tasks in your implementation plan:

1. Analyze the current tape environment.
2. Simulate the proposed tape mount management environment.
3. Implement advanced cartridge hardware.
4. Select DASD volumes to satisfy buffer requirements.
5. Define SMS classes and storage groups for candidate data sets.
6. Create the ACS routines, or add to the current routines.
7. Tune DFSMSHsm operations.

Although JCL changes are usually not necessary for implementing tape mount management, you do need to determine the effects of jobs that leave tape data sets uncataloged, and special expiration date codes that are used by some tape management systems, if these practices exist in your current environment. See Chapter 11, "Optimizing Tape Usage," on page 177 and Chapter 12, "Managing Tape Volumes," on page 219 for more information on planning considerations.

## Analyzing Tape Usage with the Volume Mount Analyzer

The volume mount analyzer is a tool that can help you analyze your installation's current tape environment. It produces reports that profile your tape mount workload and tape media usage, so that you can decide whether tape mount management might benefit your installation. In addition, the volume mount analyzer identifies data sets that, because of their large size, can benefit from advanced cartridge hardware and media technology.

The volume mount analyzer uses your installation's SMF data to analyze tape mount activity and to produce reports that help you perform the following tasks:

- Identify trends and other information about tape mount events, including data set name, job, program, and data set size (bytes transferred).
- Evaluate the tape hardware configuration.
- Quantify<sup>®</sup> the benefits of tape mount management in terms of library and tape mount reduction.
- Determine which data sets are good candidates for tape mount management.
- Determine data class and management class requirements for tape mount management candidates
- Develop ACS routine filters to select tape mount management candidates and exclude other data sets that must remain on tape.
- Determine the size of the DASD buffer and the high and low thresholds needed for the buffer's storage group.

To run the volume mount analyzer, you must have DFSORT installed.

The process of conducting a volume mount analyzer study consists of the following six major steps:

1. Determining a representative time period for the study; usually a one-month cycle that includes peak loads as well as routine processing.
2. Collecting the required SMF record types during the study period to record tape information used by the volume mount analyzer.
3. Running the volume mount analyzer's SMF data extractor program, GFTAXTR, to reduce the amount of input data.
4. Running the volume mount analyzer program, GFTAVMA, to generate the summary and detailed reports.
5. Analyzing the results. At this point you might want to modify parameters for GFTAVMA, and rerun the program until you are satisfied with the results.
6. Determining whether you want to implement tape mount management.

**Related Reading:**

- For procedures on performing a volume mount analyzer study and interpreting the results, see *z/OS DFSMS Using the Volume Mount Analyzer*.
- For information on implementing tape mount management using the volume mount analyzer reports, see Chapter 11, "Optimizing Tape Usage," on page 177, Chapter 12, "Managing Tape Volumes," on page 219, and *MVS/ESA SML: Managing Data*.

---

## Converting Tape Volumes to System Management

Installing tape libraries and converting your volumes to system management offers the following benefits:

- With automated tape library dataservers, tape mounts and demounts are automated. DFSMS and the dataserwer work together to automate the retrieval, storage, allocation, and control of standard and enhanced capacity cartridge system tape volumes.
- With the manual tape library, allocation and control of standard and enhanced capacity cartridge system tape volumes are accomplished.
- With automated tape libraries, tape subsystem performance is improved through the enhanced device selection process used for system-managed tape.
- Compaction and media interchange attributes are more easily specified for system-managed tape volumes.
- Disaster recovery procedures are simplified. Data can be written to tape volumes at an offsite tape library during backup processing and, in the event of a local disaster, rebuilt from those offsite volumes.

With the Automated Tape Library Dataserver, when you allocate a new data set in the automated environment, the system selects an appropriate tape cartridge from a scratch tape pool in the Automated Tape Library Dataserver. If you request a data set that is already stored on a cartridge in a tape library dataserwer, DFSMS enables the dataserwer to automatically locate, mount, demount, and store the correct tape cartridge for you. With the manual tape library, scratch pool support can be provided by your tape management system.

DFSMSrmm also plays an important role in this environment. It enables the tape library dataserwer to automatically recycle tapes back into the scratch pool when the data sets on the tapes expire. This eliminates the need for you to manually change these tapes back into scratch tapes. DFSMSrmm invokes the object access method (OAM) to update the volume's status during the recycle process. It also



causes OAM to eject cartridges from the tape library dataserer to move volumes offsite in preparation for disaster recovery.

## Preparing for Tape Volume Conversion

To prepare your installation for system-managed tape, you need to include the following major tasks in a conversion plan:

1. Evaluate job dependencies on the JCL UNIT parameter.

Consider the implications of certain SMS restrictions on JCL UNIT parameter usage. The device used to satisfy an allocation request is selected from device pool information associated with a given tape library. Because of this, the following situations occur:

- Demand allocation is not supported. For example, UNIT=520 (where 520 is the address of a 3490 drive) is not valid if SSMHONOR is not coded.
- The UNIT keyword is not actually used to select a tape device. However, this keyword is available to your ACS routines to use for filtering purposes.
- Devices requested using unit affinity, such as UNIT=AFF=DD1, are honored only if the volumes reside in the same tape library and use compatible devices.

Except for demand allocation, JCL changes are not required to use tape library dataservers.

2. Define the tape environment to z/OS.

Tape drives in an automated or manual tape library are defined using the hardware configuration definition. A library ID defined to the ISMF library application links the system-managed tape library definition to the tape library. The library ID is defined to HCD by specifying the LIBRARY-ID and LIBPORT-ID parameters for each library device. Both the LIBRARY-ID and the LIBPORT-ID are arbitrary numbers. The HCD help text for the LIBRARY-ID and LIBPORT-ID parameters explain how you can obtain the IDs.

3. Define OAM.

Tape library support uses OAM to define system-managed tape libraries and volumes. You need to update various PARMLIB members to define OAM.

4. Define the classes, storage groups, and associated ACS routines.

You can use the established data classes when you implemented tape mount management to control the categories of tape data sets that are allocated on system-managed tape volumes. You can also define special classes for managing DFSMSHsm-owned volumes in a tape library, and for tapes that hold objects.

If your tape management system supports the pre-ACS interface, you can use the information available to you (for example, the scratch pool and policy in the MSPPOOL and MSPOLICY variables) when coding the ACS routines to direct tape allocations to specific libraries, to a DASD pool (for tape mount management), or to keep them outside of the system-managed environment.

Tape storage groups, defined under ISMF, associate the tape libraries to tape storage groups. A scratch tape volume becomes system-managed when it is entered into a system-managed library. A scratch tape becomes part of a storage group when a system-managed data set or object is allocated on the volume. System-managed volumes are assigned to a tape storage group. Tapes that contain objects also belong to either an object or an object backup storage group.

After you define a tape storage group, you must set the status for the storage group on each system that uses the tape library dataserer.

5. Create the tape configuration database.

You must define one general volume catalog. One or more specific volume catalogs can also be defined based on your installation's requirements. The collection of your installation's general and specific volume catalogs is the tape configuration database.

DFSMS determines which catalog to update based on the first character of the volume serial number. Naming conventions for volume serial numbers can help you balance catalog update activities.

You can use access method services to define volume catalogs and use standard ICF support for backup and recovery.
6. Define the tape libraries.

Before you define your tape libraries, make sure that update authority for your tape configuration database is restricted to storage administrators.

Create a logical tape library definition for each grouping of tape volumes associated with a collection of tape devices. This definition becomes part of your active SMS configuration and a library entry is generated in the tape configuration database. It is created using a new ISMF application that is invoked from the Library Management application. ISMF also allows you to redefine the tape library from information in the tape configuration database.
7. Create any system-managed tape exits, if required.

There are several DFSMSdftp installation exits that you can use specifically with tape library dataservers. DFSMSrmm also uses selected DFSMSdftp exits to manage some of its tape processing.
8. Translate and validate the new SMS configuration.

Use the same translation and validation steps that you follow for most other SMS configurations. The new configuration is considered valid if all tape libraries associated with tape storage groups exist for SMS.
9. Test the new SMS configuration.

Use ISMF to write and run test cases to verify that your new, or modified, ACS routines properly assign the new tape classes and storage groups.
10. Activate the new SMS configuration.

Activate the new system-managed tape configuration as you would other SMS configurations.
11. Start OAM.

Place the start-up PROC for OAM in PROCLIB. You must have OAM running on all processors that will use the automated libraries.
12. Enter volumes in the tape library dataserver.

If you are using the automated tape library dataserver, add your tape cartridges to the dataserver before you begin testing allocations.
13. If you are using a manual tape library, use the Manual Cartridge Entry programming interface or the library enter console command.
14. Test library usage for SMS tape allocations.

Perform selective allocations to check library usage before converting the rest of your tape data to system-management.
15. Put the tape library into production.

Consider converting your tape data to system management by category of data, using the following suggested order:

  - a. Large temporary data sets
  - b. DFSMSHsm-owned volumes
  - c. Offsite volumes

- d. Active volumes
- e. Backup volumes

Normal SMS processing ignores the UNIT parameter. So, JCL or dynamic allocations could be specifying unit names that no longer exist in the system. However, if the new keyword SMSHONOR (in JCL) or DALSM SHR (for dynamic allocations) is coded along with a valid device name or esoteric name, device allocation will attempt to allocate to the devices that are common to the UNIT and device pools selected by SMS.

**Related Reading:** You may need more information during the various stages of the installation and conversion process. See Chapter 11, “Optimizing Tape Usage,” on page 177, Chapter 12, “Managing Tape Volumes,” on page 219, and the following publications:

- For information about JES3 support for tape library dataservers, see *z/OS JES3 Initialization and Tuning Guide*.
- For more information about using OAM to define system-managed tape libraries and using the Manual Cartridge Entry programming interface, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.
- For information on how certain DFSMSdfp installation exits affect library management, see *z/OS DFSMS Installation Exits*.
- For information about DFSMSrmm support for manual cartridge entry processing and for the pre-ACS interface, see *z/OS DFSMSrmm Implementation and Customization Guide*.

---

## Chapter 11. Optimizing Tape Usage

In an average installation, 50% of tape data sets have inactive data—data written once and never read again. Most inactive data sets are *point-in-time* backups. These are application-initiated data set backups used only if an application or system failure occurs. Although data set backup is critical, application-initiated backups to tape make poor use of the tape media and subsystem, and require many costly operator tape mounts. The remaining tape data sets, *active* data, cause the same inefficiencies in tape cartridge use. However, they are usually not very active; 60% - 90% of all tape data sets have all accesses on the same calendar date.

You can use system-managed storage facilities to optimize management of both inactive and active tape data sets. *Tape mount management* helps you understand your tape workload and use advanced tape hardware and DFSMS facilities to accomplish the following tasks:

- Reduce tape mounts

Tape mounts for data sets that are good tape mount management candidates are reduced because these data sets are written to a system-managed DASD buffer. They are subsequently written to tape by DFSMSHsm, along with other data sets, and are automatically retrieved by DFSMSHsm if accessed.

- Reduce tape library inventory and maximize media usage

Tape mount management candidates are written by DFSMSHsm to tape in single-file, compacted form. DFSMSHsm attempts to fill the entire tape volume before using another volume.

- Improve turnaround time for batch jobs depending on tape data sets

Most tape processing jobs are queued on tape drives, not mount time or tape I/O. This is because, when the drive is finally allocated, very little data is written to the cartridge. Batch jobs using data sets that are written to the system-managed DASD buffer do not wait for tape mounts, and can perform I/O at DASD or cache speed.

You can implement these tape mount management techniques without changing your JCL streams or backup and recovery procedures. Tape data sets that cannot be addressed by tape mount management techniques can also be system-managed. Refer to Chapter 12, “Managing Tape Volumes,” on page 219 for information on these data sets.

You can improve tape management efficiency by implementing tape mount management. Tape mount management is a methodology in which you use your ACS routines to re-route tape data set allocations to DASD. Once the data sets are on DASD, you can use DFSMSHsm to migrate the data sets to tape as a group, reducing the number of tape mounts required. See Figure 90 on page 178 for an illustration of how the tape mount management methodology accomplishes this.

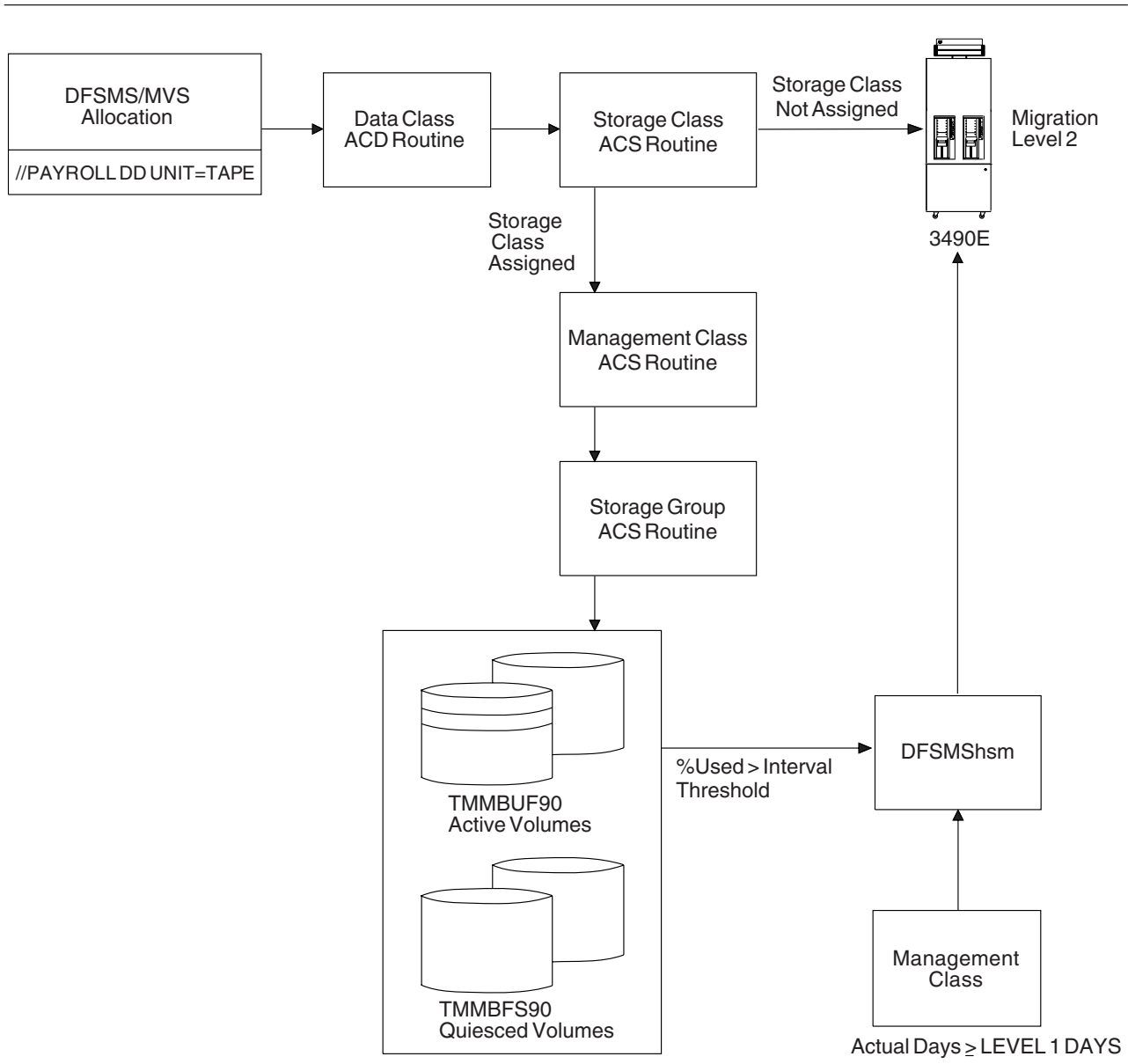


Figure 90. Redirecting Tape Allocations to DASD Buffer Managed by DFSMS

The DASD buffer is a staging area for these tape mount management candidates before they are written to tape. DFSMShsm periodically checks the occupancy of the DASD buffer's storage group. If the allocated space exceeds the midway point between low and high threshold (if you specified interval migration) for the storage group, DFSMShsm moves data sets to its migration level 1 or 2 volumes to bring the buffer down to the low threshold.

Data set movement through the storage hierarchy is based on the management class that you assign to the data set. DFSMShsm uses single-file format and data compaction technologies to create a full cartridge for each migration level 2 tape volume before requesting another.

Very large and offsite tape data sets that must remain in their current form are written directly to tape. You can also write them in a compacted form to make better use of both the media and cartridge subsystem.

DFSMSHsm automates the management of these data sets by checking the DASD buffer to ensure that space is available when needed, and that data sets are migrated to tape when no longer required on DASD storage. Your tape library inventory is reduced because DFSMSHsm tries to fill tapes by stacking data sets in single-file format, in contrast to applications that create separate tape volumes that might only partially fill a tape volume.

Data sets are recalled by DFSMSHsm automatically, if accessed by the application. Batch turnaround time is improved by several hours, because jobs are no longer artificially queued up on tape drive availability.

The following required tasks implement tape mount management:

- Analyzing the current tape environment
- Simulating the proposed tape mount management environment
- Implementing advanced cartridge hardware
- Defining DASD volumes to satisfy buffer requirements
- Defining SMS classes and groups
- Creating ACS routines
- Using tape mount management techniques
- Tuning DFSMSHsm operation

---

## Using the Virtual Tape Server (VTS) to Optimize Tape Media

Another method for optimizing your tape media is through the Virtual Tape Server (VTS) tape library hardware. You can use VTS with or without tape mount management. It does not require ACS routine or other software changes.

VTS lets you define up to 32 virtual tape drives to the host. Not visible to the host are up to 6 physical tape devices. When the host writes to one of the virtual devices, it actually writes to a virtual volume residing on the VTS DASD buffer. The VTS, transparent to the host, copies the entire virtual volume onto a logical volume that is then mapped to physical stacked volumes known only to the VTS.

These logical and physical volumes cannot be ejected directly. However, VTS offers many other advantages. For example, VTS:

- Does not need a DASD buffer
- Does not use DFSMSHsm facilities to fully stack tapes
- Does not require double data movement in the host
- Does not require changes to ACS routines
- Fully uses tape media due to the structure of the virtual volumes on the physical tape

VTS avoids the extensive analysis required to use tape mount management. You can, however, use VMA studies to use VTS more effectively, since these studies identify useful information, such as data sets needing to be stored offsite, or temporary data sets that can be written to DASD and expired.

---

## Migrating Tape Data Sets to System-Managed Storage

Migrating tape data sets to system-managed storage requires careful planning and implementation, such as the following:

- Migrating jobs that reference undefined generics and esoterics
- Migrating jobs that use uncataloged tape data sets
- Analyzing “expiration date codes” from tape management systems
- Use data set level serialization

## Changing Jobs that Reference Undefined Generics and Esoterics

To ensure correct processing of all tape mount management-eligible data sets, the system must determine the original target device class for the data set. It examines the UNIT parameter provided or defaulted on the allocation, and checks that the data set should be allocated on a tape volume.

Change the JCL or dynamic allocation that references an undefined unit name where the data set is eligible for tape mount management, to specify a unit name that exists in the configuration and contains tape devices. If you do not, the system might replace a valid data set with an empty data set, causing data loss.

## Changing Jobs that Use Uncataloged Tape Data Sets

Some data created on tape, typically backup copies of data sets, might be uncataloged. Although an application might create a data set with the same name each time it runs, these data sets must be cataloged if they are system-managed. The following three alternatives handle applications that depend on reusing data sets:

- Consider converting these data sets to generation data sets.
- Catalog the data sets, and design a management class that backs them up with DFSMSHsm and deletes the data set prior to the next application run.
- Eliminate the application-initiated backup process by assigning a management class that backs up the data at regular intervals.

## Analyzing Expiration Date Codes

Large tape installations generally have a tape management system. Most tape data sets have some form of expiration, either a real expiration or a tape management system code. Some of these codes might require special analysis. The following is a summary of common tape management system codes:

- 99000** Catalog control: Tape data sets with this code expire by normal deletion or GDG roll-off. Most data sets have this code.
- 99nnn** Cycle control: “nnn” versions of tape GDSs are kept, regardless of the number of generations specified in the GDG base. If “nnn” is higher than the GDG limit value, the remainder are kept as uncataloged data sets. Then, either the GDG limit would need to be altered, or the management class for these tape data sets could be set to keep rolled-off GDS versions.
- 98000** No control: Tape data sets with this code are not under control of the tape management system. These tapes are usually sent offsite for disaster recovery, vital records, or customer interchange.
- 98xxx** Frequency control: Tape data sets with this code can expire when they are not referenced in “xxx” days. These can be directly assigned to an equivalent management class. If the tape data sets are also GDSs, then rolled-off versions need to be retained until expiration.

You might want to allow users to use JCL to delete a tape data set that has been directed to system-managed DASD. You can use the OVRD\_EXPDT keyword in the IGDSMSxx member of the parmlib, which specifies whether the expiration date should always, or never, be overridden when deleting system-managed DASD data sets. You should use the OVRD\_EXPDT keyword only in the following situations:

- When management class cannot be used
- For use with tape allocations redirected to DASD



- If you never use expiration dates for DASD data sets

## Using Data Set Level Serialization

Ensure that you are using global resource serialization (GRS), or an equivalent product, to convert hardware RESERVE/RELEASE requests to data set ENQUEUEs. Your DASD buffer requirements might increase significantly if you do not use GRS or an equivalent product. Use the interval migration function of DFSMSHsm to manage the DASD buffer. In environments that have GRS active, interval migration or primary space management immediately moves all data sets with a *direct-to-migration-level-2* management class to tape. However, if you have multiple systems, if GRS is not active, these data sets must be at least two days old before interval migration or primary space management migrates the data sets to tape.

For a single processor with GRS not active, data sets that should be written directly to migration level 2 must be at least one day old before being migrated by DFSMSHsm.

When you use GRS to control data set integrity, specify `USERDATASETSERIALIZATION` in the DFSMSHsm PARMLIB.

## Classifying Your Tape Data Sets

Determine eligible data sets for tape mount management before you direct data sets to system-managed storage. Data sets with block sizes larger than 32 KB should not be redirected to DASD because they are not supported. Certain other data sets must stay on tape in their current form and are not eligible to be directed to a DASD buffer. They include offsite, DFSMSHsm-owned, and very large data sets. They also include those data sets with the same data set name as data sets that are already cataloged in the standard catalog search order, in those situations where the duplicate naming cannot be resolved.

Offsite data sets include interchange, disaster recovery and vital records. Disaster and vital records data is shipped to a remote retention facility. Interchange data sets are sent to another data center for processing. Interchange can also be customer input/output data.

DFSMSHsm-owned data is compacted and stored in single-file format on tape cartridges. It includes backup, migration, autodumps, and data sets generated by DFSMSHsm, such as backups of the control data sets.

Very large data sets are multivolume data sets that are not practical to intercept with SMS and manage on DASD, even on a temporary basis. This data includes volume dumps, very large image copies, and other large data sets. The definition of large varies. We use 600 MB as the large data set size for the SMS configuration described in this book. Volume dump data sets can range from 500 MB to 1200 MB, but most of them are greater than 600 MB. You can intercept 600 MB data sets with tape mount management without increasing the DASD buffer. But, if most of them are greater than 600 MB, they should be considered as large and sent directly to tape. Table 26 identifies some typical classes of data sets that must remain on tape in their current form.

Table 26. Data Sets That Must Remain on Tape

Data Type	Description
Interchange	Data that is supplied to another site



Table 26. Data Sets That Must Remain on Tape (continued)

Data Type	Description
Disaster Recovery	Data that is needed for recovery in case of a disaster
Vital Records	Data retained for an extended period for legal or business reasons
Backup	DFSMSHsm backup tape volumes
Migration	DFSMSHsm migration level 2 tape volumes
Volume Dump	Dumps of DASD volumes, created by DFSMSdss or DFSMSHsm, used for a full volume recovery
Very Large Image Copy	Database backups
Large Active Data Set	Application master files

Identify these data sets to your storage class ACS routine so that the storage class read/write variable is always set to null.

Data sets that are not truly large are considered normal-sized. These data sets are eligible to be intercepted and system-managed. They include temporary, active, and backup data sets. Table 27 describes the tape data sets that are candidates for re-direction to DASD.

Table 27. Data Sets That Can Be Redirected to DASD

Data Type	Description
Temporary	Data created and deleted in a single job
Active	Data that is read one or more times
Point-in-time Backup	Copy of an existing DASD data set written to tape for backup recovery purposes
Database Image Copy	Backup data created by database utilities

Identify these data sets to your storage class ACS routine and assign storage and management classes appropriate for the data set.

---

## Analyzing Your Current Tape Environment

The volume mount analyzer is a tool that helps you analyze your current tape environment. It identifies data sets that can be redirected to the DASD buffer for management using SMS facilities. It also identifies data sets that are very large and can benefit from advanced cartridge hardware and media technology.

The volume mount analyzer uses your SMF data to analyze tape mounts and associate them with the job and program initiating the request for the tape data set. It can also produce reports that identify information about tape mount events, including the data set name, job, program, and data set size (bytes transferred). Using the volume mount analyzer provides the following benefits:

- Quantify the benefits of tape mount management through library and tape mount reduction
- Evaluate your tape and cartridge hardware configuration
- Develop ACS routine filters to select tape mount management candidates and exclude other data sets that must remain on tape

For example, one set of filters can capture 90% of tape allocations that write single data set backups. You can then use this filter in an ACS routine to identify data that should be written to DASD with a management class specifying migration directly to tape.

- Determine the data class requirements
- Determine the management class requirements
- Determine the size of the DASD buffer and the high and low thresholds for this buffer's storage group

The volume mount analyzer features the following two programs:

- GFTAXTR reads the SMF data, and summarizes data from the tape-related records in a data set that can be processed by the analysis program. This data extraction program is usually only run once, normally with a month's worth of data.
- GFTAVMA does the major analysis and produces reports, helping you to understand the current tape environment. You can then design the optimized tape environment, using tape mount management techniques.

Figure 91 shows the major input to and output from the volume mount analyzer process.

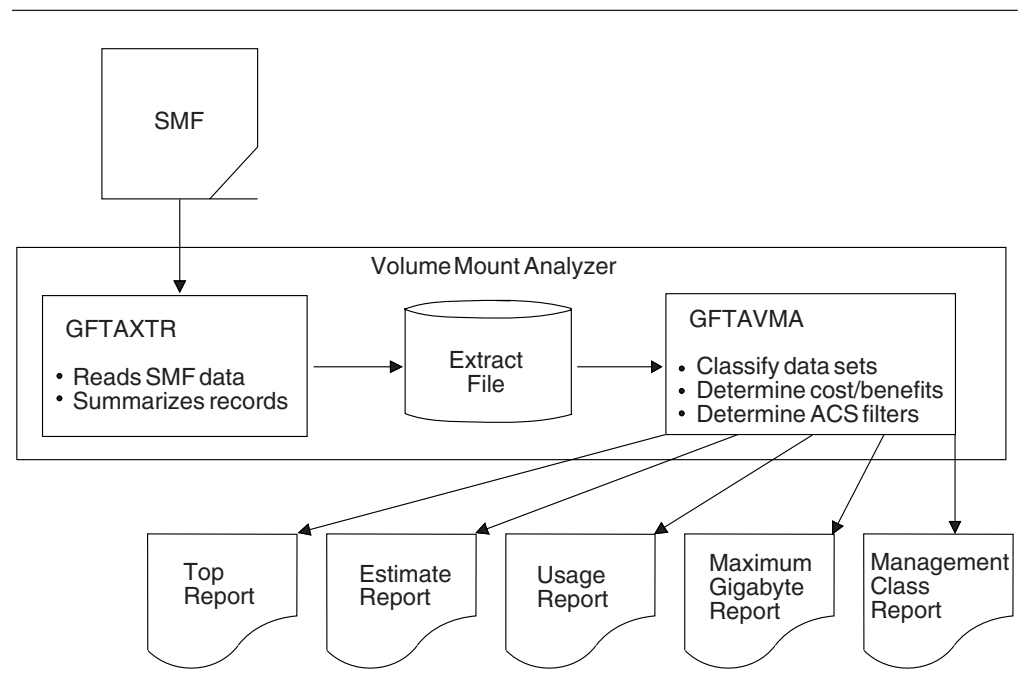


Figure 91. Volume Mount Analyzer Helps You Design Your Tape Mount Management Environment

## Summarizing Tape Usage with the Volume Mount Analyzer GFTAXTR Program

GFTAXTR matches each data set with the owning tape volume. The program creates summary records, containing data about the tape mount events, in a sequential file on DASD that serves as input to the analysis program. If GFTAXTR detects that SMF records required to profile a tape mount event are missing, it excludes any associated records from processing; minor occurrences of missing data, such as block size, are completed by GFTAXTR.

Summary statistics, including the total mount activity and number of tape data sets, are produced. Counts of single-volume, multivolume, single-file, and multi-file tape volumes are also reported. GFTAXTR attempts to assign a usage category to each data set. For example, data sets that are written and never read during the sample period are assumed to be point-in-time backups, and are assigned the *backup* usage category.

## Analyzing Tape Usage with the Volume Mount Analyzer GFTAVMA Program

GFTAVMA uses the information summarized by GFTAXTR to produce reports about your tape usage. Once you have used GFTAVMA to identify your major tape users, use its other reporting options to help you simulate a tape mount-managed environment. This simulation can help size the DASD buffer and determine the required management classes, data classes, and the filters to identify the tape data sets to be included or excluded from tape mount management.

GFTAVMA determines where each data set belongs in the storage hierarchy, based on a simulation of management class attributes, PRIMARY DAYS and LEVEL 1 DAYS.

You can specify values for these attributes or use the volume mount analyzer's defaults. Each data set has a set of management criteria, based on the usage classification that the volume mount analyzer assigns to the data set. Possible classifications include the following:

- DFSMShsm-owned
- Temporary
- Backup
- Single
- BCOPY
- Active

Data sets that are DFSMShsm-owned are on backup and migration tapes created by DFSMShsm processes.

Data sets classified as *temporary* have system-generated temporary data set names. These data sets are allocated on the primary or large storage groups, and are deleted by DFSMShsm during space management.

Data sets classified as *backup* are allocated to the DASD buffer and become eligible for migration the next time that space management is run. They migrate directly to migration level 2.

Data sets classified as *single* are referenced on only one day during the sample period. These data sets can be directed to migration level 2 after one day.

Data sets classified as *BCOPY* are data sets that are written and read once during the sample period. Typically, these might be copies of data sets to be taken offsite as part of your disaster recovery procedures. GFTAVMA does not make any assumptions about how these data sets should migrate through the storage hierarchy. They are handled as *active* data sets.

Any data sets not assigned a usage category are considered as *active*. They are allocated to the DASD buffer and migrate through the storage hierarchy according to user-specified or volume mount analyzer default values for the volume mount

analyzer keywords, LEVEL0DAYS and LEVEL1DAYS. The volume mount analyzer defaults are one day for LEVEL0DAYS and zero days for LEVEL1DAYS.

**Related Reading:** For more information about analyzing tape usage with the volume mount analyzer, see *z/OS DFSMS Using the Volume Mount Analyzer*.

### **Determining Your Tape Activity**

To prepare for this step, collect 30 to 45 days of your SMF records and use them as input to the volume mount analyzer's extract program, GFTAXTR. Use a normal business day as the ending date for the SMF collection. Use the output from the extract program as the input to all later analyses. Be sure to include data for the peaks in your tape workload.

This step introduces you to the volume mount analyzer reports and gives you an overall picture of tape use. These reports include the following:

- Estimate (EST) of your savings in tape mounts and volume counts as result of implementing the tape mount management methodology, and also the cost of additional DASD volumes for the DASD buffer
- Gigabyte (GB) report, showing you the maximum gigabyte allocations by hour, to help you determine free space requirements for the tape mount management DASD buffer
- Usage (USE) report, showing the maximum number of drives used concurrently over the sample period for your tape and cartridge subsystems
- Top (TOP) report, showing you which applications are major tape users

You can use the following GFTAVMA keywords to produce the initial volume mount analyzer reports:

```
REPORT (EST,GB,USE, TOP)
```

Use these initial reports to determine a strategy for follow-on analyses needed to design your ACS routines and classes and size of the DASD buffer.

### **Determining Mount and Volume Reductions**

An Estimate Report shows you the benefits of tape mount management implementation for mount and volume reductions, and the costs of additional DASD volumes for the DASD buffer and the DFSMSHsm RECALL activity. You can review the report to become familiar with the breakdown of your tape mounts and volumes by data category.

Figure 92 on page 186 is a sample of a savings and cost summary chart contained in an Estimate Report.

---

STATISTICAL MOUNT SAVINGS:				
1669.8	MOUNTS	(LARGE)	-	4.4%
0.0	MOUNTS	(HSM)	-	0.0%
32977.3	MOUNTS	(DFSMS)	-	86.1%
-----				
34647.2	MOUNTS		-	90.5%
- 3384.0	MOUNTS	(MGMTCLAS)	-	8.8%
-----				
31263.2	MOUNTS	(TOTAL)	-	81.6%
STATISTICAL VOLUME SAVINGS:				
234.0	VOLUMES	(LARGE)	-	3.7%
0.0	VOLUMES	(HSM)	-	0.0%
4484.3	VOLUMES	(DFSMS)	-	71.0%
-----				
4718.4	VOLUMES		-	74.7%
- 4.2	VOLUMES	(MGMTCLAS)	-	0.1%
-----				
4714.1	VOLUMES	(TOTAL)	-	74.6%
-----				
DASD BUFFER COST:				
25.0	3390-3	VOLUMES (PRIMARY)		
2.1	3390-3	VOLUMES (MGMTCLAS)		
1.7		VOLUMES (+ 'FSPACE'====>5%)		
0.0	3390-3	VOLUMES (HSM LEVEL 1)		
0.0	3390-3	VOLUMES (MGMTCLAS)		
0.0		VOLUMES (+ 'L1FSPACE'==>5%)		
-----				
28.8		TOTAL VOLUMES		

---

Figure 92. Savings and Cost Summary in Sample Estimate Report

The Estimate Report uses the current date as the last day of the input sample. DFSMSHsm runs automatic space management for the DASD buffer only once a day. The DASD buffer can be smaller if DFSMSHsm space management is performed hourly.

Compare the CURR STAT MOUNTS and TARG STAT MOUNTS columns in Figure 93 on page 187 to see the reduction of operator workload in managing all normal-sized tape data.

DATA SET CATEGORY	CURR STAT MOUNTS	TARG STAT MOUNTS	DFSMS - PRIMARY		HSM - LEVEL 1		HSM - LEVEL 2	
			CURR MNT#	TARG MNT#	CURR MNT#	TARG MNT#	CURR MNT#	TARG MNT#
TEMPORARY	242.0		242.0					
ACTIVE	2088.5	9.9	1483.1		0.0		605.4	9.9
ACTV GDG	6443.4	132.0	1930.6		0.0		4512.8	132.0
BACKUP	25476.8	1131.5	7916.6				17560.2	1131.5
LARGE	4050.0	2380.2						
HSM SFF	0.0	0.0						
HSM GEN	0.0	0.0						
MGMTCLAS		3384.0						3384.0
TOTAL	38300.8	7037.6	11572.3		0.0		22678.4	4657.4

Figure 93. Statistical Mount Savings Breakdown in Sample Estimate Report

**Note:**The fractional mounts and volume counts shown in the report are called statistical mounts and statistical volumes, respectively.

These mount and volume counts are fractional because of the presence of multiple data sets on a single tape volume. The volume mount analyzer assigns a partial mount or volume count when one of the data sets on a multi-file volume is requested. It does this so that volume mount analyzer filters, which include or exclude data sets, can properly predict tape mount management effects on the one data set, as opposed to the other data sets on the volume. Figure 94 shows the affect on tape library size.

DATA SET CATEGORY	CURR STAT VOLUMES	TARG STAT VOLUMES	DFSMS - PRIMARY		HSM - LEVEL 1		HSM - LEVEL 2		TAPE - DIRECT	
			CURR VOL#	TARG VOL#	CURR VOL#	TARG VOL#	CURR VOL#	TARG VOL#	CURR VOL#	TARG VOL#
TEMPORARY	48.2		48.2							
ACTIVE	86.8	9.9	18.7		0.0		68.1	9.9		
ACTV GDG	635.4	132.0	90.4		0.0		545.0	132.0		
BACKUP	4987.3	1131.5	760.7				4226.6	1131.5		
LARGE	557.8	323.8						557.8	323.8	
HSM SFF	0.0	0.0						0.0	0.0	
HSM GEN	0.0	0.0						0.0	0.0	
MGMTCLAS		4.2						4.2		
TOTAL	6315.5	1601.4	918.0		0.0		4839.7	1277.6	557.8	323.8

Figure 94. Statistical Volume Savings Breakdown: Sample Estimate Report

The Estimate Report also shows the DASD buffer requirements necessary to manage all normal-sized tape data. Figure 95 on page 188 displays a sample Estimate Report for DASD buffer sizings.

DATA SET CATEGORY	DSN#	ORIGINAL SIZE-MB	ORIGINAL SIZE-TK	DFSMS - DSN#	PRIMARY SIZE-TK	HSM - LEVEL 1 DSN#	SIZE-TK	HSM - LEVEL 2 DSN#	SIZE-MB	TAPE - DIRECT DSN#	SIZE-MB
TEMPORARY	141	3039	83217	6	659						
ACTIVE	200	6779	196782	29	33999	0	0	171	3942		
ACTV GDG	1685	71285	1977258	104	120649	0	0	1581	52769		
BACKUP	38794	761995	20359815	2030	1095199			36764	451926		
LARGE	418	466686	14200124							418	203378
HSM SFF	0	0	0							0	0
HSM GEN	0	0	0							0	0
MGMTCLAS					103622		0		1689		
TOTAL	41238	1309784	36817196	2169	1354128	0	0	38516	506948	418	203378

Figure 95. DASD Buffer Cost Breakdown: Sample Estimate Report

Using the Estimate Report from Figure 92 on page 186, you can observe the following:

- Tape mount management can save 81.6% of all mounts and 74.6% of all cartridges with 28.8 volumes of 3390-3 DASD (assuming that DFSMSHsm runs once every 24 hours). This can be reduced significantly for backup data that can be moved every hour using interval processing of DFSMSHsm.
- Based on the keywords set for this run, the simulation indicates that an 8.8% recall rate is incurred because of level 2 recalls from DFSMSHsm.
- Only a small portion of the mounts are saved (4.4%) by sending large files directly to a larger capacity cartridge.

From Figure 93 on page 187 and Figure 94 on page 187, you can observe that most of the mounts and tape volumes saved fall into the tape mount management category of backup, as follows:

- Mounts reduced from 25476.8 to 1131.5
- Volumes reduced from 4987.3 to 1131.5

From Figure 95, you can observe that because most of the data sets are backup data sets (38794 out of 41238), there is a good chance that the tape mount management DASD buffer can be dramatically reduced if hourly interval processing is done with DFSMSHsm.

### Determining Free Space Requirements

The Maximum Gigabyte (GB) Report shows the amount of new tape data (that is, DISP=NEW and DISP=MOD) being generated by hour. When you develop your list of tape mount management candidates, use this report to determine the free space requirements for the tape mount management DASD buffer.

Table 28 shows that the free space for the tape mount management buffer should be about 6 GB to support the period with the highest allocation activity.

Table 28. Maximum Gigabyte Allocations by Hour Report

MAX GB/ HR	00	01	02	03	04	05	06	07	08	...	20	21	22	23	MAX	TOT
HR MAX	6	6	5	3	5	5	4	6	6	...	5	6	5	6	6	2046

## Determining Your Tape Hardware Use

The Usage Report shows the maximum number of drives used concurrently over the sample period for your tape and cartridge subsystems. Currently, 85 drives must be available to support peak tape activity. Later, you can use this report to reduce these configurations, based on tape mount management implementation results.

Table 29 displays an example of a Usage Report that provides a summary of tape mounts.

Table 29. Maximum Tape Mounts: 18 Day Summary Report

	HR 00	01	02	03	04	05	...	17	18	19	20	21	22	23	MAX	TOT MNTS
HR MAX	49	64	48	54	43	23	...	19	49	85	65	5	50	72	85	6356

## Determining Major Tape Users

The Top Report shows which applications are major tape users. You can generate a separate report by job name, job account code, program name, data set name HLQ and LLQ, expiration date, date the data set was last referenced, and data set age.

The job-related variables are listed so that you can start to identify patterns you can translate into ACS routine FILTLIST statements for ACS read-only variables, such as job name, job accounting information, data set or program name.

Your standards determine the variables that are most useful in reliably identifying how your tape data sets should be processed. Program name is normally the most efficient filter to capture a lot of mounts, as a single program is often responsible for multiple data sets stacked on a tape. Intercepting multi-file sets of data sets together removes the additional complexities involved in breaking up these sets of data. You can also use job names or accounting information as filters.

Your ACS routines use the filters that you code in FILTLIST statements to intercept tape allocations for the tape mount management candidates. Take note of the patterns in variables such as data set name or program name.

Table 30 shows data from a Top Report for the most tape-intensive users by program name.

Table 30. First Five Entries from a Top Report: Program Names-LARGE = 600 MB

RANK	PROG	# DSNS	% TOT	CUM DSN	% TOT	> LARGE	AVG SIZE	# MNTS	% TOT	CUM MNTS	% TOT
1	ADRDSSU	556	14.8	556	14.8	489	759.8	1532.0	24.1	1532.0	24.1
2	IDCAMS	1097	29.3	1653	44.1	22	99.9	1397.9	22.4	2929.9	46.1
3	ARCCTL	4	0.1	1657	44.2	2	17573	728.0	11.5	3657.9	57.5
4	COPY	672	17.9	2329	62.1	5	21.8	573.0	7.0	4230.9	66.6
5	ICEMAN	116	3.1	2445	65.2	2	57.2	417.5	6.6	4648.4	73.1

This report identifies the five top-ranked programs accounting for 73.1% of all tape mount activity and 65.2% of all tape data sets during the sample period. AVG SIZE shows the average size in MBs of the input and output data sets for the five programs. The column, > LARGE, shows the number of data sets processed by the program that exceeded the LARGE value, 600 MB, for the sample. The DFSMSDss's ADRDSSU program and DFSMSHsm's ARCCTL program produce data sets that typically exceed the size of a good tape mount management candidate.



For the ADRDSSU application in Table 30 on page 189, the average size of all data sets, 759.8 MB, and the size of 489 data sets (out of 556 data sets) are greater than the LARGE value of 600 MB. This indicates that the entire application (that is, PGM=ADRDSSU) should be considered as LARGE and sent directly to tape.

For this sample installation, use program name as a filter in ACS FILTLISTS to exclude ADRDSSU and ARCCTL from tape mount management. However, IDCAMS, COPY, and ICEMAN output are good tape mount management candidates, based on data set size, representing a large portion of the mounts (22.4%, 7.0%, and 6.6%, respectively). Therefore, you would include IDCAMS, COPY, and ICEMAN in the ACS FILTLISTS so that these programs are included for tape mount management.

Similarly, the Top Report for data set HLQ shows that several HLQs are linked with large tape data sets. Filters based on data set name are discussed in “Identifying and Excluding Large Data Sets” on page 191.

---

## Simulating the Tape Mount Management Environment

This section provides a recommended sequence of volume mount analyzer analyses, including keywords and sample output, to help you design your ACS routines and classes. It also helps you size the DASD buffer and determine your free space requirements for tape mount management volumes.

**Recommendations:** When designing ACS routines and classes, consider the following:

- Successful implementation of tape mount management depends on ACS routine design, using the least number of filters to represent the most mount activity.
- ACS routines designed to handle *all* exceptions seldom work correctly and consistently.

There is little consistency in data set size over time, so that this month's large data sets are not the same as next month's.

Avoid developing extensive FILTLISTS (> 100 filters) to exclude *all* large data sets from being allocated on the DASD buffer. Allocate more DASD buffer volumes rather than increasing the ACS routine design complexity and maintenance requirements.

- Avoid using data set size thresholds, such as 50 MB or 100 MB, to determine what data sets should be directed to the DASD buffer.

This introduces additional complexity and implementation difficulties, and reduces benefits.

To run your sequence of volume mount analyzer simulations, perform the following steps:

- Identify exception tape data
- Define tape mount management environment

## Identifying Exception Tape Data

This step helps you identify data sets that are *not* candidates for re-direction to the DASD buffer. They are either too large, belong to jobs with special requirements, or are DFSMSHsm-owned.

Identify exception tape data sets to your ACS routines by filters and assign them to a null storage class to bypass system management. Data sets assigned a null storage class are written to tape as before.

To achieve optimal tape performance, create these data sets using advanced cartridge subsystems, such as the 3490E with Enhanced Capacity Cartridge System Tape.

To identify exception tape data, perform your volume mount analyzer simulations in the following sequence:

1. Identify and exclude large data sets
2. Develop filters for large data sets
3. Identify and exclude special jobs or data sets
4. Finalize exclusion filter list

### **Identifying and Excluding Large Data Sets**

You need to create filters for your large data sets. The volume mount analyzer default for large is 600 MB, the amount of data that can be contained on a single cartridge system tape, compacted three-fold with Improved data recording capability (IDRC). You can supply your own value for LARGE. Look for programs or HLQs that tend to generate primarily large data sets, and exclude those entire filters. Consider as LARGE those data sets that are large enough to make efficient use of available tape capacity. Use the LARGE value used by the volume mount analyzer as an indicator rather than an absolute cutoff value. In the sample, most of the tape data sets are candidates for the DASD buffer, but some of the very large data sets should continue to go directly to tape.

For the sample, program name identifies some large data sets that account for many mounts. Typically, program name can be used as a filter to identify mounts for large data sets that can be excluded with ACS filtering techniques. Often, one program is responsible for creating multiple data sets stacked on tape. Intercepting mounts for multi-file data sets in this way simplifies your design problems linked with breaking up these stacked data sets.

Use the Top Report to identify efficient filters for capturing many large data sets. The >LARGE and AVG SIZE columns give a good indication of the tendency of particular programs or applications to generate LARGE data.

### **Developing Filters For the LARGE Category**

Use the names or patterns that you have identified in the last step as filters to the volume mount analyzer. Set the LARGE keyword value to 0 to tell the volume mount analyzer that all of the data sets represented by this filter should continue to go to tape directly. The Estimate Report shows the mount and volume savings expected from managing all of these large data sets directly on tape with advanced cartridge technology.

Use the following volume mount analyzer request to predict the affect of the 3490E with Enhanced Capacity Cartridge System Tape on large tape data set activity:

```
REP(USE,EST)
PGM(INC(ADRDSU,ARCCTL))
UNIT(EXC(3420))
LARGE(0)
TAPELEN(2)
```

The sample request for estimate shows the reduction in mounts and volumes when all tape data created is written using 3490E technology. Table 31 on page 192 shows that, for the programs included so far in the filters, mounts can be reduced by 54.2% and volumes by 61.5%, if data is written using the 3490E with both IDRC and Enhanced Capacity Cartridge System Tape. The program name filters become

part of an exclusion list of variables. This list is used as input to the next volume mount analyzer analysis and becomes part of your ACS routine logic.

Table 31. Savings and Cost Summary: Savings From 3490E Implementation

STATISTICAL MOUNT SAVINGS:				
1529.8	MOUNTS	(LARGE)		47.4%
223.0	MOUNTS	(HSM)		6.9%
0.0	MOUNTS	(DFSMS)		0.0%
<hr/>				
1752.8	MOUNTS			54.3%
- 2.0	MOUNTS	(MGMTCLAS)		0.1%
<hr/>				
1750.8	MOUNTS	(TOTAL)		54.2%
STATISTICAL VOLUME SAVINGS:				
395.9	VOLUMES	(LARGE)		50.8%
83.0	VOLUMES	(HSM)		10.7%
0.0	VOLUMES	(DFSMS)		0.0%
<hr/>				
478.9	VOLUMES			61.5%
+ 0.0	VOLUMES	(MGMTCLAS)		0.0%
<hr/>				
478.9	VOLUMES	(TOTAL)		61.5%
DASD BUFFER COST:				
0.0	3390-3	VOLUMES	(PRIMARY)	
0.0	3390-3	VOLUMES	(MGMTCLAS)	
0.0		VOLUMES	(+'FSPACE'⇒5%)	
0.0	3390-3	VOLUMES	(HSM LEVEL 1)	
0.0	3390-3	VOLUMES	(MGMTCLAS)	
0.0		VOLUMES	(+'L1FSPACE'⇒5%)	
<hr/>				
0.0	TOTAL	VOLUMES		

## Identifying and Excluding Special Jobs

Exclude some data sets from tape mount management as the data needs to remain on tape for the following reasons:

- Disaster recovery
- Vital records
- Interchange data

These data sets sometimes have a recognizable pattern, such as the following:

- `"*.DR.*"` can identify the disaster recovery data sets
- `EXPDT=9800` usually indicates offsite tapes to the tape management system
- `VRJOB` is the job name for all the vital records jobs.

**Tip:** If your tape management system provides support for the pre-ACS routine exit and supplies the `MDEST` parameter, you might be able to identify most of these data sets by using an `MSDEST` parameter value other than `'NULL'`.

## Finalizing the Exception Filter List

This step excludes the effects of all exception tape data sets identified to this point, and generates the baseline profile for tape mount management candidates that will be system-managed. You have developed filters to exclude large and special data

sets. Now, specify USAGE(NONHSM) to tell the volume mount analyzer to exclude all DFSMSHsm-owned tape volumes from the baseline data.

The following keywords are required to assess the affect of 3490E hardware on tape data sets that will continue to be written directly to tape:

```
REP(EST)
TAPEDEV(3490E)
EXPDT(INC(98000))
JOB(INC(VRJOB))
PGM(INC(ADRDSU,ARCCTL))
DSN(INC(*.DR.**))
```

## Defining the Tape Mount Management Environment

By excluding all of the tape mount management exceptions (large, DFSMSHsm, and special data sets), the remaining list has the data sets that are good tape mount management candidates. For the previous example, you would specify the following keywords:

```
PGM(EXC(ADRDSU,ARCCTL))
EXPDT(EXC(98000))
DSN(EXC(*.DR.**))
JOB(EXC(VRJOB))
```

After you identify your tape data sets, develop a set of characteristics for them and:

- Determine the management classes
- Determine the space values for data classes
- Quantify tape mount management benefits and costs
- Determine free space requirements
- Determine tape and cartridge configuration changes

### Determining the Management Classes

After you identify the filters to intercept the allocations and redirect them to DASD, determine the proper management class attributes, such as residency time on primary and migration level 1 storage. Consider the amount of space required to store the data set on primary or migration level 1 storage versus the number of mounts required to recall the data. The volume mount analyzer produces a report ranking all of the management classes by number of migration level 2 mounts, number of recalls, and additional tracks needed for the migration and recall activity for that management class.

The Management Class Report is sorted in descending order by L2-MOUNTS, RECALLS, and TOT TRKS. The most important value to keep low is the L2-MOUNTS, because it represents specific mounts. All other tape mount management mounts are scratch mounts. When you select a management class, you should rerun the Estimate Report with that management class.

The following control statements generate a Management Class Report for all data sets created by PGM=IDCAMS, but not those on 3420 tapes:

```
PGM(INC(IDCAMS))
UNIT(EXC(3420))
REP(MC)
```

Table 32 on page 194 shows the five top-ranked management classes for the sample installation's tape mount management data sets generated by IDCAMS.

Table 32. Management Class Report for IDCAMS Data. L0AGE is the number of days that the data set resides on primary storage before migration. L1AGE shows the number of days that the data set resides on migration level 1.

ORDER	L0AGE	L1AGE	L2-MNTS	RECALLS	TOT TRKS	L0-TRKS	L1-TRKS	L2-MB
1	1	1	451	451	3191	3191	0	3
2	0	1	451	451	7	0	7	0
3	0	0	451	451	0	0	0	8
4	1	2	99	451	3192	3191	1	0
5	0	2	99	451	8	0	8	0

### Determining the Space Values for Data Classes

To determine the space values for data classes, look at the tape data set size range table produced by the GFTASRT3 phase of the volume mount analyzer. In the sample table shown in Figure 96, 89.8% of the data sets are less than, or equal to, 50 MB.

DSN SIZE	# OF DSNS	% TOT DSNS	CUM # DSNS	% TOT DSNS	# OF STAT MOUNTS	% TOT MNTS	CUM STAT MOUNTS	% TOT MNTS	# OF STAT VOLUMES	% TOT VOLS	CUM STAT VOLUMES	% TOT VOLS
= 0MB	136	1.1	136	1.1	113.4	1.7	113.4	1.7	19.6	1.5	19.6	1.5
<= 1MB	7812	63.1	7948	64.2	3515.5	53.2	3628.9	54.9	785.9	59.8	805.6	61.3
<= 5MB	1149	9.3	9097	73.4	503.1	7.6	4132.0	62.5	84.3	6.4	889.9	67.7
<= 10MB	402	3.2	9499	76.7	249.6	3.8	4381.7	66.3	53.2	4.0	943.1	71.8
<= 25MB	1211	9.8	10710	86.5	1153.7	17.5	5535.4	83.7	196.0	14.9	1139.1	86.7
<= 40MB	323	2.6	11033	89.1	318.0	4.8	5853.4	88.5	60.9	4.6	1200.0	91.3
<= 50MB	84	0.7	11117	89.8	50.1	0.8	5903.5	89.3	9.7	0.7	1209.7	92.1
<= 75MB	411	3.3	11528	93.1	117.3	1.8	6020.8	91.1	31.4	2.4	1241.1	94.5
<= 100MB	332	2.7	11860	95.8	305.9	4.6	6326.7	95.7	26.9	2.0	1268.0	96.5
<= 125MB	172	1.4	12032	97.1	112.1	1.7	6438.8	97.4	15.7	1.2	1283.7	97.7
<= 150MB	47	0.4	12079	97.5	26.3	0.4	6465.1	97.8	2.5	0.2	1286.2	97.9
<= 175MB	76	0.6	12155	98.1	22.4	0.3	6487.5	98.1	6.8	0.5	1292.9	98.4
<= 200MB	74	0.6	12229	98.7	3.8	0.1	6491.3	98.2	1.3	0.1	1294.3	98.5
<= 300MB	95	0.8	12324	99.5	4.3	0.1	6495.6	98.2	1.0	0.1	1295.3	98.6
<= 400MB	31	0.3	12355	99.7	58.0	0.9	6553.6	99.1	9.2	0.7	1304.5	99.3
<= 500MB	0	0.0	12355	99.7	0.0	0.0	6553.6	99.1	0.0	0.0	1304.5	99.3
<= 600MB	31	0.3	12386	100.0	58.0	0.9	6611.6	100.0	9.5	0.7	1314.0	100.0

Figure 96. Example of a Tape Data Set Size Range Table

### Quantifying Tape Mount Management Savings and Costs

For the IDCAMS program, assume that an L0AGE(1) and L1AGE(3) provide the best values overall. Also, in the original Top Report (Table 30 on page 189), there are 22 data sets out of 1097 data sets that are greater than 600 MB. Because that is less than one large data set per day, assume that you will buffer these data sets to keep the filtering simple.

Now you need to run the Estimate Report using all of the assumptions. Because you chose L0AGE(1) and L1AGE(3), concede that all data sets created by IDCAMS are active. Specify this to the volume mount analyzer to force all of these data sets into the active category. Also, because you are not excluding the 22 LARGE data sets, set the LARGE value high enough so the volume mount analyzer does not separate them into LARGE.

The following keywords request the storage sizing information for this subset of data produced by IDCAMS.

```
PGM(INC(IDCAMS))
CLASSIFY(ACTIVE)
L0AGE(1)
L1AGE(3)
LARGE(99999)
TAPEDEV(3490E)
TAPELEN(2)
REP(EST)
```

Figure 97 shows the savings and cost summary. This report assumes that the current date is the last day of the input sample, and that DFSMSHsm processes the tape mount management DASD buffer only once a day.

---

STATISTICAL MOUNT SAVINGS:				
0.0	MOUNTS	(LARGE)	-	0.0%
0.0	MOUNTS	(HSM )	-	0.0%
6569.1	MOUNTS	(DFSMS)	-	99.4%
-----				-----
6569.1	MOUNTS		-	99.4%
- 196.0	MOUNTS	(MGMTCLAS)	-	3.0%
-----				-----
6373.1	MOUNTS	(TOTAL)	-	96.4%
STATISTICAL VOLUME SAVINGS:				
0.0	VOLUMES	(LARGE)	-	0.0%
0.0	VOLUMES	(HSM )	-	0.0%
1271.4	VOLUMES	(DFSMS)	-	96.8%
-----				-----
1271.4	VOLUMES		-	96.8%
- 0.9	VOLUMES	(MGMTCLAS)	-	0.1%
-----				-----
1270.5	VOLUMES	(TOTAL)	-	96.7%
-----				
DASD BUFFER COST:				
3.5	3390-3	VOLUMES	(PRIMARY)	
0.1	3390-3	VOLUMES	(MGMTCLAS)	
0.2		VOLUMES	(+ 'FSPACE'====> 5%)	
2.1	3390-3	VOLUMES	(HSM LEVEL 1)	
0.1	3390-3	VOLUMES	(MGMTCLAS)	
0.0		VOLUMES	(+ 'L1FSPACE'==> 5%)	
-----				
6.0		TOTAL	VOLUMES	

---

Figure 97. Savings and Cost Summary

Now decide if you are satisfied with the projected improvements. If not, go back to "Identifying and Excluding Large Data Sets" on page 191, exclude the data sets

that you have already determined how to manage, and try to identify more potential tape mount management candidates.

### Determining Free Space Requirements

Determine the amount of space required to satisfy new allocations. When you determine the final estimates for each of the major applications, their sum is the total tape mount management DASD buffer estimate. This sum does not include the real free space needed, based on the actual traffic of the tape mount management management candidates. You can determine this by excluding all of the tape mount management exclusions, and generating a Maximum Gigabyte (GB) Report.

Enter the following keywords to determine your free space requirements:

```
REP(GB)
JOB(EXC(VRJOB))
EXPDT(EXC(98000))
PGM(EXC(ADRDSU,ARCCTL))
DSN(EXC(*,DR.**))
UNIT(EXC(3420))
```

Table 33 shows the report, displaying the maximum GB allocations by hour.

Table 33. Sample Maximum Gigabyte Allocations by Hour Report

MAX GB/ HR	00	01	02	03	04	05	06	07	08	09	10	11	12	..	20	21	22	23	MAX	TOT
HR MAX	3	3	2	2	1	0.4	<.1	<.1	0.1	0.2	<.1	0.1	0.1	..	0.2	0.3	2	3	3	294

Divide the maximum GBs required (3 GB in the sample, report displayed in Table 33) by the total size of the DASD buffer, to obtain the free space percentage, as shown in the following formula:

$$\text{free space \%} = \frac{\text{maximum GB required}}{\text{total size of DASD buffer}}$$

Use the free space percentage to set the low threshold for the DASD buffer storage group. For example, the final sum of all the estimates was 12 3390-3 volumes for level 0. The 3GB maximum each hour is a little larger than one 3390-3 volume. This is approximately 10% of the total estimate (12 volumes). If you choose a high threshold of 95%, the low threshold would be 85%, or even 80% to allow for some growth. All of this logic assumes that DFSMSshsm processes the tape mount management DASD buffer every hour.

### Determining Tape and Cartridge Configuration Changes

The total number of tape and cartridge transports, and the maximum number of drives used concurrently, were determined from the first analysis. Because allocations for tape mount management candidates are intercepted and rerouted to DASD, you should re-examine tape requirements and check if you can reduce the number of tape transports.

Enter the following keywords to determine the tape configuration effect:



```

PGM(EXC(ADRSSU,ARCCTL))
EXPDT(EXC(98000))
JOB(EXC(VRJOB))
DSN(EXC(*.DR.**))
UNIT(EXC(3420))
REP(USE)

```

Table 34 displays an 8-day summary report of maximum tape allocations.

Table 34. Sample Report of Maximum Tape Allocations

	HR	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	MAX	TOT
	00																									MNTS
HR	5	12	12	12	11	6	4	2	2	2	5	5	6	6	6	4	6	3	6	8	9	1	4	6	12	3232
MAX																										

**Rule:** There are increased requirements for cartridge transports resulting from increased level 2 migration activity storage in the new environment. Consider these requirements when sizing the cartridge configuration.

## Implementing Advanced Cartridge Hardware

Advanced cartridge hardware helps you optimize your use of the cartridge subsystems and media with the following features:

- Improved data recording capability
- Advanced cartridge technology hardware
- Enhanced Capacity Cartridge System Tape
- Automatic and integrated cartridge loaders

### Using the Improved Data Recording Capability

Improved data recording capability (IDRC) is a data compaction feature of the 3480 subsystem, and part of the base model of the 3490 and 3490E cartridge controller. Data compaction in the cartridge control unit eliminates the CPU overhead from software compression. IDRC-compacted data is created using a technique called *auto-blocking*, which ensures optimal physical blocking for cartridge media. The block size is unaffected by this technique. A typical compaction ratio is 3:1; however, inefficiently blocked tape data sets can achieve more dramatic compaction.

IDRC helps to reduce the number of cartridges required to store data, and reduces the elapsed time for batch jobs depending on cartridge I/O. This makes IDRC effective for both single volume and multivolume data sets. It optimizes the data exchange between the controller and the device, increasing the number of devices that can be used concurrently.

To implement IDRC with SMS data classes, perform the following steps:

- Define data classes for your offsite tape data and very large backup, active, and temporary tape data sets. Set `COMPACTION` to `Y` for these data classes. Data classes for these tape data set categories are `TAPOSITE`, `TAPBKUP`, `TAPACTV`, and `TAPTEMP`.
- Allow your data class ACS routine to assign these classes during allocation. This method writes the data in IDRC-compacted format. The same result occurs if you specify `DCB=TRTCH=COMP` on the DD statement.

Remember that the data class ACS routine is driven for both system-managed and non-system-managed data sets.



Using data class lets the system determine an optimal block size for the tape data set if you do not specify one. *z/OS DFSMS Using Data Sets* describes how the system determines block size for tape. Using system-determined block size improves tape channel usage and buffer management.

- Define a data class, NONTMM, with COMPACT=N, so tape data sets are directed to tape in a non-compacted form. Use the NONTMM data class on your DD statements to tell your data class ACS routine that the data sets should not be compacted or redirected to the DASD buffer. These might be data sets shipped offsite to facilities without IDRC-capable drives or those used by applications that call the READ BACKWARDS command. This command is simulated for IDRC data sets; compacting them severely degrades performance. Few data set types rely on the READ BACKWARDS command. For example, the IMS log is read backwards if it is processed during recovery. Do not compact DFSORT work files.

## Using the Enhanced Recording Capability

The 3490E cartridge controller offers enhanced recording capability. Thirty-six-track recording doubles the capacity of a cartridge and eliminates rewind times for full cartridges. Through the combination of IDRC and the enhanced recording capability, media use can, on average, improve six-fold.

## Using Enhanced Capacity Tape

The 3490E cartridge controller also offers support for Enhanced Capacity Cartridge System Tape, which is a thinner media. A single Enhanced Capacity Cartridge System Tape is twice the length of standard tape, doubling the capacity of a cartridge. The 3490E with Enhanced Capacity Cartridge System Tape can contain 2.4 GB of data, compared to the standard 200 MB of non-compacted tape. This capacity lets a single cartridge to contain one full 3390-3 volume.

## Using Automatic Cartridge Loaders

Use automatic or integrated cartridge loaders (ACLs or ICLs) to reduce cartridge mount time for multivolume, for example DFSMSHsm migration level 2 and backup volumes, and scratch mounts. Typically, scratch mounts represent about 60% to 80% of all tape mounts. Directing these mounts to devices with ACLs or ICLs can significantly reduce mount times, compared to operator performed mounts.

---

## Defining DASD to Satisfy Buffer Requirements

Estimate the number of volumes required to support the DASD buffer. You can dedicate volumes to do this, because their management requirements are different from other volumes.

**Note:**When an overflow storage group contains more volumes than a buffer storage group, specified volume counts might result in volumes in the overflow storage group being preferred over volumes in the buffer storage group during volume selection.

---

## Defining SMS Classes and Groups

Data classes help to automate the allocation of tape mount management candidates on the DASD buffer. During your initial implementation of tape mount management, you can supply space information directly for tape data sets that are allocated without using the SPACE parameter. SMS data classes can specify this space without changing JCL.

Data classes can also be used as artificial classes, or flags, in your ACS routines to determine the data category that the tape data set belongs to, and whether to system manage it. You can set these artificial classes in the data class ACS routine, and check them in the storage class ACS routine when you determine if the data set is to be system-managed.

In general, tape mount management data sets do not have special performance or availability requirements, so new storage classes are not required. After you gain experience with tape mount management implementation, evaluate whether some candidates might benefit from sequential data striping.

The TMMBKUP and TMMACTV management classes support the two major categories of tape mount management data. TMMBKUP data moves to migration level 2 directly. TMMACTV data resides in the DASD buffer and migrates normally through the storage hierarchy, based on frequency of use.

Define new storage groups to support tape mount management data.

**Recommendation:** Define at least one separate storage group for the DASD buffer, because the threshold management policy for this set of volumes differs from others. Set the low threshold based on the space requirements for new tape mount management data allocations during periods of peak tape usage. Set the high threshold so a full cartridge of data, at least, is written during interval migration.

Consider setting up an overflow storage group of volumes (also known as a "spill" storage group). An overflow storage group serves as a reserve area for primary space allocations during periods of high space utilization in your pool storage groups. You can designate an overflow storage group in ISMF using the Pool Storage Group Define Panel. Assign designated overflow storage groups in your storage group ACS routine to make them eligible for allocations.

During volume selection, volumes in an overflow storage group are less preferred than those in an enabled storage group but more preferred than those in a quiesced storage group.

**Exception:** When an overflow storage group contains more volumes than a buffer storage group, specified volume counts might result in volumes in the overflow storage group being preferred over volumes in the buffer storage group.

Table 35 on page 200 summarizes the types of data sets, tape mount management technique, and their corresponding SMS classes and groups.

Table 35. SMS Classes and Groups for Tape Data Sets

Data Type	Technique	Storage Class	Data Class	Management Class	Storage Group
Interchange	Advanced Cartridge HW	Null	TAPOSITE	—	—
Disaster	Advanced Cartridge HW	Null	TAPOSITE	—	—
Vital Record	Advanced Cartridge HW	Null	TAPOSITE	—	—
DFSMSHsm Backup	Advanced Cartridge HW	Null	—	—	—
DFSMSHsm Migration	Advanced Cartridge HW	Null	—	—	—
Auto Dump	Advanced Cartridge HW	Null	—	—	—
Volume Dump	Advanced Cartridge HW	Null	—	—	—
Very Large Image Copy	Advanced Cartridge HW	Null	TAPBKUP	—	—
Very Large Temporary	Advanced Cartridge HW	Null	TAPTEMP	—	—
Large Active Data Set	Advanced Cartridge HW	Null	TAPACTV	—	—
Temporary	tape mount management	STANDARD	TMMTEMP	—	LARGE90, LARGE80
Active	tape mount management	STANDARD	TMMACTV	TMMACTV	TMMBUF90, TMMBUF80, TMMBFS90, TMMBFS80
Point-in-time Backup	tape mount management	STANDARD	TMMBKUP	TMMBKUP	TMMBUF90, TMMBUF80, TMMBFS90, TMMBFS80
Database Image Copy	tape mount management	STANDARD	TMMBKUP	TMMBKUP	TMMBUF90, TMMBUF80, TMMBFS90, TMMBFS80

## Defining Data Classes

Use the results of your tape mount management analysis to help determine the sizes of the tape data sets that you direct to DASD. You can create several data classes to correctly allocate space for these data sets and develop ACS filter lists to

help assign the proper data classes to the data sets. You can also have the user code the proper data class on the JCL DD statement. Many tape allocations do not specify the SPACE parameter, which is required when allocating data sets on DASD. You should plan to specify space parameters in your tape mount management data classes. This avoids situations where SPACE is not specified and a tape data set allocation is rerouted to DASD, causing the allocation to fail.

Using the Partial Release management class attribute can reduce the number of data classes needed to allocate space for tape mount management candidates. The CI option, *release-on-close*, releases allocated but unused space when the data set is closed or during the next space management cycle.

The following sample data classes are defined for use with tape mount management data sets:

**TAPOSITE**

Assigned to data sets that are usually stored offsite, such as vital records, disaster recovery backups, archives, and interchange data.

These tape data sets are allocated directly to tape.

**TAPACTV**

Assigned to active data sets larger than 600 MB. These data sets are allocated directly to tape.

**TAPBKUP**

Assigned to backup data sets, such as volume dumps and database image copies, that are larger than 600 MB. These data sets are allocated directly to tape.

**TAPTEMP**

Assigned to system-generated temporary data sets that are larger than 600 MB. These data sets are allocated directly to tape.

**TMMACTV**

Assigned to active data sets that are generally not LARGE. This data can be directed to the DASD buffer storage group as multivolume data sets (up to 10 volumes maximum).

These data sets are not usually allocated with a space request.

Data assigned to TMMACTV that contains no space information has a 200 MB primary allocation and a 20 MB secondary allocation.

**TMMBKUP**

Assigned to backup tape data sets that are generally not LARGE. This data can be directed to the DASD buffer storage group.

These data sets are not usually allocated with a space request.

Data assigned to TMMBKUP that does not contain space information has a 200 MB primary allocation and a 20 MB secondary allocation.

**TMMTEMP**

Assigned to system-generated, temporary tape data sets that are generally not LARGE. Data assigned to TMMTEMP that contains no space information has a 200 MB primary allocation and a 20 MB secondary allocation.

**NONTMM**

Assigned to data sets that should go to tape directly in a non-compacted

form. This data is not intended for tape mount management. Only storage administrators and system programmers can assign this class.

## Defining Storage Classes

This storage class is for use with tape mount management data sets.

### STANDARD

Assigned as the default for tape data sets directed to the tape mount management DASD buffer.

## Defining Management Classes

The following sample management classes are for use with tape data sets:

### TMMACTV

Assigned to active data sets. If they are RECALLED, they are directed to primary volume storage groups: PRIME90, PRIME80, LARGE90, or LARGE80.

If the data set is a member of a GDG, the latest generation is kept on the primary volume, and the older generations are migrated to migration level 1. Rolled-off GDSs are expired, and become candidates for deletion by DFSMSHsm.

### TMMBKUP

Assigned to all categories of backup data sets that are to be migrated directly from the tape mount management DASD buffer storage group volumes to DFSMSHsm migration level 2 tape volumes.

Table 36 shows the attributes of the management classes assigned by the ACS routine, shown in Figure 101 on page 208.

**Related Reading:** For descriptions of the management class attributes, see *z/OS DFSMSdfp Storage Administration*.

*Table 36. Sample Management Classes for Tape Mount Management*

ATTRIBUTE	TMMBKUP	TMMACTV
EXPIRE NON-USAGE	15	200
EXPIRE DATE/DAYS	NOLIMIT	NOLIMIT
MAX RET PERIOD	20	NOLIMIT
PARTIAL RELEASE	YI	YI
PRIMARY DAYS	0	2
LEVEL 1 DAYS	0	10
CMD/AUTO MIGRATE	BOTH	BOTH
#GDS ON PRIMARY	–	1
ROLLED OFF GDS ACTION	–	EXPIRE
BACKUP FREQUENCY	–	1
#BACKUPS (DS EXISTS)	–	2
# BACKUPS (DS DELETED)	–	1
RETAIN® DAYS ONLY BACKUP	–	60
RETAIN DAYS EXTRA BACKUP	–	30

Table 36. Sample Management Classes for Tape Mount Management (continued)

ATTRIBUTE	TMMBKUP	TMMACTV
ADM/USER BACKUP	NONE	ADMIN
AUTO BACKUP	NO	YES
BACKUP COPY TECHNIQUE	S	S

## Defining Storage Groups

Use the following sample storage groups with data sets normally allocated on tape, but eligible for tape mount management:

### TMMBUF80/90

Assigned to data directed to the DASD buffer for migration to either migration level 2 tape or migration level 1 DASD volumes.

Set the MIGR HIGH threshold to 95%, because the data sets routed to the buffer are typically less than 600 MB. Interval migration is called when the threshold exceeds 95%.

Set the MIGR LOW threshold to the value determined by the GB Report (for example, 80%) for DFSMSHsm to keep this group available for new allocations. DFSMSHsm migrates all data that is eligible on an hourly basis.

### TMMBFS80/90

Assigned to very large data sets that exceed the amount of free space available in the TMMBUFxx storage group.

We set the MIGR HIGH threshold to 25% to accommodate the large data set sizes. Interval migration is called when the threshold exceeds 25%. Data sets from this group are migrated frequently to ensure as much free space as possible for other large data set allocations.

Set the MIGR LOW threshold to 0% for DFSMSHsm to keep this group available for additional large allocations after interval migration processing.

Table 37 shows the attributes of the storage groups assigned by the ACS routine, shown in Figure 102 on page 209. See *z/OS DFSMSdfp Storage Administration* for a description of storage group attributes.

Table 37. Sample Storage Groups for Tape Mount Management

STORGRP NAME	TMMBUFxx	TMMBFSxx
POOL TYPE	POOL	POOL
AUTO MIGRATE	INTERVAL	INTERVAL
AUTO BACKUP	YES	YES
AUTO DUMP	NO	NO
MIGR HIGH	95	25
MIGR LOW	80	0
VOLUME LIST	TMM001 TMM002 TMMnnn	TMMS01 TMMS02 TMMSnn

## Creating ACS Routines

ACS routines assign storage, data, and management classes to data sets that are intercepted and redirected to DASD. The storage group ACS routine then assigns data sets to the appropriate storage group.

DFSMSrmm supports the SMS pre-ACS interface. The SMS subsystem calls DFSMSrmm before the data class ACS routine obtains control. Then, DFSMSrmm optionally sets the initial value for the ACS routine MSPPOOL and MSPOLICY read-only variables if the pre-ACS installation exit has not done so. DFSMSrmm however does not use the installation exit.

**Related Reading:** For detailed information on the DFSMSrmm support for the SMS pre-ACS interface, see *z/OS DFSMSrmm Implementation and Customization Guide*. Or, check with your tape management vendors for information on support for the variables.

## Creating the Data Class ACS Routine

Figure 98 on page 205 shows the FILTLIST section of a sample data class ACS routine.

This routine uses the data categories from the tape mount management analysis to accomplish the following:

- Filter out data sets intended to be stored offsite.
- Filter out very large data sets intended to go directly to tape.
- Assign an appropriate data class name to data sets intended for tape mount management. These data sets are directed to the tape mount management DASD buffer.

---

```

//*****
//*          Start of FILTLIST Statements          */
//*****
FILTLIST BACKUP      INCLUDE('IEBGENER','COPY','IDCAM5')
FILTLIST DASD        INCLUDE('3380','3390','SYSDA','')
FILTLIST DATACOMP    INCLUDE(*LOG,'LOG')
FILTLIST DATAF      INCLUDE('DATA','FDATA','SYSGO','SYSLIN',OBJ*)
FILTLIST DATAV      INCLUDE('TEXT','VDATA')
FILTLIST DS_STACK    INCLUDE('STK=NSMS','STK=SMSD')
FILTLIST HFS         INCLUDE(HFS*)
FILTLIST HSM         INCLUDE(**.HMIGTAPE.DATASET,
                          **.BACKTAPE.DATASET,
                          **.DMP.*.V*.D*.T*,
                          **.COPY.BACKTAPE.*,
                          **.COPY.HMIGTAPE.**)
FILTLIST LARGE_BACKUP INCLUDE(PAYROLL.DFDSS.BACKUP*.*,
                              IMS*.IMS%TAPE.ICOPY*.*,)
FILTLIST LARGE_ACTIVE INCLUDE(PARTS.INVENTORY.MASTER*.*,)
FILTLIST LIST        INCLUDE('SYSOUT',LIST*)
FILTLIST LOAD        INCLUDE('RESLIB','PGMLIB',LOAD*)
FILTLIST OFFSITE     INCLUDE(**.INTER*.*,*.IFACE*.*,*.DR*.*,
                              **.VAULT*.*,*.OFFSITE*.*,
                              **.ARCHIVE*.*,*.ARC*.*,)
FILTLIST LARGE_TEMP  INCLUDE(SYS9*.T*.RA0000.PAYMTHLY.R*)
FILTLIST SPECIAL_USERS INCLUDE('SYSPROG','STGADMIN','DBA')
FILTLIST SRCF        INCLUDE(COB*,FOR*,CNTL*,JCL*)
                          EXCLUDE(**.SPFTEMP%.CNTL)
FILTLIST SRCV        INCLUDE('PLI','PL1','SCRIPT','CLIST')
FILTLIST TAPE        INCLUDE('3490','3480','3480X',TAPE*,'T3480','T3490',
                              '3494','3495')
FILTLIST VALID_DATA_CLASS INCLUDE('DATACOMP','DATAF','DATAV','DIRECT',
                              'ENTRY','GDGF80','GDGV104',
                              'KEYED','LINEAR','LISTING','LOADLIB',
                              'NONTMM','SRCFLIB','SRCVLIB','HFSDS',
                              'TMMACTV','TMMBKUP')
//*
//*****
//*          End of FILTLIST Statements          */
//*****

```

---

*Figure 98. FILTLIST Section of a Sample Data Class ACS Routine for Tape Mount Management*

Figure 99 on page 206 uses the filters previously defined in the data class routine to identify the classifications of tape data, and sets the appropriate artificial class. These classes are later used in the storage class, management class, and storage group ACS routines to determine if and how the data should be managed, and where it should be stored.



## Example: Data class ACS routine for tape mount

```
/******  
/*      Start of Tape Data Set Mainline      */  
/******  
WHEN (&UNIT = &TAPE or &UNIT=&DS_STACK)  
DO  
  SELECT                                /*Start of Tape Select  */  
  WHEN (&GROUP = &SPECIAL_USERS &&    /*Permit system pgms.  */  
        &DATACLAS = 'NONTMM')          /*and storage admin. to */  
  DO                                     /*write tape volumes    */  
    SET &DATACLAS = &DATACLAS  
    EXIT  
  END  
  WHEN (&DSN = &OFFSITE)                /*Write data sets to be */  
  DO                                     /*sent offsite to own   */  
    SET &DATACLAS = 'TAPOSITE'          /*data class           */  
    EXIT  
  END  
  WHEN (&DSN = &LARGE_BACKUP)           /*Write large data set  */  
  DO                                     /*backups to tape      */  
    SET &DATACLAS = 'TAPBKUP'  
    EXIT  
  END  
  WHEN (&DSN = &LARGE_ACTIVE)           /*Write other large,    */  
  DO                                     /*permanent data sets  */  
    SET &DATACLAS = 'TAPACTV'          /*to tape              */  
    EXIT  
  END  
  WHEN (&DSN = &LARGE_TEMP)             /*Write large, temporary */  
  DO                                     /*data sets to tape    */  
    SET &DATACLAS = 'TAPTEMP'  
    EXIT  
  END  
  WHEN (&DSN = &HSM)                    /*Write HSM ML2, dump,  */  
  DO                                     /*backup, and TAPECOPY  */  
    SET &DATACLAS = 'HSMDC'           /*data sets to tape    */  
    EXIT  
  END  
  WHEN (&DSTYPE = 'TEMP')               /*Identify temporary   */  
  DO                                     /*data sets that are   */  
    SET &DATACLAS = 'TMMTEMP'        /*TMM candidates      */  
    EXIT  
  END  
  WHEN (&PGM = &BACKUP)                 /*Set TMM backup       */  
  DO                                     /*data class           */  
    SET &DATACLAS = 'TMMBKUP'  
    EXIT  
  END  
  OTHERWISE  
  DO  
    SET &DATACLAS = 'TMMACTV'          /*Set TMM active data  */  
    EXIT                               /*class to all other data */  
  END                                  /*sets                 */  
  END                                  /*End of Tape DO       */  
  END                                  /*End of Tape SELECT   */  
/******  
/*      End of Tape Data Set Mainline      */  
/******
```

Figure 99. Sample Data Class ACS Routine for Tape Mount Sample Data Class ACS Routine for Tape Mount

## Creating the Storage Class ACS Routine

Figure 100 on page 207 shows the storage class ACS routine that assigns tape mount management candidates to the STANDARD storage class. All other tape data sets are assigned the null storage class, so they go to tape, as initially requested by the user. In this routine, artificial data classes are used to distinguish the tape mount management candidates from other tape data sets that should remain unmanaged.

## Example: Sample storage class ACS routine for tape mount management

```
/*
*****
*/
/*      Start of FILTLIST Statements      */
/*
*****
*/
FILLIST VALID_DEVICE      INCLUDE('3380','3390','3420',3480',3490',
                                'SYSDA',',',3480X',TAPE*,'3494',
                                '3495','9345')
FILLIST TMM_DATA_CLASS    INCLUDE('TMMACTV','TMMBKUP')
FILLIST TAPE_DATA_CLASS  INCLUDE('TAPACTV','TAPBKUP',TAPTEMP',
                                'TAPOSITE','NONTMM')
FILLIST VALID_STORAGE_CLASS INCLUDE('BACKUP','CRITICAL','DBCRT','FAST',
                                'FASTREAD','FASTWRIT','GSPACE',
                                'MEDIUM','NONVIO','STANDARD')
/*
*****
*/
/*      End of FILTLIST Statements      */
/*
*****
*/
SELECT
WHEN (&UNIT ^= &VALID_DEVICE && &UNIT ^= 'STK=SMSD')
    /* Unit must be valid DASD*/
    /* or tape device or not */
    DO
        SET &STORCLAS = ''
        /* externally specified */
        EXIT
    END
WHEN (&HLQ = &HSM_HLQ &&
      &DSN(2) = &HSM_2LQ)
    /* Do not manage data sets*/
    /* on ML1, ML2 */
    DO
        SET &STORCLAS = ''
        EXIT
    END
WHEN (&DATACLAS = &TAPE_DATA_CLASS)
    /* Do not manage "large" */
    /* or offsite tape data */
    /* sets */
    DO
        SET &STORCLAS = ''
        EXIT
    END
WHEN (&GROUP = &SPECIAL_USERS &&
      &STORCLAS = 'NONSMS')
    /* Permit storage admin. */
    /* or data base admin. */
    /* to create */
    /* non-system-managed */
    /* data sets */
    DO
        SET &STORCLAS = ''
        EXIT
    END
WHEN (&DATACLAS = &TMM_DATA_CLASS)
    /* Manage active, backup,*/
    /* temporary data sets */
    /* that are tape mount */
    /* management candidates */
    DO
        SET &STORCLAS = 'STANDARD'
        EXIT
    END
END
```

Figure 100. Sample Storage Class ACS Routine for Tape Mount Management

## Creating the Management Class ACS Routine

New data allocations are assigned a management class for tape mount management appropriate for the particular data type. This data migrates through the storage hierarchy in the same way as other DASD data sets.

### Example: Sample management class ACS routine for tape mount management

Figure 101 on page 208 shows a sample management class ACS routine.

---

```

/*****
/*      Start of FILTLIST Statements      */
/*****
FILTLIST ACTIVE          INCLUDE('TMMACTV')
FILTLIST BACKUP         INCLUDE('TMMBKUP')
FILTLIST HSMENV         INCLUDE('RECOVER','RECALL')
FILTLIST VALID_STORAGE_CLASS INCLUDE('BACKUP','CRITICAL','FAST','DBCRT',
                                'DBLOG','FASTREAD','FASTWRIT','GSPACE',
                                'MEDIUM','NONVIO','STANDARD')
FILTLIST VALID_MGMT_CLASS INCLUDE('DBML2','DBSTAN','EXTBAK',
                                'GDGBKUP','GDGPROD','INTERIM',
                                'NOACT','STANDARD','TMMACTV',
                                'TMMBKUP')
/*****
/*      End of FILTLIST Statements      */
/*****
/*****
/*      Start of Mainline SELECT      */
/*****
SELECT
  WHEN (&ACSENVIR = &HSMENV &&          /* Handle RECALLED TMM   */
        &MGMTCLAS = 'TMMACTV')         /* data sets as any other */
    DO                                  /* data set              */
      SET &MGMTCLAS = 'STANDARD'
      EXIT
    END
  WHEN (&ACSENVIR = &HSMENV &&          /* Handle RECALLED TMM   */
        &MGMTCLAS = 'TMMBKUP')         /* data sets as a        */
    DO                                  /* standard GDS          */
      SET &MGMTCLAS = 'GDGBKUP'
      EXIT
    END
  WHEN (&DATACLAS = &ACTIVE)           /* Let 'all other' TMM   */
    DO                                  /* data sets migrate thru */
      SET &MGMTCLAS = 'TMMACTV'         /* storage hierarchy     */
      EXIT
    END
  WHEN (&DATACLAS = &BACKUP)           /* Let 'point in time'   */
    DO                                  /* backup TMM data sets  */
      SET &MGMTCLAS = 'TMMBKUP'         /* go directly to ML2    */
      EXIT
    END
/*****
/*      End of Mainline SELECT      */
/*****
END                                  /* End of Management Class Procedure */

```

---

Figure 101. Sample Management Class ACS Routine for Tape Mount Management

**Related Reading:** To enable setting of management class names and storage group names, DFSMSrmm calls the management class ACS routine for non-SMS tape data sets. See *z/OS DFSMSrmm Implementation and Customization Guide* for further information on the variables set for the RMMVRS environment.

## Creating the Storage Group ACS Routine

When data sets are RECALLED from the DASD buffer, they are directed to the PRIME<sub>xx</sub> or LARGE<sub>xx</sub> storage group, based on data set size. They are not redirected to the DASD buffer. New data allocations are directed to the TMMBUF<sub>xx</sub> or TMMBFS<sub>xx</sub> storage group.

### Example: Sample storage group ACS routine for tape mount management

Figure 102 on page 209 shows a sample storage group ACS routine.

---

```

/*****
/* Start of FILTLIST Statements */
/*****
FILTLIST SPECIAL_USERS INCLUDE('SYSPROG','STGADMIN','DBA')
/*****
/* End of FILTLIST Statements */
/*****
/*****
/* Start of Mainline SELECT */
/*****
SELECT
  WHEN (&DATACLAS = 'TMMTEMP') /* Temporary tape data sets*/
  DO /* go to the LARGE group */
    SET &STORGRP = 'LARGE90','LARGE80'
    EXIT
  END
  WHEN (&MGMTCLAS = TMM*) /* Other tape data sets */
  DO /* selected for TMM go */
    SET &STORGRP = 'TMMBUF90', /* to TMM buffer; BFS */
              'TMMBUF80', /* is a "spill" area only */
              'TMMBFS90', /* used if BUF is full */
              'TMMBFS80'
    EXIT
  END
/*****
/* End of Mainline SELECT */
/*****

```

---

Figure 102. Sample Storage Group ACS Routine for Tape Mount Management

If you choose, DFSMSrmm can call storage group ACS routines for non-system-managed tapes to obtain a storage group name and use it as a scratch pool ID. For information on how to choose this option, see *z/OS DFSMSrmm Implementation and Customization Guide*.

---

## Using Tape Mount Management Techniques

Tape mount management includes the following techniques:

- “Data Set Stacking”
- “Unit Affinity” on page 213
- “Using Volume Reference to System-Managed Data Sets” on page 215
- “Using Volume Reference to Non-System-Managed Data Sets” on page 216
- “Volume Reference Chains” on page 218

### Data Set Stacking

*Data set stacking* places several data sets on the same tape volume or set of tape volumes. It is used to accomplish the following:

- Increase efficiency of tape media usage
- Reduce the overall number of tape volumes needed when allocating new data sets

To request data set stacking, you must have the following JCL options on the DD statement:

- Data-set-sequence-number subparameter on the LABEL parameter is greater than one.

This subparameter is used to identify the relative position of a data set on a tape volume. For existing cataloged data sets, the system obtains the data-set-sequence-number from the catalog.

- VOL=SER or VOL=REF parameter

This parameter indicates the tape volume or volumes on which to stack. When VOL=SER is used, at least one of the volume serial numbers specified must match one of the volume serial numbers for the data set on which this data set is being stacked (either explicitly specified or obtained from the catalog). Use VOL=SER when stacking multi-volume, multi-data sets within the same job step.

**Related Reading:** For more recommendations on when to use VOL=REF versus VOL=SER, see *z/OS MVS JCL User's Guide*.

The following example shows data set stacking using VOL=SER.

```
//P123456 JOB
//STEP1 EXEC
//D1 DD DSN=A,DISP=(NEW,CATLG),VOL=SER=(VOL1,VOL2),
// UNIT=TAPE,LABEL=(1,SL)
//D2 DD DSN=B,DISP=(NEW,CATLG),VOL=SER=VOL2,
// UNIT=TAPE,LABEL=(2,SL)
//
```

## Data Set Collection

A *data set collection* is the collection of data sets that are intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking.

**Requirement:** All data sets in the data set collection *must* be directed to the same device category, of the following four device categories:

- System-managed DASD
- System-managed tape
- Non-system-managed DASD
- Non-system-managed tape

A mixture of these device categories is not allowed for the following reasons:

- There would be problems accessing the data sets later. For example, a data set with a data-set-sequence-number of three could be placed as the first data set on the tape if the first two data sets were redirected to DASD.
- There could be problems locating data sets later since some types of data sets must be cataloged and others may be uncataloged.

If you have an allocation in a previous job or step that specifies VOL=SER and that is rerouted to SMS DASD, it will be cataloged on a volume other than that specified in the JCL.

For example, in the following statement, you must not specify volume VOL001 in the JCL unless the volume is in a DUMMY storage group if you later want to allocate the data set as OLD:

```
//DD1 DD DSN=A,VOL=SER=VOL001,DISP=(NEW,CATLG),LABEL=(1,SL)
```

Otherwise, it will cause the locate for the data set to be bypassed, probably resulting in an OPEN abend since the data set was rerouted and is not on the volume.

As a result, a statement such as the following is not recognized as data set stacking:

```
//DD1 DD DSN=A,DISP=OLD
//DD2 DD DSN=B,VOL=SER=VOL001,DISP=(NEW,CATLG),LABEL=(2,SL)
```

This is because the system does not include existing SMS-managed requests in any data set collection.

When you analyze your current tape usage, you might determine that one of the tape data sets in a data collection would be a good candidate to redirect to DASD. The system attempts to ensure that all data sets in a data set collection are directed to the same device category.

### Data Set Stacking Using Volume Reference

When data set stacking is requested with the VOL=REF parameter, the ACS routines are passed information that indicates that volume reference is used. Therefore, the ACS routines can direct the requests within a data collection to the same device category.

Additional information is now passed to the ACS routines when VOL=REF is used. The &ALLVOL and &ANYVOL ACS read-only variables contain one of the following values when the reference is to a system-managed data set:

- 'REF=SD' - The reference is to an SMS-managed DASD or VIO data set
- 'REF=ST' - The reference is to an SMS-managed tape data set
- 'REF=NS' - The reference is to a non-SMS-managed data set

When a data set is referenced, the name of the referenced storage group is passed to the storage group ACS routine in the &STORGRP read-write variable. The ACS routines can allow the allocation in the storage group of the referenced data set or select a different storage group or list of storage groups. For NEW to NEW references, multiple storage groups might have been assigned to the referenced data set. In this case, only the first storage group is passed as input to the ACS routines for the referencing data set; this might not be the storage group in which the referenced data set is actually located.

Figure 103 shows an example of an ACS routine fragment to assign the referencing data set to the same storage group as the referenced data set.

```
PROC STORGRP
FILTLIST REF_SMS INCLUDE('REF=SD','REF=ST')

IF &ANYVOL = &REF_SMS THEN
  IF &STORGRP = '' THEN
    SET &STORGRP = &STORGRP
  ELSE
    IF &ANYVOL = 'REF=SD' THEN
      SET &STORGRP = 'POOLSG'
    ELSE
      SELECT(&LIBNAME)
        WHEN('ATL1')
          SET &STRGRP = 'TAPESG1'
        WHEN('ATL2')
          SET &STRGRP = 'TAPESG2'
    END
  END
END
```

Figure 103. Sample ACS Routine to Assign Same Storage Group as Referenced Data Set

**Rule:** The assignment of &STORGRP = &STORGRP does not work if you have entered private tapes into the library with a blank storage group name, since a valid storage group name is unavailable.

## Data Set Stacking Using Volume Serial

With the enhancements, the system is now aware of data set stacking when requested with volume serial (either with the VOL=SER parameter or obtained from the catalog) within a job step.

**Recommendation:** Use VOL=REF for data set stacking across jobs or steps. If you use VOL=SER to stack data sets across steps or jobs, the system cannot detect data set stacking. In these cases, you can do one of the following:

- Change your JCL to specify VOL=REF instead of VOL=SER.
- Ensure that your ACS routines don't redirect these data set allocations, unless you guarantee that they can be redirected to a consistent device category.

If the ACS routines initially directed the stacked allocations to different device categories, the system detects this and re-invokes the ACS routines, passing them additional information. The ACS routines can then take one of the following actions:

- Correct the problem and route the allocations to consistent device categories
- Fail the stacked allocation (if the ACS routine exits with a non-zero return code)
- Fail to correct the inconsistency; SMS fails the allocation

Figure 104 and Figure 105 show examples of ACS routine fragments to assign the referencing data set to a consistent device category as the referenced data set.

```
PROC STORCLAS
FILTLIST DS_STACK INCLUDE('STK=SMSD','STK=NSMS')

IF &UNIT = &DS_STACK THEN
  SELECT (&UNIT)
    WHEN('STK=SMSD')
      SET &STORCLAS = 'DASDSC'
    OTHERWISE
      SET &STORCLAS = '
END
:
END
```

Figure 104. Storage Class ACS Routine Fragment to Assign Consistent Device Category

```
PROC STORGRP
FILTLIST DS_STACK INCLUDE('STK=SMSD','STK=NSMS')

IF &UNIT = &DS_STACK THEN
  SELECT (&UNIT)
    WHEN('STK=SMSD')
      SET &STORGRP = 'MAIN3380','MAIN3390','SPIL3380','SPIL3390'
    OTHERWISE
      :
  END
:
END
```

Figure 105. Storage Group ACS Routine Fragment to Assign Consistent Device Category

The system re-invokes the ACS routines only when all of the following conditions are true:

- The request is part of a data set collection based on:
  - A data-set-sequence-number greater than one specified on the LABEL parameter
  - VOL=SER, either specified or obtained from the catalog, where at least one of the volume serial numbers matches one of the volume serial numbers for a previous request in the *same step*.

**Recommendation:** Because data set stacking cannot be detected across jobs and steps when VOL=SER is used, the ACS routines are not reinvoked in these cases. Use VOL=REF instead, or ensure that your ACS routines don't redirect those data set allocations requesting data set stacking using VOL=SER across steps or jobs.
- The request is currently directed to a different device category than the other requests in the data set collection.
- The request is DISP=NEW (or DISP=MOD treated as NEW).

Since data set stacking might cause a second or third invocation of the ACS routines, you might want to take special care when using WRITE statements to avoid duplicates in the job log.

Additionally, existing SMS-managed data sets are not checked by the system for inclusion in a data set collection. For SMS-managed data sets that are cataloged, the system cannot assume that the volume information in the catalog represents the original intended volume for the data set. For example, the data set might have been redirected, in which case if the system uses the volume information in the catalog to allocate the data set, the data set might be incorrectly placed in the wrong data set collection.

## Unit Affinity

Data set stacking can be used in conjunction with unit affinity. In a tape environment, unit affinity is a JCL keyword (UNIT=AFF) used to minimize the number of tape drives used in a job step. The system attempts to use the same tape drive for a request that specifies UNIT=AFF for both the referenced and referencing DD statements.

Additional information is now passed in the &UNIT ACS read-only variable so that the ACS routines will know when unit affinity was requested and whether or not data set stacking was used. With the increased awareness of data set stacking, the ACS routines are invoked another time if inconsistent device categories were detected.

Table 38 shows the values to which the &UNIT read-only variable are set when UNIT=AFF is requested, as well as what each value means to the ACS routines:

*Table 38. Values for &UNIT ACS Read-Only Variable*

&UNIT Value	ACS Invocation	Data Set Stacking Indication	Device Category of Data Set on Which to Stack
AFF=	First	Unknown	Not applicable
STK=SMSD	Second	Yes and different device categories	System-managed DASD



Table 38. Values for &UNIT ACS Read-Only Variable (continued)

&UNIT Value	ACS Invocation	Data Set Stacking Indication	Device Category of Data Set on Which to Stack
STK=NSMS	Second	Yes and different device categories	Non-system-managed DASD or Non-system-managed Tape
STK=SMSD or STK=NSMS	Third	Yes and different device categories	Non-system-managed DASD or Non-system-managed Tape
<b>Note:</b> ACS routines can be invoked three times in a JES3 environment.			

*z/OS MVS JCL User's Guide* discusses data set stacking and unit affinity and provides examples.

## Non-Data Set Stacking Allocations

The following considerations apply when UNIT=AFF' is used to reduce the number of units for a job, instead of data set stacking.

When unit affinity is specified on a DD statement, three new values are set depending on the unit of the AFF'ed DD. The following example explains how these values are set. In this example, DD1 is directed to SMS DASD, and DD2 is directed to SMS tape.

```
//DD1 DD UNIT=SYSDA,DISP=NEW,..
//DD2 DD UNIT=AFF=DD1,DISP=NEW,..
//DD3 DD UNIT=AFF=DD2,DISP=NEW,..
//DD4 DD UNIT=AFF=DD1,DISP=NEW,..
```

DD Being Processed	&UNIT Read-Only Variable Value
DD1	'SYSDA'
DD2	'AFF=SMSD' (if DD1 is directed to SMS DASD)
DD3	'AFF=SMST' (if DD2 is directed to SMS tape)
DD4	'AFF=SMSD' (if DD1 is directed to SMS DASD)

With the exception of the JES3 environment, ACS routines are called multiple times. When the ACS routines are invoked by the JES3 PRESCAN processing, the &UNIT read-only variable is set to 'AFF=' for DD2, DD3, and DD4. The ACS routines are invoked again later during the allocation process with the values shown in the example.

Table 39 illustrates how &LIBNAME, &STORGRP, &ALLVOL, &ANYVOL, and &STORCLAS are set. These read-only variables are set depending on the value of the AFF'ed DD (for example, DD1 if DD2 is being processed) or the VOLSER value on the AFF'ing DD (DD2).

Table 39. AFFing DD Volser Values

AFFing DD Volser Values		&LIBNAME	&STORGRP	&ALLVOL	&ANYVOL	&STORCLAS
DD2 volser= V2	V2 = lib resident	libnamesofV2	sgnameofV2	V2	V2	blank
	V2 ≠ lib resident	blank	blank	V2	V2	blank

Table 39. AFFing DD Volser Values (continued)

AFFing DD Volser Values		&LIBNAME	&STORGRP	&ALLVOL	&ANYVOL	&STORCLAS	
DD2 volser= blank	DD1= SMSDASD	blank	sgnameofDD1	blank	blank	scnameofDD1	
	DD1= nonSMS	blank	blank	blank	blank	blank	
	DD1= SMSTAPE	DD1volser =V1	libnameofV1	sgnameofV1	blank	blank	blank
		DD1volser =blank DISP=NEW	blank	sgnameofDD1	blank	blank	blank

## Using Volume Reference to System-Managed Data Sets

For SMS DASD, you can assign different storage groups to data sets when the VOL=REF parameter is specified in the JCL. Therefore, you can use overflow storage groups (also known as "spill" storage groups) with data sets when VOL=REF is specified. If a data set that uses VOL=REF is redirected to the tape mount management DASD buffer and its allocation exceeds the available space, it spills into the overflow storage group.

You can designate an overflow storage group using the ISMF Pool Storage Group Define panel.

**Note:** When an overflow storage group contains more volumes than a buffer storage group, specified volume counts might result in volumes in the overflow storage group being preferred over volumes in the buffer storage group during volume selection.

For SMS mountable tape, the referencing data must be assigned to the same storage group as the referenced data set, if the referencing data set is also to be SMS mountable tape data set.

For example, consider two storage groups, BIG and SMALL, that are defined based on data set size. If the referenced data set is assigned to storage group BIG, you must also ensure that the referencing data set goes to storage group BIG, even if its size would logically assign it to storage group SMALL. Conversely, if the referenced data set is assigned to storage group SMALL, then the referencing data set must also be assigned to storage group SMALL. If the referencing data set is large, this can result in out-of-space abends for allocations in storage group SMALL.

Consider the following rules for coding ACS routines:

- If the reference is to a data set on one or more system-managed tape volumes, then the two data sets must be assigned to the *same* storage group.

When you use the VOL=REF parameter with tape, this indicates that the two data sets should be allocated so they have at least one volume in common (or share the same volume). If the tape volume is system-managed and the two data sets must share the tape volume, then the data sets must also be allocated in the same storage group. A single tape volume cannot reside in more than one storage group.

- If the reference is to a data set on system-managed media other than tape, the two data sets must be assigned to compatible *types* of storage groups.

The two data sets must be assigned to compatible *types* of storage groups to ensure consistency for locates. For example, if the referenced data set is allocated on DASD, then allocating referencing data set on tape could result in potential locate problems.

The following storage group types cannot be mixed with TAPE storage groups:

- POOL (DASD volumes)
- VIO (paging space)

**Restriction:** If you use VOL=REF processing to refer to a temporary data set, you might get different results in storage group assignments than expected. This is because temporary data sets are assigned a storage group by the system, based on a list of eligible storage groups, such as: VIO, PRIME, STANDARD, etc. Data sets that use VOL=REF are assigned a storage group based on this list of eligible storage groups, not on the name of the storage group used to successfully allocate the first data set being referenced. This might result in the data sets being allocated in different storage groups.

Information on the referenced device type and the referenced storage group can be passed to the ACS routines when VOL=REF is used. The &ALLVOL and &ANYVOL ACS read-only variables contain one of the following values when the reference is to a system-managed data set:

- 'REF=SD' - The reference is to an SMS-managed DASD or VIO data set
- 'REF=ST' - The reference is to an SMS-managed tape data set

The &STORGRP read-write variable contains the name of the referenced storage group when the ACS routine is entered. You can then allow the allocation in the storage group of the referenced data set or select a different storage group or list of storage groups.

Figure 106 shows an example of how to code your ACS routine to assign the referencing data set to a *different* storage group than the referenced data set.

```
PROC STORGRP
SELECT (&ANYVOL)
WHEN('REF=SD')
  IF &DSTYPE = 'TEMP' && &DSORG NE 'VS' THEN
    SET &STORGRP = 'VIOG', 'MAIN3380', 'MAIN3390', 'SPIL3380', 'SPIL3390'
  ELSE
    SET &STORGRP = 'MAIN3380', 'MAIN3390', 'SPIL3380', 'SPIL3390'
WHEN('REF=ST')
  IF &STORGRP = '' THEN
    SET &STORGRP = &STORGRP
  ELSE
    SELECT(&LIBNAME)
    WHEN('ATL1')
      SET &STORGRP = 'TAPESG1'
    WHEN('ATL2')
      SET &STORGRP = 'TAPESG2'
    END
  OTHERWISE
END
END
```

Figure 106. Sample ACS Routine to Assign Different Storage Group than Referenced

## Using Volume Reference to Non-System-Managed Data Sets

When a volume reference is made to a non-system-managed data set, control is passed to the ACS routines and information is provided to the ACS routines that the referenced data set is non-system-managed. The &ALLVOL and &ANYVOL

ACS read-only variables contain the following value when the reference is to a non-system-managed data set: 'REF=NS'. This means the reference is to a non-system-managed data set.

You can code your ACS routines to do one of two of the following actions:

- Allow the allocation to proceed as a non-system-managed data set.  
Figure 107 shows an example of how to code your ACS routine to allow those that meet certain criteria to proceed and warn others that the allocations will be failed after a specific date.

```
PROC 0 STORGRP
FILTLIST AUTH_USER INCLUDE('SYSPROG1','SYSPROG2','STGADMIN','SYSADMIN')

IF &ANYVOL = 'REF=NS' THEN
  IF &USER NE &AUTH_USER THEN
    DO
      WRITE 'INVALID USE OF VOL=REF TO A NON-SYSTEM-MANAGED DATA SET'
      WRITE 'AS OF 12/31/04 ALLOCATION WILL BE FAILED FOR ' &DSN
    END
  :
END
```

Figure 107. Sample ACS Routine to Allow Allocation of Non-System-Managed Data Set

- Fail the allocation by exiting with a non-zero return code.  
Figure 108 shows an example of how to code your ACS routine to fail the invalid uses of VOL=REF after the specified date.

```
PROC 1 STORGRP
FILTLIST AUTH_USER INCLUDE('SYSPROG1','SYSPROG2','STGADMIN','SYSADMIN')

IF &ANYVOL = 'REF=NS' THEN
  IF &USER NE &AUTH_USER THEN
    DO
      WRITE 'INVALID USE OF VOL=REF TO A NON-SYSTEM-MANAGED DATA SET'
      WRITE 'DATA SET ' &DSN ' MUST BE SYSTEM-MANAGED'
      EXIT CODE(4)
    END
  :
END
```

Figure 108. Sample ACS Routine to Fail Allocation of Non-System-Managed Data Set

If the ACS routines attempt to make the referencing data set SMS-managed, SMS fails the allocation. This is because allowing the data set to be allocated as an SMS-managed data set could cause problems locating the data set, as well as potential data integrity problems. These problems occur when the data set is allocated with DISP=OLD or DISP=SHR, due to the tendency to copy old JCL. For example, the following sample referenced data set is non-SMS-managed:

```
//DD1 DD DSN=A.B.C,DISP=OLD,VOL=REF=D.E.F,....
```

If the referenced data set is SMS-managed, a LOCATE is done for the referencing data set, and the data set is accessed using the result of the LOCATE. If the referenced data set is non-SMS-managed, then no LOCATE is done, and the referencing data set is assumed to reside on the same volume as the referenced

data set. If the referencing data set is in fact non-SMS-managed, then performing a LOCATE could result in finding another data set of the same name, causing a potential integrity problem. If the referencing data set was migrated to SMS, then not performing a LOCATE would cause the data set not to be found.

## Volume Reference Chains

A volume reference chain is a group of DD statements in which a subsequent DD statement uses the VOL=REF parameter to refer to a previous DD statement that also uses the VOL=REF parameter to refer to another preceding DD statement.

The following is an example of volume reference chaining:

```
//DD1 DD DSN=A,DISP=OLD  
//DD2 DD DSN=B,DISP=(NEW,CATLG),VOL=REF=*.DD1 NEW to OLD reference  
//DD3 DD DSN=C,DISP=(NEW,CATLG),VOL=REF=*.DD2 NEW to NEW reference
```

---

*Figure 109. Example of Volume Reference Chaining*

In a tape mount management environment, you can determine that any of the data sets is a good candidate to redirect to the DASD buffer.

---

## Tuning DFSMSHsm Operation

After you develop your SMS configuration and tape mount management is working, you can tune the DFSMSHsm parameters to optimize performance.

You can establish up to 15 concurrent interval migration tasks to migrate data from primary volumes to tape. You can improve the effective data rate up to three times by increasing the number of tasks from one to seven. The SETSYS MAXINTERVALTASKS setting controls the maximum number of these tasks that can operate concurrently. Ensure that one cartridge drive per interval migration task is available to support this multi-tasking.

---

## Chapter 12. Managing Tape Volumes

This chapter describes migrating your tape volumes to system management using:

- Automated Tape Libraries
- Manual Tape Libraries

Most of the information in this chapter applies to both of these kinds of libraries. Information that applies to only one is identified as such.

In Chapter 11, “Optimizing Tape Usage,” on page 177, you used tape mount management to direct some of your tape data sets to two storage groups, TMMBUFxx and TMMBFSxx, to have the data sets managed by DFSMSHsm. They migrate through the storage hierarchy and eventually reside on migration level 2 volumes. Migration level 2 volumes, and the volumes containing data sets written by DFSMSHsm or your own applications directly to tape, can be system-managed. The addition of system-managed tape enables you to manage all types of storage media-DASD, optical disks, and tape volumes-in the DFSMS environment.

---

### Understanding the Benefits of System-Managed Tape

This section describes some of the benefits of placing your tape volumes under system management.

#### Using Automated Tape Libraries

Tape volumes can be migrated to system management by installing the IBM TotalStorage Enterprise Automated Tape Library (3494 or 3495), including the Virtual Tape Server, or by using the manual tape library.

- You can automate mounting and demounting of tape volumes using the IBM TotalStorage Enterprise Automated Tape Library (3494 or 3495).
- With system-managed tape, through data class, you can easily specify a media type or recording technology preference and whether compaction should be used. This enables you to direct allocations to a subset of devices and media.

**Related Reading:** For more information about media types and recording technology, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

- You can set up an offsite tape library dataserver.

An IBM TotalStorage Enterprise Automated Tape Library can be serially attached to processors up to 14 miles (23 kilometers) away. Tapes created during backup processes can then be written to an offsite tape library dataserver and used to rebuild your data at a recovery site.

You need to define a *tape library* to support system-managed tape data. A set of integrated catalog facility user catalogs, called the *tape configuration database*, contains information about your tape libraries and volumes. You can use tape storage groups to direct new allocation requests to tape libraries.

The tape library definitions are created using ISMF. This builds a library record in the tape configuration database and in the specified SCDS. A tape library contains a set of volumes and the tape subsystems that are used to support mount activity for the library's volumes. A tape library supports both scratch and private volumes.

The tape library definition includes the environmental characteristics for a set of tape drives and the volumes that are mounted on them. Characteristics include scratch thresholds, default data class, and media information.

The tape configuration database consists of the tape library and volume records residing in one or more tape library volume catalogs. Volume catalogs are integrated catalog facility user catalogs containing system-managed tape library and volume entries. A general tape library catalog contains all library records. If specialized catalogs do not exist, volume entries are placed in this catalog. You can create specialized catalogs, selected based on the first character of the volume serial number, to hold data about related tape volumes.

## Using Manual Tape Libraries

The manual tape library extends the scope of system-managed storage to tape devices and their associated volumes outside of the ATLDs environment. It enables the definition of such tape devices as belonging to logical libraries, much as they are defined in an automated tape library. By means of this support, drives and media are associated with a library so that allocation of drives and their appropriate media are properly managed.

Unlike the ATLDs, the manual tape library does not use the library manager. With the manual tape library, a human operator responds to mount messages generated by the host and displayed on a console. There are no robotics associated with an MTL, it is a purely logical grouping of non-ATLDs drives — standalone drives — that is supported with MTL. Mount messages are displayed on consoles for human operators to see and respond to, just as they would for standalone, non-ATLDs tape drives.

Volumes can be associated with manual tape libraries so that only those volumes defined for a specific manual tape library can be mounted on drives in that MTL. See “Setting Up a Manual Tape Library” on page 229 for information about defining the manual tape library.

**Exception:** Manual tape libraries are not intended to operate within competitive robotic tape libraries. If you need to try such a definition, contact the manufacturer of the specific robotic library system for assistance.

Because media associated with any new tape devices will likely be incompatible with the real devices that are being emulated, there is a need to take this management out of the hands of the user and into system management. The manual tape library provides this ability by recognizing the real underlying device type rather than the device type that is being emulated. By defining these libraries, associating the media with these libraries and properly defining the SMS constructs, the allocation of drives and the mounting of the appropriate media can be accomplished through system control.

No JCL changes are required to use MTL. The SMS storage group ACS routines can be updated to influence the placement of new tape data sets to an MTL. However, you must use HCD to identify tape drives as being MTL resident.

## Using Tape Storage Groups

You do not define tape volumes in tape storage groups as you do for DASD. Volumes are added to storage groups on use at open. Tape storage group definitions do not contain tape drives; they have privately owned tape volumes only. Multiple tape libraries can belong to a given tape storage group.



You can write the data class ACS routine to select the media interchange characteristics for the tape data set. If it is a tape mount management candidate, it can be assigned the TMMACTV, TMMBKUP, or TMMTEMP data class, and directed to the system-managed DASD buffer. If the data class ACS routine determines that the data set should be directly written to tape, based on the type of data written or the size of the data set, the data set is assigned the TAPOSITE, TAPTEMP, TAPACTV, or TAPBKUP data class. These data sets are written on system-managed volumes in the following way:

- The storage class ACS routine must assign a storage class for the request to be SMS managed.
- The storage group ACS routine assigns a tape storage group to the new tape allocation directly, and to a system-managed tape library and tape device pool indirectly.

When the storage group ACS routine selects a tape storage group:

- A list of tape device pools for the tape libraries belonging to the storage group is built.

A device pool is a string of tape drives that is part of a system-managed tape library.

- The preferred tape device pools belong to tape libraries that are above their scratch volume threshold.
- This ordered list of tape device pools is used to select the tape drive.

For a scratch volume, drives with active cartridge loaders containing the appropriate media type will be given preference. Tape management based scratch pools can be used with manual tape libraries and may restrict which volumes are allowed in response to the scratch request. DFSMSrmm pooling can be based on storage group assignments.

Note that when you use DFSMSHsm, **you must use** global scratch pools for storage group containing multiple automatic tape libraries. In this scenario, the tape device is selected first, followed by a tape. The tape and device must be in the same library, so using a private (HSM) scratch pool can result in running out of empty tapes for the tape device that was allocated while empty tapes exists for other tape libraries in the storage group.

- Once the data set is opened, the volume record in the tape configuration database is updated with the storage group assigned to the request. If you specify an expiration date, it is also stored in the volume entry. Lastly, recording technology, compaction, and media type are updated.
- If the user requests to catalog the data set, unlike SMS DASD, it is cataloged at disposition time, rather than allocation time.

#### **Related Reading:**

- For specific RMM support, see *z/OS DFSMSrmm Implementation and Customization Guide*.
- For information on the use of the Initial Access Response Time (IART) value in the Virtual Tape Server Environment, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

For non-system-managed tape, you can use the SMS ACS routines to determine the scratch pooling on tape storage group names. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information.



## Using DFSMSrmm

DFSMSrmm is a full-function tape management system that supports your current non-system-managed volumes, as well as your system-managed volumes in tape library dataservers. DFSMSrmm supports the implementation of both automated and manual tape libraries:

- DFSMSrmm and DFSMSdftp share volume information.

Information for cartridges entered in the tape library datserver automatically updates the DFSMSrmm control data set. If the volume is defined to DFSMSrmm before it enters the tape library datserver, DFSMSrmm informs DFSMSdftp of the volume status in the DFSMSrmm control data set to update the tape configuration database.

DFSMSrmm can also drive Manual Cartridge Entry processing, which is another way of sharing information. In this case, using DFSMSrmm information to drive updates to the TCDB for system-managed manual libraries.

- DFSMSrmm works with DFSMSdftp to manage volume scratch status.

If all data sets on a tape volume have expired, DFSMSrmm calls OAM to update the volume's status and returns the volume to scratch status. DFSMSrmm also calls OAM to update the volume's status if a volume is reclaimed from scratch to private status.

- DFSMSrmm supports OAM's use of scratch thresholds defined for system-managed tape libraries.

OAM creates a write-to-operator (WTO) message when the tape library gets below the scratch volume threshold. This message is handled directly by DFSMSrmm to start the procedure to return eligible volumes to scratch status, or it can also be intercepted by NetView<sup>®</sup> (or an equivalent product) and drive submission of an equivalent job.

- DFSMSrmm can be used to request that OAM eject cartridges from the tape library datserver to move volumes offsite for disaster recovery.

---

## Planning Your Migration to System-Managed Tape

This section describes some of the tasks you should consider as you plan your migration of tape volumes to system management.

### Organizing Your Migration Tasks

The major tasks for moving tape to system-management are:

- Define the tape environment to z/OS.
- Define the OAM subsystem
- Define the storage classes and ACS routine
- Define the storage groups and ACS routine
- Create the tape configuration database
- Define the tape libraries
- Translate and validate the new SMS configuration
- If required, create any system-managed tape exits
- Test the new SMS configuration
- Activate the new SMS configuration
- Start the OAM subsystem
- Enter volumes in the tape library
- Test usage of the library for SMS tape allocations
- Put the tape library datserver into production

**Related Reading:** For a list of the tape exits available for DFSMS, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

These are the requirements for UNIT parameter usage for system-managed tape:

- The UNIT keyword, if externally specified, is available to your ACS routines for tape allocations, as it is for allocation of DASD data sets; however, it is not used by DFSMS to select a tape device. The device pools linked to the selected tape libraries form the candidate list of devices used to select a drive that can satisfy the allocation request.
- Devices requested using unit affinity, such as UNIT=AFF=DD1, are only honored if the volumes reside in the same tape library and use compatible devices.
- Except for demand allocation, JCL changes are not required to use a tape library dataserwer.

## Categorizing Tape Volumes for Migration

To place tape data under system management, subdivide the tape data into categories and migrate it by category. Following is a suggested order for this migration:

1. Large temporary data sets (&DATACLAS=TAPTEMP)
2. DFSMSHsm-owned volumes
3. Offsite volumes (&DATACLAS=TAPOSITE)
4. Active volumes (&DATACLAS=TAPACTV)
5. Backup volumes (&DATACLAS=TAPBKUP)

The storage class ACS routine must assign a storage class for the request to be SMS managed. For information about the use of storage class values, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

During the migration, if existing multivolume data sets are entered in libraries, ensure that all volumes for a multivolume tape data set reside in the same tape library.

OAM automatically updates your tape configuration database as you enter the cartridges into the library. OAM uses the information passed by DFSMSrmm (such as private or scratch, 18-track or 36-track recording). Make sure that the following DFSMSrmm storage location names are not used as tape library names: REMOTE, LOCAL, DISTANT, BOTH, CURRENT.

---

## Placing Tape Volumes under System Management

This section describes the tasks involved in placing your tape volumes under system management.

### Defining the Tape Environment

The Hardware Configuration Definition (HCD) is used to define a tape drive as library resident. The library ID defined to the ISMF Library Application, and to HCD, links the system-managed tape library definition with the tape library. Additionally, for an automated tape library dataserwer, this library ID is also defined at the library by the customer engineer.

### Defining OAM

OAM plays a central role in the SMS tape library support. OAM manages, maintains, and verifies the tape volumes and tape libraries within a tape storage environment.

**Related Reading:** For a complete discussion of OAM's role in the tape library support and installation requirements, and for instructions on how to tailor and start OAM, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

## Defining the Storage Classes

Storage Class ACS routines must assign a class to direct the tape allocation to SMS. See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information about the use of storage class values.

## Writing the Storage Class ACS Routine

You can use the data classes established during your implementation of tape mount management to control the categories of tape data sets that are allocated on system-managed tape volumes. The tape data classes named TAPxxxx define the data sets that are inhibited from tape mount management. You can place these tape volumes under system management by tailoring the filter lists developed for these categories so that more of these data sets are system-managed. Figure 110 shows the processing for tape data sets that are not system-managed. The storage class ACS routine must be changed to assign a storage class to DFSMSHsm-owned volumes and tape data sets. One way to do this is to eliminate the statements that assign the null storage class. This results in the STANDARD storage class being assigned by the OTHERWISE statement at the end of the storage class ACS routine.

---

```
WHEN (&HLQ = &HSM_HLQ &&                               /* Do not manage data sets*/
      &DSN(2) = &HSM_2LQ)                               /* on ML1, ML2          */
  DO
    SET &STORCLAS = ''
    EXIT
  END
WHEN (&DATACLAS = &TAPE_DATA_CLASS)                   /* Do not manage "large" */
  DO                                                    /* or offsite tape data  */
    SET &STORCLAS = ''                                 /* sets                  */
    EXIT
  END

OTHERWISE
  DO
    SET &STORCLAS = 'STANDARD'
    EXIT
  END
END

END /* END OF STORAGE CLASS ROUTINE PROC */
```

---

Figure 110. Storage Class Routine Fragment for Tape Data

## Designing the Physical Storage Environment

The TAPE storage group type links tape libraries to tape storage groups. A scratch tape volume becomes system-managed when a system-managed data set is allocated on the volume. The tape storage group assigned by your storage group ACS routine for the allocation is stored in the tape configuration database on the volume record.

The tape storage group definition links a storage group with tape libraries. Figure 111 on page 225 shows the Storage Group Application definition for the

HSMLOC tape storage group. It consists of DFSMSHsm migration, backup, and dump volumes. This storage group is linked to a local 3495 tape library called TAPELOC. Similarly, another storage group can be defined, called HSMREM and linked to TAPEREM, an offsite tape library dataserer containing tape devices that are eligible to satisfy the tape volume allocation request. This way, you can direct disaster recovery tapes to a remote library (TAPEREM), and all others to a local library (TAPELOC).

```

Panel  Utilities  Help
-----
                                TAPE STORAGE GROUP DEFINE
Command ==>

SCDS Name . . . . . : YOUR.OWN.SCDS
Storage Group Name  : HSMLOC

To DEFINE Storage Group, Specify:

Description ==> Tape storage group for HSM
              ==>

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

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

Use ENTER to Perform Verification and Selection;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 111. Defining a Tape Storage Group

After you define the storage group, you set the status for the storage group to each system that uses the tape library dataserer. You can temporarily prevent a storage group from being assigned by your storage group ACS routine by assigning its SMS Storage Group Status to DISNEW or DISABLE. The default for this attribute is ENABLE.

You can inquire about system-managed tape volumes as you do for system-managed DASD volumes. The Mountable Tape Volume List Application enables you to display data, including the volume's storage group, tape library, and expiration information. You can also use the LISTVOL line operator from the Storage Group Application to access the same information about system-managed tape volumes. Customers using DFSMSrmm can use the RMM LISTVOLUME command to list this same information. You can update the definitions of existing TAPE storage groups, or define new TAPE storage groups to include the names of tape library dataservers in the storage group definition.

## Writing the Storage Group ACS Routine

Figure 112 on page 226 shows how DFSMSHsm volumes are directed to system management.

```

WHEN (&HLQ = &HSM_HLQ &&
      &DSN(2) = &HSM_2LQ)
DO
  SET &STORGRP = 'HSMLOC'
  EXIT
END
/* Put HSM volumes in 3495*/
/* If out of scratches, */
/* create on 'old' drives */

```

Figure 112. Storage Group ACS Routine Fragment to Assign Tape Storage Groups

## Defining Data Classes for Tape

If you use a default data class for cartridges entered in the tape library dataserver, you must define it to SMS. The only attributes used in the default data class are:

- Compaction
- Media Type
- Recording Technology

The UNIT and TAPEHARDWARECOMPACTION options of the DFSMSHsm PARMLIB control the creation of non-system-managed DFSMSHsm tape volumes. These parameters cause DFSMSHsm to select the tape devices defined by the UNIT parameter, and determine if data should be written in compacted format. With system-managed DFSMSHsm volumes, the data class, if assigned, determines if compaction is used; the storage group ACS routine determines the storage group and, indirectly, the tape library and device that should be selected for DFSMSHsm.

DFSMSHsm uses the compaction data class attribute to determine whether to create the DFSMSHsm volume using IDRC. If you do not assign a data class for DFSMSHsm tape volumes in your data class ACS routine, then the options of TAPEHARDWARECOMPACTION, TAPEHWC and NOTAPEHWC, are used to make this determination for 3480X and 3490 devices, as before. For 3490E devices, data class must be assigned for DFSMSHsm to inhibit the use of IDRC compaction. If it is not, the tape is written using IDRC compaction.

**Related Reading:** For more information on controlling compaction in a DFSMSHsm environment, see *z/OS DFSMSHsm Implementation and Customization Guide*.

## Validating the SMS Configuration

The new SMS configuration is valid if the following conditions are true:

- If a library belongs to a storage group, the library must exist in the configuration.
- Every tape storage group must have at least one tape library.
- A tape storage group cannot be linked to an optical library.
- Tape storage group connectivity must match the connectivity of libraries associated with the storage group.

**Related Reading:** For additional guidelines on validating the SMS configuration, see *z/OS DFSMSdftp Storage Administration*.

## Creating the Tape Configuration Database

You can define an integrated catalog facility catalog to contain information about tape libraries and their volumes. First, define one general volume catalog. Then define one or more specific volume catalogs, based on your requirements. The collection of your general and specific volume catalogs is the tape configuration database. DFSMS determines which catalog to update, based on the first character of the volume serial number. If a specific volume catalog has been defined for the high-order character of the volume serial number, then DFSMS updates the specific volume catalog. If none exists, the general volume catalog is updated.

Naming conventions for volume serial numbers can help you balance the volume catalog update activity.

Before defining your tape libraries, ensure that only storage administrators can update the tape configuration database. Add STGADMIN.IGG.LIBRARY to the set of SMS facilities that are protected by RACF.

Figure 113 shows how to define a specific volume catalog. The name of the general catalog is SYS1.VOLCAT.VGENERAL, and SYS1.VOLCAT.VH is an example of the name of a specific volume catalog for tapes having serial numbers beginning with H. The HLQ, SYS1, can be replaced by another one if the LOADxx member of the PARMLIB is changed appropriately.

```
//DEFTPVL EXEC PGM=IDCAMS,REGION=400K
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

DEFINE UCAT -
  (NAME(SYS1.VOLCAT.VH) -
  VOLCAT -
  VOLUME(D65DM4) -
  CYL(1 1))
/*
```

Figure 113. Defining a Specific Volume Catalog

## Setting Up an Automated Tape Library

You create a tape library, a logical definition, for each grouping of tape volumes linked to a collection of tape devices. This definition becomes part of your active SMS configuration, and a library entry is generated in the tape configuration database. To create it, use option 3 on the Library Management Selection Menu. With ISMF, you can also redefine the tape library from information in the tape configuration database.

### Defining the Tape Configuration Characteristics

You connect your physical tape library and the SMS logical tape library through the ISMF Define Tape Library Application. You can also use this application to identify a console for tape messages and the tape library's system connectivity.

**Recommendation:** Ensure that systems in an SMS complex requiring access to a system-managed tape library are explicitly defined with a system name in the SMS configuration, rather than implicitly as part of a system group. If a system-managed tape library is attached to a system group, rather than to an

individual system, you cannot vary the library online and offline to the individual systems within the system group. The library must be online or offline to all systems in the group.

**Defining the Automated Tape Library Dataserver to SMS:** Another identification field, called the LIBRARY-ID, links the tape library definition with a tape library dataserver. Customer engineers establish this ID when the 3494 or 3495 is installed. You enter this ID in the LIBRARY-ID attribute.

**Defining the Tape Console:** You should identify a console to receive critical messages about 3494 or 3495 tape processing. Standard mount messages handled by the 3494 or 3495 accessor are not routed to the console, but are directed to a console log. Enter the name of this console as defined in your PARMLIB member, CONSOLxx, in the Console Name attribute.

**Defining Tape Library Connectivity:** You enable z/OS systems to use the tape library by defining system names in the Initial Online Status attribute. These system names must also reside in the base configuration of your active SMS configuration. A tape library that is defined to z/OS and physically connected to a system can be online or offline. If a tape library is offline, you can use the VARY SMS,LIBRARY command to bring the tape library online. If you do not set a status, SMS assumes that the tape library is not connected. Ensure that the tape configuration database is available to every system that uses the tape library.

### **Partitioning a Tape Library Dataserver**

You can partition your IBM TotalStorage Enterprise Automated Tape Library (3494 or 3495) to allow access from multiple SMS complexes that do not share a tape volume catalog. Partitioning can be viewed as dividing a physical library into multiple logical libraries.

**Related Reading:** For more information about partitioning a tape library dataserver, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

### **Setting Media-Related Attributes for Automated Tape Libraries**

Before you define your tape library, consider the tape subsystem characteristics of the devices in the tape library dataserver and their ability to support media interchange attributes-recording technology modes and media types required to support your tape data.

Volumes inserted in the tape library dataserver can automatically be assigned values for recording capability, media type, and compaction with the Entry Default Data Class. Or, you can supply this information in the cartridge entry installation exit or through DFSMSrmm.

Initially, the volumes in your tape library might be scratch volumes, private volumes, or a combination of both types of volumes. Enter the predominant type of use attribute, Private or Scratch, in the Entry Default Use attribute.

When you or DFSMSShsm eject volumes from the tape library dataserver, the volume entry in the tape configuration database can be retained or purged. Use the Eject Default to set this attribute to Keep or Purge based on your requirements. If you expect volumes to be reused in the library, use the default value, Keep, for this attribute.



When you use DFSMSrmm to eject volumes from the tape library daserver, the entry in the tape configuration database is optionally purged. DFSMSrmm has all the information needed to recreate the entries when the volumes are returned for reuse.

**Related Reading:** For more information about using DFSMSrmm, see *z/OS DFSMSrmm Implementation and Customization Guide*.

## Maintaining the Tape Library

Occasionally, you move tape cartridges stored in the tape library daserver offsite, extend the expiration date for a cartridge, or return a private volume to scratch. Information about system-managed tape volumes is maintained locally in the tape library daserver and in the system's volume catalogs, as follows:

- The tape library daserver's library manager maintains a database containing system-managed volume and tape device status information required to dispatch the accessor to mount, store, or eject a volume.
- The host maintains the tape configuration database containing cartridge information required to allocate system-managed tape volumes.

DFSMSdfp works together with the library manager to keep the tape configuration database synchronized with the library manager database.

The ISMF Mountable Tape Application, accessed from the ISMF Volume Application, lets you change information about system-managed tape volumes. You can ensure that the modification is reflected both in the tape configuration database and the library manager database. For example, you can change the use attribute of a system-managed tape volume from private to scratch status or from scratch to private, change the owner or storage group, eject cartridges, or change their shelf location.

**Recommendation:** Changes that are made using the ISMF Mountable Tape Application can be automatically synchronized with the tape management system if it fully supports the OAM tape management exits. You can use access method services to do the same; instead, use the ISMF application to ensure consistency between the library manager and the tape configuration database. Only the host's volume catalogs are updated by access method services.

You can also use DFSMSrmm to do this with the same level of integrity.

You can produce tailored lists of volumes and their usage characteristics using the ISMF Mountable Tape Volume Selection Entry panel. For a list of tape volumes, you can use the AUDIT list command to assess the accuracy of the contents of the tape configuration database. Issue the AUDIT function to schedule the audit. The AUDIT causes the automatic tape library daserver's accessor to go to the location referenced in the tape configuration database and verify the contents. If the results of the physical examination conflict with the volume catalog information, the error status field for the volume is updated with a code, indicating the type of error found. When the audit is complete, an acknowledgment is sent to the TSO session, and the storage administrator can view any audit errors by refreshing the tape volume list. You can also audit a tape library from the Tape Library List panel.

## Setting Up a Manual Tape Library

You can use the Library Management Selection Menu (Option 3) to create a tape library for each group of tape volumes linked to a collection of tape devices. This



definition becomes part of your active SMS configuration. A library entry is generated in the tape configuration database. With ISMF, you can also redefine the tape library from information in the tape configuration database.

Before you define the tape library to SMS, consider the type of tape subsystems in your installation. The following tape subsystems are supported in an MTL:

- 3480
- 3480x
- 3490
- 3590-Bxx
- 3590-Exx

### Using the HCD to Define the Manual Tape Library

The mere application of the MTL code onto a system — without an IPL or activate — enables the ability to define MTL UCBs. You specify the MTL keyword, in conjunction with the LIBRARY-ID and LIBPORT-ID keywords, to define the UCB as MTL resident. Figure 114 is an example of the Define Device Parameters / Features panel.

Define Device Parameters / Features Row 1 of 9

Command ==> \_\_\_\_\_ Scroll ==> PAGE

Specify or revise the values below.

Configuration ID . . : MVS1  
 Device number . . . : 0960                      Number of devices : 1  
 Device type . . . . : 3590

Parameter/ Feature	Value	P Req.	Description	
OFFLINE	No		Device considered online or offline at IPL	
DYNAMIC	Yes		Device supports dynamic configuration	
LOCANY	Yes		UCB can reside in 31 bit storage	
LIBRARY	No		Device supports auto tape library	
MTL	Yes		MTL resident device	
AUTOSWITCH	No		Device is automatically switchable	
LIBRARY-ID	12345		5 digit library serial number	
LIBPORT-ID	01		2 digit library string ID (port number)	
SHARABLE	No		Device is Sharable between systems	
F1=Help	F2=Split	F3=Exit	F4=Prompt	F5=Reset
F7=Backward	F8=Forward	F9=Swap	F12=Cancel	F22=Command

Figure 114. Define Device Parameters / Features

1. Invoke the Hardware Configuration Definition (HCD) facility.
2. Select the Define Device Parameters / Features panel
3. Specify MTL keyword, Yes
4. Specify LIBRARY-ID, observing the following rules:
  - a. Values are arbitrary, five-digit, hexadecimal values
  - b. Values cannot be all zeroes
  - c. Values must be unique among libraries, MTL or ATL
  - d. Values must be unique among systems that share TCDBs
  - e. Values must correspond with values specified in the SCDS
  - f. Devices attached to the same control unit cannot span control units within the same MTL

For example, given a physical string of four 3590-E11s, two of the drives on this control unit can be defined to one MTL and the second two to another MTL. Or, each drive can be defined to a different MTL.

5. Specify LIBPORT-ID, observing the following rules:
  - a. Values must be arbitrary, two-digit, hexadecimal values
  - b. Values cannot be all zeroes
  - c. Values Identify all devices attached to a specific control unit in a given library. Because LIBPORT-IDs map to control units, all devices on a given control unit must have the same LIBPORT-ID.
  - d. The number of devices with a given LIBPORT-ID must not exceed 16.

If the IODF resulting from this definition can be shared with systems that have no MTL support installed, or that have the full-function code installed but they are in coexistence mode (MTLSHARE has been specified in the LOADxx member), then the drives on MTL-defined UCBs will be defined as standalone (non-ATLDS) drives.

### **Defining the Tape Configuration Characteristics for MTL**

You connect your physical tape library and the SMS logical tape library through the ISMF Define Tape Library Application. You can also use this application to identify a console for tape messages and the tape library's system connectivity.

**Recommendation:** Ensure that systems in an SMS complex requiring access to a system-managed tape library are explicitly defined with a system name in the SMS configuration, rather than implicitly as part of a system group. If a system-managed tape library is attached to a system group, rather than to an individual system, you cannot vary the library online and offline to the individual systems within the system group. The library must be online or offline to all systems in the group.

**Defining the Manual Tape Library to SMS:** You use the LIBRARY-ID to link the tape library definition with a tape library. The installation creates the LIBRARY-ID for MTL libraries. You enter this ID in the LIBRARY-ID attribute.

Another ID field called the LIBPORT-ID field links the tape library definition to the particular control unit within the library. You enter this ID in the LIBPORT-ID attribute.

**Defining the Tape Console for MTL:** You should identify a console to receive critical messages about MTL processing. Enter the name of this console as defined in your PARMLIB member, CONSOLxx, in the Console Name attribute. For maximum visibility, MTL mount and demount messages are then issued to the named console and to the specified routing codes.

**Defining Manual Tape Library Connectivity:** You enable z/OS systems to use the tape library by defining system names in the Initial Online Status attribute. These system names must also reside in the base configuration of your active SMS configuration. A tape library that is defined to z/OS and physically connected to a system can be online or offline. If a tape library is offline, you can use the VARY SMS,LIBRARY command to bring the tape library online. If you do not set a status, SMS assumes that the tape library is not connected. Ensure that the tape configuration database is available to every system that uses the tape library.

**Supporting Devices and Device Mixtures within an MTL:** Devices that emulate 3490s and use media that is incompatible with real 3490s are not supported in an

MTL. This is because the mixture of such devices with real 3490s is not supported. Devices that are detected as operating in this mode are not allowed into the library. Currently those devices are:

- 3590-Bxx drives in 3490 emulation mode
- 3590-Exx drives in 3490 emulation mode
- 3591 drives which only run in 3490 emulation mode
- DAT tapes emulating 3490s

### **Indexing the Automatic Cartridge Loader**

MTL provides specific support for indexing of the Automatic Cartridge Loader. In a non-ATLDS environment, an ACL is indexed when the following conditions are met for a mount request:

- The request is for SCRTCH or PRIVAT
- The ACL is physically attached

In an MTL system, the SETCL default for MTL devices is NONE. This means that indexing is not to be done on this system. The reason for this default is that the cartridge loader status for an MTL device is not maintained across an IPL, so it is safest to default the cartridge loader to NONE during IPL processing. This requires the installation to explicitly state, through use of the LIBRARY SETCL command, the intended use of the device. In this way, if devices are being shared across systems, or are being dynamically set using the LIBRARY SETCL command, the ACL does not inadvertently get indexed with the wrong system's volumes or with the wrong media type. Otherwise, the ACL could be indexed when not appropriate, exhausting the ACL.

**Note:** The meaning of NONE on an MTL device is different from its meaning for drives in an automated tape library environment, in which it means that the cartridge loader is to be emptied.

Devices residing at non-ATLDS or non-MTL addresses (stand-alone addresses) are indexed. Other considerations apply when running in coexistence mode. In an MTL environment, the MTL system owns the ACLs of all MTL devices.

The existing LIBRARY SETCL command can be used to set the cartridge loader scratch media type for library-resident devices including MTL resident devices. See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for full details of this command.

For devices in a manual tape library, a new media type, ANY, can be specified. This indicates that media type preferencing through dataclass is not being used so that the ACL should be indexed for SCRTCH or PRIVAT mounts. This enables you to load any valid media type for the device.

#### **Rules:**

- If a preferenced allocation is made, the ACL will not be indexed.
- The LIBRARY SETCL command, when issued in a manual tape library environment, takes effect only on the system in which the command was issued (unlike the automated tape library environment). If multiple systems are sharing the scratch volumes in the cartridge loader, the same command should be issued on each sharing system with the non-sharing systems being set to NONE.

The following rules apply for indexing an ACL on a full-function MTL system.

**Exception:** There are other considerations if the system is running in coexistence mode (for example, if MTLSHARE has been specified).

- If the following conditions exist, indexing can take place.
  - The request must be a scratch request (the requested volser must be either SCRTCH or PRIVAT)
  - The ACL must be physically attached
  - INDEX=NO cannot have been coded on the MSGDISP call that prompted the service. This will be the case for system code in all but special circumstances.
- For stand-alone devices (tape devices residing at non-ATLDS or non-MTL addresses), indexing occurs.
- The following is true for MTL resident devices:
  - The default is to not index. This default can be overridden using the LIBRARY SETCL command. If the default has not been overridden, indexing does not occur. The default can be restored by using the NONE option with the LIBRARY SETCL command.
  - Specifying ANY with the LIBRARY SETCL command indicates that you want to index for all non-preferenced scratch mounts (those in which media type preferencing through dataclass is not being used). It also allows mixing of media types in the same ACL, if so desired.
    - If a mediatype is specified, indexing does not occur
    - If a mediatype is not specified, indexing occurs
  - If LIBRARY SETCL with a mediatype as been specified for this device, then:
    - If the allocation is not preferenced, index
    - If the allocation is preferenced and the requested mediatype matches that set for the device, then index.

### Setting Media-Related Attributes for MTL

Before you define your tape library, consider the tape subsystem characteristics of the devices in the tape library and their ability to support media interchange attributes, recording technology modes and media types required to support your tape data.

**Related Reading:** For more information about media types and recording technology, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Volumes inserted in the tape library daserver can automatically be assigned values for recording capability, media type, and compaction with the Entry Default Data Class. Or, you can supply this information in the cartridge entry installation exit or through DFSMSrmm.

Initially, the volumes in your tape library might be scratch volumes, private volumes, or a combination of both types of volumes. Enter the predominant type of use attribute, Private or Scratch, in the Entry Default Use attribute.

Estimate the number of scratch volumes by media type needed on average at all times to ensure that allocations proceed without interruption. When the count of available scratch volumes falls below the scratch threshold, DFSMSdftp sends a message to your console (designated in your tape library definition). The message stays on the console until the available scratch volumes exceed twice the specified threshold.

When you or DFSMSShsm eject volumes from the tape library, the volume entry in the tape configuration database can be retained or purged. Use the Eject Default to set this attribute to Keep or Purge based on your requirements. If you expect volumes to be reused in the library, use the default value, Keep, for this attribute.

When you use DFSMSrmm to eject volumes from the tape library, the entry in the tape configuration database is optionally purged. DFSMSrmm has all the information needed to recreate the entries when the volumes are returned for reuse.

Tape drives can be shared between systems. For more information, see “Sharing an IODF” on page 235.

---

## Managing DFSMSShsm Volumes

When you manage DFSMSShsm-owned volumes using DFSMS, you should evaluate how this management changes your current DFSMSShsm setup and operational control of DFSMSShsm media.

Currently, you can allocate volumes for an DFSMSShsm-managed scratch pool, or have DFSMSShsm use your general scratch pool.

**Recommendation:** Set up DFSMSShsm to use a general scratch pool managed by DFSMSdfp or DFSMSrmm. To implement this, specify the DEFERMOUNT parameter on the SETSYS SELECTVOLUME, TAPEDELETION, or PARTIALTAPE commands. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

Parameters controlling tape subsystem selection and compaction are affected by having system-managed DFSMSShsm volumes. Additionally, DFSMSShsm command options help you relate DFSMSShsm offline control data set contents to the DFSMS tape configuration database.

## Using DFSMSShsm Messages

Using a tape library dataserver to support DFSMSShsm mount activity reduces the necessity for some DFSMSShsm messages, such as messages to the operator confirming that the correct tape is mounted. This type of message is no longer produced by DFSMSShsm. However, if a hardware problem causes the tape library dataserver to only be usable in manual mode, DFSMSShsm continues to display these messages as before. The TAPEOUTPUTPROMPT on the TAPECOPY command is only valid when the tape library dataserver is operating in manual mode.

## Tracking Volumes in Connected Groups

If you plan to let the system manage a subset of your DFSMSShsm tape volume inventory, or if during migration you have volumes both inside and outside system-managed tape libraries, volumes in a given connected group might be separated. Use the LIST TIOC command to help you diagnose this condition, as follows:

- The SELECT(CONNECTED) parameter lists all volumes that are members of connected groups.
- The SELECT(CONNECTED(volser)) parameter lists all volumes in the connected group that are related to the volume serial number specified.

DFSMSHsm error messages highlight any inconsistencies within a connected group. These result when volumes are not entirely contained within the same tape library.

You can also use the DFSMSHsm LIST TTOC SELECT with the LIB or NOLIB option to check which DFSMSHsm migration level 2 and backup volumes are system-managed or non-system-managed. The LIST DUMPVOLUME SELECT with the LIB or NOLIB options do the same for dump volumes.

You can audit the data status of both migration and backup volumes, using the LIST TTOC SELECT command with the FULL, NOTFULL, or EMPTY options.

---

## Recommendations for Volsers

In an ATLDs, volumes must have a barcode strip that can be read by the vision system. Barcode values must be all alphanumeric, (uppercase A-Z, and 0-9) and no special characters are allowed. Because the internal and external volsers must match in an ATLDs, the same restriction applies to the internal volser on the tape volume itself.

Because there is no vision system associated with an MTL, there are no barcode strips on MTL volumes. Therefore, this restriction does not apply to MTL volsers, and the full range of valid volser characters is allowed. However, because there might be a future need for you to move MTL volumes to an ATLDs, ensure that all volsers on MTL volumes are alphanumeric. All other rules that apply to tape volumes in an ATLDs also apply to those in an MTL. Specifically:

- Both scratch and private tapes can be entered into an MTL
- A scratch volume cannot be requested using a specific volume serial number
- All volumes of a multivolume data set should reside in the same library, or all should reside outside a library. However, if they do not, you can enter the volumes through the Volume Not In Library installation exit (CBRUXVNL).
- All volumes of a multivolume data set must belong to the same tape storage group
- All volumes of a multivolume data set must be recorded using the same tape recording technology
- Volumes of a multivolume data set may be of media types that are consistent with the recording technology. For example, MEDIA9, MEDIA11, and MEDIA13 volumes can be used with enterprise format 4 (EFMT4) and enterprise encrypted format 4 (EEFMT4) recording technology.

---

## Sharing an IODF

A system having no MTL support, that uses an IODF containing MTL definitions, displays error messages such as CBDA384I during IPL or ACTIVATES. Otherwise, the IPL is successful, the UCBs are built correctly, and the devices are treated as standalone devices. If you have a sysplex, you might not want to enable MTL on all your systems at the same time. This might be because of IPL schedules for the various machines (enablement of the full-function code requires an IPL), or you can install MTL and IPL, but find that the work required to update SMS constructs, ACS routines, etc, on all systems is more than you can handle at one time.

Tape drives can be shared between systems. A device defined as MTL resident on one system can be used as a standalone device on a sharing system. This will happen by default if the IODF on the non-MTL system contains no references to

the MTL keyword in the definitions. However, environments often require the sharing of IODFs among systems that may be at varying support levels.

To support the sharing of IODFs that contain MTL definitions on systems without MTL support, coexistence is provided in the form of the full-function Tape UIM. This avoids warning messages otherwise issued during IPL or Activate when the MTL feature is encountered in the IODF on the coexisting system. The result is a tape device recognized and initialized as a standalone, non-ATLDS, drive.



## Appendix A. Sample Project Plan for DFSMS Implementation

This section presents a sample DFSMS implementation plan. This sample plan details the steps required to implement the DFSMS environment for implementing centralized storage management.

Build on these models to create your own plans. List the tasks in detail, and include:

- Interdependencies between tasks
- External dependencies
- The person responsible for each task
- A project schedule, including completion dates for each task
- Checkpoints to evaluate progress of tasks
- A written agreement of ownership and support of all areas involved
- Regular reviews of status and progress to date
- A way to modify the plan if required
- Management commitment to the plan

These sample plans cover items collected from numerous installations to provide a complete list.

ENABLING THE SYSTEM-MANAGED SOFTWARE BASE	Dependencies	Start Date	End Date	Evaluation Dates	Responsible Person
Install DFSMS					
Install DFSORT					
Install RACF					
Cache user catalogs in the catalog address space					
Cache VSAM buffers in hiperspace					
Set up ISMF Storage Administrator options					
Use ISMF cache support					
Use ISMF media support					
Use ISMF to identify data sets that cannot be system-managed					
Use system determined block size					

ACTIVATING SMS	Dependencies	Start Date	End Date	Evaluation Dates	Responsible Person
Allocate SMS control data sets					
Define Global Resource Serialization (GRS) resource names for active SMS control data sets					
Define a minimal SMS configuration					
Allocate a test case library					
Use ISMF option 7 or NaviQuest (ISMF option 11) to build test cases					
Translate and validate minimal configuration					
Test ACS routines					



<b>ACTIVATING SMS</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
Define the Storage Management Subsystem to z/OS					
Update SYS1.PARMLIB, members IGDSMSxx, IEFSSNxx, IEASYSxx					
Activate the Storage Management Subsystem					
Control Storage Management Subsystem processing					
Update the ACS routines to enforce standards					
Modify installation exits and user written system code					
Use simplified JCL to allocate data sets					
Use default unit and device geometry for non-system- managed data sets					

<b>MANAGING TEMPORARY DATA</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
Determine temporary data sets eligible for VIO					
Evaluate the amount of expanded and central storage on all CPUs if you have multiple processors					
Create a specific VIO storage group for each CPU based on its storage size					
Define appropriate storage class and storage groups					
Update storage class and storage group ACS routines					
Translate and validate new SMS configuration					
Define SMS storage classes and groups					
Initialize DASD volumes for LARGExx and PRIMExx storage groups					
Reactivate the configuration					
Test the ACS routines					
Prepare contingency plan					
Clean up old pools and volumes					

<b>MANAGING PERMANENT DATA</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
<b>TSO Data</b>					
Design and test performance and availability services for TSO data					
Design and test backup and space management services for TSO data					

<b>MANAGING PERMANENT DATA</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
Determine your physical space requirements for TSO data and add volumes to the PRIMExx and LARGExx storage groups					
Determine any additional resources required for DFSMSHsm space and availability management					
Activate new configuration					
Prepare contingency plan					
Place TSO data under system management					
Design and test automated data allocation using data classes					
<b>Batch Data</b>					
Evaluate applicability of sequential data striping for your batch data					
Evaluate use of pattern DSCBs by your batch jobs					
Design and test performance and availability services for batch data					
Design and test backup and space management services for TSO data					
Determine your physical space requirements for batch data and add volumes to the PRIMExx and LARGExx storage groups					
Determine any additional resources required for DFSMSHsm space and availability management					
Activate new configuration					
Design data classes for batch data					
Prepare contingency plan					
Place batch data under system management					
<b>Database Data</b>					
Evaluate cataloging procedures for database data					
For IMS data, consider converting any OSAM data sets to VSAM					
Verify consistency between DB2 STOGROUPs and SMS storage groups					
Ensure that SMS management class expiration attributes are synchronized with DB2's expiration information					
Set up DB2's DSNZPARM to have DFSMSHsm automatically recall DB2 data sets during DB2 access					

<b>MANAGING PERMANENT DATA</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
Design the storage classes and ACS routine					
Place the end-user databases and related database data sets under system management					
Test migrate/recall performance for end-user databases					
Place the production databases and related database data sets under system management					

<b>MANAGING TAPE</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
<b>Optimizing Tape Usage</b>					
Alter jobs that use uncataloged tape data sets					
Identify required JCL changes					
Remove pattern DSCB dependencies					
Identify jobs that use concatenated data sets					
Use global resource serialization					
Analyze the current tape environment					
Simulate the proposed tape mount management environment					
Implement advanced cartridge hardware					
Define DASD volumes to satisfy buffer requirements					
Define SMS classes and groups					
Create ACS routines					
Tune DFSMSHsm operation					
<b>Managing Tape Volumes</b>					
Evaluate job dependencies on demand allocation					
Evaluate job dependencies on UNIT keyword					
Evaluate job dependencies on volume affinity					
Define the tape environment to z/OS					
Define the OAM subsystem					
Define the storage classes and ACS routine					
Define the storage groups and ACS routine					
Create the tape configuration database					

<b>MANAGING TAPE</b>	<b>Dependencies</b>	<b>Start Date</b>	<b>End Date</b>	<b>Evaluation Dates</b>	<b>Responsible Person</b>
Define the tape libraries					
Translate and validate the new SMS configuration					
Create any system-managed tape exits, if required					
Test the new SMS configuration					
Activate the new SMS configuration					
Start the OAM subsystem					
Enter volumes in IBM 3495 Tape Library Dataserver					
Verify that new allocations are system-managed					
Convert the data by data category					



---

## Appendix B. Sample Classes, Groups, and ACS Routines

This appendix documents general-use programming interface and associated guidance information provided by DFSMS.

The DFSMS product tape contains a set of sample ACS routines. This appendix contains sample definitions of the SMS classes and groups that are used in the sample ACS routines.

You can base your SMS configuration on these routines, modifying them as needed.

**Related Reading:** For more information about sample classes, groups and ACS routines, see *z/OS Program Directory*.

---

### Data Classes Used in the Sample ACS Routines

The following list describes the data classes used in the examples.

#### **DIRECT**

Assigned to all VSAM relative record data sets and provides defaults for space assignments.

#### **ENTRY**

Assigned to all VSAM entry-sequenced data sets and provides defaults for space assignments.

#### **HFSDS**

Assigned to all hierarchical file system (HFS) data sets. It does not provide defaults for space assignments.

#### **KEYED**

Assigned to all VSAM key-sequenced data sets and provides defaults for space assignments and key offset. It does not provide a default for key length.

#### **LINEAR**

Assigned to all VSAM linear data sets and provides defaults for space assignments.

#### **NONTMM**

This data class indicates that a data set being written to tape should not be redirected to the tape mount management DASD buffer. Only system programmers and storage administrators are allowed to specify this data class during allocation.

#### **TMMACTV**

This data class directs tape data sets to the tape mount management DASD buffer. It provides default space assignments and allows the data set to span 10 volumes.

#### **TMMBKUP**

This data class directs tape data sets to the tape mount management DASD buffer, if the data set is a backup of another data set. It provides default space assignments and allows the data set to span 10 volumes.

**TMMTEMP**

This data class directs system-generated temporary tape data sets to the large storage group, rather than to tape or the tape mount management DASD buffer. It provides default space assignments and allows the data set to span 10 volumes.

**TAPACTV**

Assigned to tape data sets that are active and larger than 600 MB. It specifies that the data sets are to be compacted when written to the tape.

**TAPBKUP**

Assigned to tape data sets that are backups of other data sets and are larger than 600MB. It specifies that the data sets are to be compacted when written to the tape.

**TAPOSITE**

Assigned to tape data sets that are to be sent to offsite storage. It specifies that the data sets are to be compacted when written to the tape.

**TAPTEMP**

Assigned to system-generated tape data sets that are larger than 600MB. It specifies that the data sets are to be compacted when written to the tape.

**GDGF80**

This data class provides a model for generation data sets that require fixed-length records of 80 bytes. It provides defaults for space assignments, record format, and logical record length. This data class is not assigned by the sample ACS routine; users must explicitly assign this data class during allocation.

**GDGV104**

This data class provides a model for generation data sets that require variable-length records of 104 bytes. It provides defaults for space assignments, record format, and logical record length. This data class is not assigned by the sample ACS routine; users must explicitly assign this data class during allocation.

**DATAF**

Assigned to physical sequential data sets when the LLQ is DATA, FDATA, SYSGO, SYSLIN, or starts with the characters OBJ. It specifies fixed-length blocked records of 80 bytes, and provides a default space assignment.

**DATAV**

Assigned to physical sequential data sets when the LLQ is TEXT or VDATA. It specifies variable-length blocked records of 255 bytes, and provides a default space assignment.

**LISTING**

Assigned to physical sequential data sets when the LLQ is SYSOUT or begins with the characters LIST. It specifies variable-length blocked records of 137 bytes, with an ISO/ANSI control character, and provides a default space assignment.

**LOADLIB**

Assigned to partitioned data sets when the LLQ is RESLIB, PGMLIB, or begins with the characters LOAD. These data sets typically contain load modules. This data class specifies undefined records and provides a default space assignment.

**SRCFLIB**

Assigned to partitioned data sets and PDSEs when the LLQ begins with

the characters COB, FOR, CNTL, or JCL. However, it excludes data sets with the last two qualifiers SPFTEMP%.CNTL, where the % is replaced by any character. These data sets typically contain program source code or JCL procedures that require fixed-length records. This data class specifies fixed-length blocked records of 80 bytes, and provides a default space assignment.

### SRCVLIB

Assigned to partitioned data sets and PDSEs when the LLQ is PL1, PLI, SCRIPT, or CLIST. These data sets typically contain program source code or documentation that requires variable-length records. This data class specifies variable-length blocked records of 255 bytes, and provides a default space assignment.

### DATACOMP

Assigned to physical sequential data sets when the LLQ is LOG or ends with the characters LOG. It specifies variable-length blocked records of 137 bytes, and provides a default space assignment. This data class specifies that compression be used.

### HSMDC

Assigned to all DFSMSHsm system-managed data sets. It provides no defaults for space assignments.

Table 40 summarizes the attributes assigned to each data class for the sample SMS ACS routines.

*Table 40. Sample Data Classes for Data Sets*

Attributes	VSAM Data Classes			
NAME	DIRECT	ENTRY	KEYED	LINEAR
RECORG	RR	ES	KS	LS
SPACE AVGREC	U	U	U	U
SPACE AVG VALUE	4096	4096	4096	4096
SPACE PRIMARY	100	100	100	100
SPACE SECONDARY	100	100	100	100
VOLUME COUNT	1	1	1	1
Attributes	Tape Mount Management Data Classes			
NAME	NONTMM	TMMACTV	TMMBKUP	TMMTEMP
SPACE AVGREC	—	M	M	M
SPACE PRIMARY	—	200	200	200
SPACE SECONDARY	—	20	20	50
VOLUME COUNT	—	10	10	10
Attributes	Tape Data Classes			
NAME	TAPACTV	TAPBKUP	TAPOSITE	TAPTEMP
COMPACTION	Y	Y	Y	Y
MEDIA TYPE	MEDIA1	MEDIA1	MEDIA1	MEDIA1
RECORDING TECHNOLOGY	36TRACK	36TRACK	36TRACK	36TRACK
Attributes	Generation Data Set and Sequential Data Classes			
NAME	GDGF80	GDGV104	DATAF	DATAV
RECFM	F	V	FB	VB
LRECL	80	104	80	255
SPACE AVGREC	K	M	U	U
SPACE AVG VALUE	80	104	80	255



Table 40. Sample Data Classes for Data Sets (continued)

SPACE PRIMARY	10	5	5000	5000
SPACE SECONDARY	20	2	5000	5000
VOLUME COUNT	1	1	1	1
<b>Attributes</b>	<b>Sequential and Partitioned Data Classes</b>			
NAME	LISTING	LOADLIB	SRCFLIB	SRCVLIB
RECFM	VBA	U	FB	VB
LRECL	137	—	80	255
SPACE AVGREC	U	—	U	U
SPACE AVG VALUE	137	23476	80	255
SPACE PRIMARY	2000	50	5000	5000
SPACE SECONDARY	2000	50	5000	5000
SPACE DIRECTORY	—	62	62	62
DATA SET NAME TYPE	—	PDS	LIBRARY	LIBRARY
VOLUME COUNT	1	1	1	1
<b>Attributes</b>	<b>Additional Data Classes</b>			
NAME	DATACOMP	HFS	HSMDC	
RECFM	VBA	U	—	
LRECL	137	—	—	
SPACE AVGREC	M	—	—	
SPACE AVG VALUE	137	—	—	
SPACE PRIMARY	90000	—	—	
SPACE SECONDARY	5000	—	—	
DATA SET NAME TYPE	EXT	HFS	—	
IF EXT	P	—	—	
VOLUME COUNT	3	—	—	
COMPACTION	Y	—	Y	
MEDIA TYPE	—	—	MEDIA1	
RECORDING TECHNOLOGY	—	—	36TRACK	

See SAMPLIB for the sample data class ACS routine. This routine handles data set allocations on DASD and tape.

The routine first handles data allocations on DASD volumes. It allows users to specify any valid data class when allocating data sets on DASD.

If a user has not specified the data class, and is allocating a VSAM data set, the routine assigns a data class according to the record organization of the data set.

Non-VSAM data sets are assigned a data class according to the LLQ of the data set name. Separate classes are assigned for load libraries, source code libraries with fixed-length records, source code libraries with variable-length records, listing data sets, sequential data sets with fixed-length records, and sequential data sets with variable-length records. If the LLQ does not match those defined in the filter lists at the beginning of the routine, no data class is assigned to the data set.

Next, the routine handles tape data set allocations. Storage administrators and system programmers are allowed to use the NONTMM data class, which is tested for in the storage class routine so that these allocations are not redirected to the tape mount management buffers.

The routine distinguishes between tape allocations that should be allowed, and those that should be redirected to DASD so that DFSMSHsm can move the data to tape later. Data sets identified by data set name for offsite use, or as large backup, active, or temporary data, are assigned a data class that is used in the storage class routine to ensure the data is directed to tape rather than DASD.

All remaining tape allocations are assigned data classes that the subsequent routines direct to appropriate pool storage groups. These data sets are categorized as temporary based on data set type, backup based on program name, or active.

The remaining data sets are not assigned a data class.

SAMPLIB also contains the sample data class ACS routine for the permanent milestone.

---

## Storage Classes Used in the Sample ACS Routines

The following list describes the storage classes used in the examples.

### **STANDARD**

Assigned to most of your data sets that have average performance and availability requirements.

### **DFMCLASS**

Assigned to data sets created by Distributed FileManager/MVS.

### **GSPACE**

Lets the system programmer, database administrator, and storage administrator allocate data sets on specific volumes when required (for example, when data sets needed to recover a database should be allocated on a different volume than the one containing the database). It requires that specific volume requests are honored. It must be placed on the allocation request, because the ACS routine does not specifically assign the class.

### **NONVIO**

Assigned to DFSORT temporary work data sets to prevent them from being written to VIO.

### **NONSMS**

Used when creating data sets that should not be system-managed, and can only be used by system programmers, storage administrators, and database administrators. It is explicitly specified in JCL. The ACS routine must check for the NONSMS storage class and assign a null storage class.

### **MEDIUM**

Assigned to CICS/VSAM and user database data sets that can benefit from the DASD fast write capability of a 3990 storage control with extended functions.

**FAST** Assigned to DB2 and IMS production databases, IMS change accumulation logs and selected DL/1 databases. It is also assigned to CICS user data sets, temporary storage, and production libraries, which require higher than average performance.

### **FASTREAD**

Provides higher than average performance for data sets that are mostly read rather than written to. It must be specified on an allocation request,

because the ACS routine does not specifically assign the class. This class can only be used by system programmers, storage administrators, and database administrators.

#### **FASTWRIT**

Assigned to the following test system database data sets that can benefit from the DASD fast write capability of the 3990 storage control with extended functions:

- CICS intrapartition data set and selected CICS/VSAM databases
- IMS write ahead and online log data sets, and scratch pad area
- DB2 active log

#### **DBCRT**

Assigned to database data sets that provide system definition and recovery support. It specifies that the data sets are to be allocated on dual copy volumes and use DASD fast write.

#### **CRITICAL**

Provides continuous availability and accessibility with better than average performance. It must be entered on an allocation request, because the ACS routine does not specifically assign the class. This class can only be used by system programmers, storage administrators, and database administrators.

#### **MTLREMOT**

Assigned to a tape data set allocation. It ensures that the data set is allocated to a remote system-managed manual tape library dataserver.

#### **MTLSC**

Assigned to a tape data set allocation. It ensures that the data set is assigned to a system-managed manual tape library dataserver.

#### **ATLSC**

Assigned to a tape data set allocation. It ensures that the data set is assigned to a system-managed automatic tape library dataserver.

Table 41 summarizes the attributes assigned to each storage class for the sample SMS ACS routines.

*Table 41. Sample Storage Classes for Data Sets*

<b>Attributes</b>	<b>General Storage Classes</b>			
NAME	STANDARD	GSPACE	NONVIO	NONSMS
AVAILABILITY	STANDARD	STANDARD	STANDARD	STANDARD
ACCESSIBILITY	STANDARD	STANDARD	STANDARD	STANDARD
GUARANTEED SPACE	NO	YES	NO	NO
GUARANTEED SYNCHRONOUS WRITE	NO	NO	NO	NO
<b>Attributes</b>	<b>High Performance Storage Classes</b>			
NAME	MEDIUM	FAST	FASTREAD	FASTWRIT
DIRECT MILLISECOND RESPONSE	10	5	5	5
DIRECT BIAS	—	—	R	W
SEQUENTIAL MILLISECOND RESPONSE	10	5	5	5
SEQUENTIAL BIAS	—	—	R	W

Table 41. Sample Storage Classes for Data Sets (continued)

AVAILABILITY	STANDARD	STANDARD	STANDARD	STANDARD
ACCESSIBILITY	STANDARD	STANDARD	STANDARD	CONTINUOUS
GUARANTEED SPACE	NO	NO	NO	NO
GUARANTEED SYNCHRONOUS WRITE	NO	NO	NO	NO
<b>Attributes</b>	<b>High Availability Storage Classes</b>			
NAME	DBCRT	CRITICAL		
DIRECT MILLISECOND RESPONSE	10	10		
DIRECT BIAS	W	—		
SEQUENTIAL MILLISECOND RESPONSE	10	10		
SEQUENTIAL BIAS	R	—		
AVAILABILITY	CONTINUOUS	CONTINUOUS		
ACCESSIBILITY	CONTINUOUS	CONTINUOUS		
GUARANTEED SPACE	YES	NO		
GUARANTEED SYNCHRONOUS WRITE	NO	NO		
<b>Attributes</b>	<b>Additional Storage Classes</b>			
NAME	DFMCLASS	MTLREMOT	MTLSC	ATLSC
DIRECT MILLISECOND RESPONSE	5	—	—	—
SEQUENTIAL MILLISECOND RESPONSE	5	—	—	—
INITIAL ACCESS RESPONSE SECONDS	—	2	1	1
SUSTAINED DATA RATE	—	3	3	3
AVAILABILITY	STANDARD	STANDARD	STANDARD	STANDARD
ACCESSIBILITY	STANDARD	STANDARD	STANDARD	STANDARD
GUARANTEED SPACE	NO	NO	NO	NO
GUARANTEED SYNCHRONOUS WRITE	NO	NO	NO	NO

See SAMPLIB for the sample storage class ACS routine. This routine handles data set allocations on DASD and tape.

The routine first ensures that no storage class is assigned for data on devices not considered valid, for data on migration level 1 or 2 storage, for tape data, and for system data sets. Storage administrators and system programmers are also allowed to specify the NONSMS storage class during allocation, and this routine ensures that no storage class is assigned. These data sets are not system-managed.

Storage administrators, system programmers, and database administrators are allowed to assign any valid storage class.

Tape allocations that are assigned one of the tape mount management data classes are assigned either the STANDARD or NONVIO storage class:

- STANDARD means there are no special performance attributes for tape devices.
- Temporary DFSORT data sets are assigned the NONVIO storage class, which allows the storage group routine to assign the data to non-VIO storage. VIO tends to degrade DFSORT performance.

Next, the routine assigns storage classes to CICS, DB2, and IMS database data.

- The DBCRIT storage class, used for dual copy, is assigned to selected CICS, DB2, and IMS data.
- The FAST storage class, used for must-cache data, is assigned to selected CICS and IMS data.
- The FASTWRIT storage class, used for DASD fast write, is assigned to selected CICS, DB2, and IMS data.
- The MEDIUM storage class, used to provide better than average performance, is assigned to some CICS data.

All other CICS, DB2, IMS, and miscellaneous data sets are assigned the STANDARD storage class.

SAMPLIB also contains the sample storage class ACS routines for the temporary and permanent milestones.

---

## Management Classes Used in the Sample ACS Routines

The following list describes the management classes used in the examples.

### **STANDARD**

Assigned to most data sets and establishes your standard backup and retention policy.

### **STANDEF**

Assigned as the default for data intended for system management, but not assigned a specific management class by the management class ACS routine. It defines a full management environment where the data sets are migrated and backed up by DFSMSHsm, no automatic expiration is forced, and two backup versions are kept for each data set.

### **INTERIM**

Assigned to listing data sets based on the LLQ of the data set name. This class is designed for short life data, such as output or compiler listing data sets.

### **EXTBAK**

Assigned to source data sets based on the LLQ of the data set name. Designed for data sets that require additional backup copies, for example, source and program library data sets.

### **GDGBKUP**

Assigned to allow at least one generation of a GDG to remain on a primary volume. If the data set is not referenced in two days, DFSMSHsm directs the data set to migration level 2.

### **GDGPROD**

Assigned to allow at least one generation of a GDG to remain on a primary volume. If the current generation is not used in 15 days,

DFSMSHsm migrates it to migration level 1. The generations remain on migration level 1 for 60 days before being moved to migration level 2.

**TMMACTV**

Assigned to active tape data sets that are to be allocated in the tape mount management DASD buffer. If they are recovered or recalled, they are directed to the primary or large storage groups.

If the data set is a member of a GDG, the latest generation is kept on the primary volume, and the older generations are migrated to migration level 1. Rolled-off generation data sets are expired.

**TMMBKUP**

Assigned to all categories of backup tape data sets that are to be migrated directly from the tape mount management DASD buffer storage group to DFSMSHsm migration level 2 tape volumes.

**DBML2**

Assigned to DB2 image copy and archive log data sets, CICS extrapartition data sets, and IMS change accumulation logs. Image copy and archive log data sets are usually allocated as GDGs.

**DBSTAN**

Assigned to DB2 and CICS/VSAM user databases.

**NOACT**

Assigned to all production databases not assigned the DBML2 or DBSTAN management classes.

**PAYROLL1**

Assigned to the aggregate backup data set for the payroll application. It specifies attributes used to control retention and backup copy technique for the payroll application.

Table 42 summarizes the attributes assigned to each management class for the sample ACS routines.

**Exception:** This table does not include the management class attributes for objects. All object attributes are allowed to default to blanks.

*Table 42. Sample Management Classes for Data Sets*

Attributes	General Management Classes			
	STANDARD	STANDEF	INTERIM	EXTBAK
NAME	STANDARD	STANDEF	INTERIM	EXTBAK
EXPIRE AFTER DAYS NON-USAGE	NOLIMIT	NOLIMIT	3	NOLIMIT
EXPIRE AFTER DATE/DAYS	NOLIMIT	NOLIMIT	3	NOLIMIT
RETENTION LIMIT	NOLIMIT	NOLIMIT	3	NOLIMIT
PARTIAL RELEASE	YES	NO	YES IMMED	COND IMMED
MIGRATE PRIMARY DAYS NON-USAGE	15	2	3	15
LEVEL 1 DAYS NON-USAGE	60	15	60	60
COMMAND OR AUTO MIGRATE	BOTH	BOTH	BOTH	BOTH
# GDG ELEMENTS ON PRIMARY	—	—	1	—
ROLLED-OFF GDS ACTION	—	—	EXPIRE	—
BACKUP FREQUENCY	0	1	1	0
NUMBER BACKUP VERSIONS, DATA EXISTS	2	2	2	5

Table 42. Sample Management Classes for Data Sets (continued)

NUMBER BACKUP VERSIONS, DATA DELETED	1	1	1	1
RETAIN DAYS ONLY BACKUP VERSION	60	30	60	60
RETAIN DAYS EXTRA BACKUP VERSIONS	30	15	30	30
ADMIN OR USER COMMAND BACKUP	BOTH	BOTH	BOTH	BOTH
AUTO BACKUP	YES	YES	NO	YES
BACKUP COPY TECHNIQUE	STANDARD	STANDARD	STANDARD	STANDARD
AGGREGATE BACKUP COPY SERIALIZATION	FAIL	FAIL	FAIL	FAIL
ABACKUP COPY TECHNIQUE	STANDARD	STANDARD	STANDARD	STANDARD
<b>Attributes</b>	<b>Management Classes for Generation Data Group and Tape Mount Management</b>			
NAME	GDGBKUP	GDGPROD	TMMACTV	TMMBKUP
EXPIRE AFTER DAYS NON-USAGE	NOLIMIT	NOLIMIT	200	15
EXPIRE AFTER DATE/DAYS	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
RETENTION LIMIT	NOLIMIT	NOLIMIT	NOLIMIT	20
PARTIAL RELEASE	YES IMMED	YES IMMED	YES IMMED	YES IMMED
MIGRATE PRIMARY DAYS NON-USAGE	2	15	2	0
LEVEL 1 DAYS NON-USAGE	0	60	10	0
COMMAND OR AUTO MIGRATE	BOTH	BOTH	BOTH	BOTH
# GDG ELEMENTS ON PRIMARY	1	1	1	—
ROLLED-OFF GDS ACTION	EXPIRE	EXPIRE	EXPIRE	—
BACKUP FREQUENCY	—	0	1	—
NUMBER BACKUP VERSIONS, DATA EXISTS	—	1	2	—
NUMBER BACKUP VERSIONS, DATA DELETED	—	1	1	—
RETAIN DAYS ONLY BACKUP VERSION	—	60	60	—
RETAIN DAYS EXTRA BACKUP VERSIONS	—	30	30	—
ADMIN OR USER COMMAND BACKUP	NONE	BOTH	ADMIN	NONE
AUTO BACKUP	NO	YES	YES	NO
BACKUP COPY TECHNIQUE	STANDARD	STANDARD	STANDARD	STANDARD
AGGREGATE BACKUP COPY SERIALIZATION	FAIL	FAIL	FAIL	FAIL
ABACKUP COPY TECHNIQUE	STANDARD	STANDARD	STANDARD	STANDARD
<b>Attributes</b>	<b>Database and Miscellaneous Management Classes</b>			
NAME	DBML2	DBSTAN	NOACT	PAYROLL1
EXPIRE AFTER DAYS NON-USAGE	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
EXPIRE AFTER DATE/DAYS	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
RETENTION LIMIT	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
PARTIAL RELEASE	COND IMMED	NO	NO	NO
MIGRATE PRIMARY DAYS NON-USAGE	2	15	—	—
LEVEL 1 DAYS NON-USAGE	0	60	—	—

Table 42. Sample Management Classes for Data Sets (continued)

COMMAND OR AUTO MIGRATE	BOTH	BOTH	NONE	NONE
# GDG ELEMENTS ON PRIMARY	1	1	—	—
ROLLED-OFF GDS ACTION	EXPIRE	EXPIRE	—	—
BACKUP FREQUENCY	1	0	—	—
NUMBER BACKUP VERSIONS, DATA EXISTS	2	3	—	—
NUMBER BACKUP VERSIONS, DATA DELETED	1	1	—	—
RETAIN DAYS ONLY BACKUP VERSION	60	400	—	—
RETAIN DAYS EXTRA BACKUP VERSIONS	30	100	—	—
ADMIN OR USER COMMAND BACKUP	BOTH	BOTH	NONE	NONE
AUTO BACKUP	YES	YES	NO	NO
BACKUP COPY TECHNIQUE	CONCURRENT PREFERRED	CONCURRENT PREFERRED	STANDARD	STANDARD
AGGREGATE BACKUP # VERSIONS	—	—	—	2
AGGREGATE BACKUP RETAIN ONLY VERSION	—	—	—	1
AGGREGATE BACKUP RETAIN ONLY UNIT	—	—	—	MONTHS
AGGREGATE BACKUP RETAIN EXTRA VERSIONS	—	—	—	2
AGGREGATE BACKUP RETAIN EXTRA UNIT	—	—	—	WEEKS
AGGREGATE BACKUP COPY SERIALIZATION	FAIL	FAIL	FAIL	FAIL
ABACKUP COPY TECHNIQUE	STANDARD	STANDARD	STANDARD	STANDARD

SAMPLIB contains the sample management class ACS routine. This routine handles data set allocations on DASD and tape.

The routine first handles tape data sets. Data sets that are recalled or recovered by DFSMSHsm, that originally were written to the tape mount management buffer, are assigned the STANDARD or GDGBKUP management class, as appropriate. The storage group routine ensures that these data sets are not recalled to the buffer, but are placed in one of the standard pool storage groups.

New tape allocations are assigned the TMMACTV or TMMBKUP management classes. The storage group routine ensures that these data sets are assigned to the tape mount management buffers.

Storage administrators and system programmers are allowed to assign any valid management class.

The remainder of this routine handles standard new DASD data set allocations. Database data sets and generation data sets are assigned separate management classes to ensure special treatment.

All other data sets are assigned the STANDARD management class. These data sets are backed up and migrated by DFSMSHsm using standard management criteria.



SAMPLIB also contains a sample management class ACS routine for the permanent milestone.

---

## Storage Groups Used in the Sample ACS Routines

The following list describes the storage groups used in the examples. Table 44 on page 256 summarizes the attributes assigned to tape storage groups, and Table 45 on page 256 summarizes the attributes assigned to sample tape libraries.

### **PRIME80, PRIME90**

The primary storage groups are assigned to most system-managed data. This data includes interactive, batch, VSAM, striped, and multivolume data sets and some temporary data. These storage groups are also assigned to non-production database data.

Only data sets that are 285 MB or smaller are assigned to these storage groups.

PRIME80 contains 3380 DASD volumes of all models. PRIME90 contains 3390 DASD volumes of all models. Both storage groups allow automatic migration, backup, and dump processing.

### **LARGE80, LARGE90**

The large storage groups are assigned to data sets larger than 285MB. This includes database image copies, archive logs, some temporary data sets, and large interactive, batch, VSAM, striped, and multivolume data sets.

LARGE80 contains 3380 DASD volumes of all models. LARGE90 contains 3390 DASD volumes of all models. Both storage groups allow automatic migration, backup, and dump processing.

The high and low migration thresholds are set at 75% and 60%, respectively, so that sufficient space is available for new and extended data sets.

### **TMMBUF80, TMMBUF90**

The tape DASD buffer storage groups are assigned to new tape data sets that are to be allocated on DASD volumes before DFSMSHsm moves them to tape. The low threshold is set to 0% so that all data sets can be moved to tape. The AUTO MIGRATE attribute is set to I so that DFSMSHsm can migrate data hourly as needed.

### **TMMBFS80, TMMBFS90**

The overflow tape DASD buffer storage groups (also known as "spill" storage groups) are assigned to new tape data sets that are to be allocated on DASD volumes before DFSMSHsm moves them to tape. The overflow storage groups are only used when TMMBUF80 and TMMBUF90 are full. The low threshold is set to 0% so that all data sets can be moved to tape. The AUTO MIGRATE attribute is set to I so that DFSMSHsm can migrate data hourly as needed.

**CICS** The CICS database storage group is assigned to CICS production database data. It contains either 3380 or 3390 DASD volumes, but not both.

**DB2** The DB2 database storage group is assigned to DB2 production database data. It contains either 3380 or 3390 DASD volumes, but not both.

**IMS** The IMS database storage group is assigned to IMS production database data. It contains either 3380 or 3390 DASD volumes, but not both.

**VIO** The VIO storage group is assigned to temporary non-VSAM data sets

smaller than 20 MB, except DFSORT temporary data sets, which are assigned to the primary or large storage groups.

### ATLSG

This is a tape storage group for the 3494 or 3495 Automated Tape Library Dataserver. The tape storage group associates the automated tape library with the ATLSG storage group.

### MTLSG

This is a tape storage group for the 3495-M10 Manual Tape Library Dataserver. The tape storage group associates the manual tape library with the MTLTG storage group.

### MTLREMOT

This is a tape storage group for the 3495-M10 Manual Tape Library Dataserver. The tape storage group associates the "MTLREMOT" tape library with the MTLREMOT storage group. This storage group defines a remotely attached manual tape library connected by ESCON channels.

Table 43 summarizes the attributes assigned to each storage group for the sample SMS ACS routines.

Table 43. Sample DASD Storage Groups

Attributes	Primary and Large Storage Groups			
NAME	PRIME80	PRIME90	LARGE80	LARGE90
TYPE	POOL	POOL	POOL	POOL
AUTO MIGRATE	YES	YES	YES	YES
AUTO BACKUP	YES	YES	YES	YES
AUTO DUMP	YES	YES	YES	YES
DUMP CLASS	ONSITE, OFFSITE	ONSITE, OFFSITE	ONSITE, OFFSITE	ONSITE, OFFSITE
HIGH THRESHOLD	95	95	75	75
LOW THRESHOLD	80	80	60	60
GUARANTEED BACKUP FREQUENCY	15	15	15	15
SMS VOLUME OR STORAGE GROUP STATUS	ENABLE	ENABLE	ENABLE	ENABLE
Attributes	Tape Mount Management Storage Groups			
NAME	TMMBUF80	TMMBUF90	TMMBFS80	TMMBFS90
TYPE	POOL	POOL	POOL	POOL
AUTO MIGRATE	INTERVAL	INTERVAL	INTERVAL	INTERVAL
AUTO BACKUP	YES	YES	YES	YES
AUTO DUMP	YES	YES	YES	YES
DUMP CLASS	ONSITE, OFFSITE	ONSITE, OFFSITE	ONSITE, OFFSITE	ONSITE, OFFSITE
HIGH THRESHOLD	95	95	75	75
LOW THRESHOLD	0	0	0	0
GUARANTEED BACKUP FREQUENCY	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
SMS VOLUME OR STORAGE GROUP STATUS	ENABLE	ENABLE	ENABLE	ENABLE
Attributes	Database and VIO Storage Groups			
NAME	CICS	DB2	IMS	VIO
TYPE	POOL	POOL	POOL	VIO

Table 43. Sample DASD Storage Groups (continued)

VIOMAXSIZE	—	—	—	20MB
VIO UNIT	—	—	—	3380
AUTO MIGRATE	YES	YES	YES	—
AUTO BACKUP	YES	YES	YES	—
AUTO DUMP	YES	YES	YES	—
DUMP CLASS	ONSITE, OFFSITE	ONSITE, OFFSITE	ONSITE, OFFSITE	—
HIGH THRESHOLD	75	75	75	—
LOW THRESHOLD	60	60	60	—
GUARANTEED BACKUP FREQUENCY	NOLIMIT	NOLIMIT	NOLIMIT	—
SMS VOLUME OR STORAGE GROUP STATUS	ENABLE	ENABLE	ENABLE	ENABLE

Table 44. Sample Tape Storage Groups

Attributes	Tape Storage Groups				
NAME	ATLSG	MTLSG	MTLREMOT	ATLHMSG	MTLHMSG
TYPE	TAPE	TAPE	TAPE	TAPE	TAPE
LIBRARY	ATL	MTL	MTLREMOT	ATL	MTL
SMS STORAGE GROUP STATUS	ENABLE	ENABLE	ENABLE	ENABLE	ENABLE

Table 45. Sample Tape Libraries

Attributes	Tape Libraries		
NAME	ATL	MTL	MTLREMOT
LIBRARY ID	00AC1	10AC0	AAAC0
CONSOLE NAME	TLIC1CON	ALIB1CON	TLIC1CON
ENTRY DEFAULT DATA CLASS	—	—	—
ENTRY DEFAULT USE ATTRIBUTE	SCRATCH	SCRATCH	SCRATCH
EJECT DEFAULT	KEEP	KEEP	KEEP
MEDIA1 SCRATCH THRESHOLD	150	150	150
MEDIA1 SCRATCH NUMBER	0	0	0
MEDIA2 SCRATCH THRESHOLD	150	150	150
MEDIA2 SCRATCH NUMBER	0	0	0

SAMPLIB contains the sample storage group ACS routine. This routine handles DASD data and tape allocations that are redirected to DASD using tape mount management techniques. It does not assign tape storage groups.

Filter lists are used to identify production databases for CICS, DB2, and IMS, and the storage classes assigned to them.

The routine first implements tape mount management. Tape allocations of temporary data sets are assigned to the large storage groups, LARGE90 and LARGE80, thus preventing unnecessary tape mounts. Other tape allocations are identified according to management class, and redirected to the tape mount management buffers (TMMBUF90, TMMBUF80, TMMBFS90, and TMMBFS80). DFSMSHsm moves the data onto tape, reducing tape mounts and making more efficient use of tape resources.

The routine then handles temporary DASD data set allocations. Data sets smaller than 285MB are eligible for VIO or, if larger than the maximum size allowed by the VIO storage group, are allocated in the primary storage groups PRIME90 or PRIME80. The routine checks the storage class assigned, so that only data sets with the STANDARD storage class are eligible for VIO. This ensures that temporary VSAM and DFSORT work data sets are not assigned to the VIO storage group, because the storage class routine assigns the NONVIO storage class to those data sets. Temporary VSAM and DFSORT work data sets are assigned to the primary storage groups by the OTHERWISE statement at the end of the routine.

Next, the routine places CICS, DB2, and IMS production databases in the corresponding storage group.

Most data allocations are handled by the last two steps. Data sets 285MB or larger, including temporary data sets, are placed in the large storage groups. All other data sets are placed in the primary storage groups.

SAMPLIB also contains the sample storage group ACS routines for the activating, temporary, and permanent milestones.



---

## Appendix C. Installation and User Exits

You can use installation and user exits to call reporting programs, accounting routines, and housekeeping functions. You can also use installation and user exits to enforce local policy requirements, such as data set naming standards and space allocation restrictions.

In DFSMS, you can use ACS routines instead of some installation exits. For example, rather than using an installation exit to standardize JCL or enforce standards, consider using ACS routines.

Before you implement DFSMS, review your existing exits. If you continue to use existing installation and user exits, review their function and order of execution before designing the ACS routines. This ensures that the decisions made in the ACS routines are not unintentionally overridden.

The following tables list and describe the DFSMSdfp, DFSMSHsm, DFSMSdss, and MVS installation exits to review. They also indicate which exits are used for system-managed and non-system-managed data sets.

**Related Reading:** For detailed information on these exits, see the following publications:

- *z/OS DFSMS Installation Exits*
- *z/OS DFSMSdfp Storage Administration*
- *z/OS DFSMSHsm Implementation and Customization Guide*
- *z/OS MVS Installation Exits*

---

### DFSMSdfp Installation Exits

Programming Interface Information	
-----------------------------------	--

Table 46 describes DFSMSdfp installation exits used for either system-managed or non-system-managed data.

*Table 46. DFSMSdfp Installation Exits*

Name	Usage
IFG0EX0A	OPEN/EOV Exit for VTOC Entry Not Found: <ul style="list-style-type: none"><li>• Non-system-managed volumes.</li><li>• Called in OPEN or EOV if the VTOC entry is not found on the volume.</li><li>• If you already have DFSMSHsm installed or intend to install it, be aware that a DFSMSHsm-supplied exit routine recalls a data set if the VTOC entry is not found because the data set has been migrated.</li><li>• The data set that the user is attempting to allocate might be a migrated system-managed data set. If the exit is taken, an OPEN error occurs, indicating that the data set could not be found on the volume.</li><li>• To prevent this situation from occurring, physically remove the volume from the system and define it to a DUMMY storage group.</li></ul>

Table 46. DFSMSdfp Installation Exits (continued)

Name	Usage
IFGOEX0B	DCB OPEN Exit: <ul style="list-style-type: none"> <li>• Non-system-managed data sets.</li> <li>• Because IFGOEX0B exit is called before OPEN, you can mark the data set as reblockable. To do this, preserve the existing data set's block size and exit with condition code 12. This tells DFSMSdss to use the system-determined block size, and sets the reblockable indicator.</li> <li>• Some data sets (for example, load libraries) are sensitive to reblocking. Use care when reblocking them.</li> </ul>
IGBDCSX1 and IGBDCSX2	DASD Calculation Services Exits: <ul style="list-style-type: none"> <li>• Non-system-managed data.</li> <li>• These pre-DASD calculation services (DCS) exits and post-DCS exits provide some flexibility in, and control over, the selection of optimum VSAM control interval (CI) size or non-VSAM block size.</li> <li>• The block size you return is used not only by DFSMSdss, but also for all reblockable data sets.</li> <li>• <b>Recommendation:</b> Because DFSMS selects the optimum block size for each device type, do not use exits IGBDCSX1 and IGBDCSX2.</li> </ul>
IGGPREE00	DADSM Preallocation Exit routine associated with the IGGPRE00_EXIT dynamic exit: <ul style="list-style-type: none"> <li>• Non-system-managed data.</li> <li>• It is a common practice to use the IGGPRE00 exit to restrict allocation on specific volumes.</li> <li>• Do not use this exit for system-managed data sets.</li> <li>• See <i>z/OS DFSMS Installation Exits</i> for further information.</li> </ul>
IGDACSDC IGDACSSC and IGDACSMC	Data, Storage, and Management Class ACS Routine Exits: <ul style="list-style-type: none"> <li>• Data class ACS Routine Exit applies to system-managed and non-system-managed data.</li> <li>• Storage and management class ACS Routine Exits apply to system-managed data.</li> <li>• Called after the data, storage, and management class ACS routines have been run.</li> <li>• Can override the name of the class assigned by its ACS routine.</li> <li>• Can request that an ACS routine be reentered.</li> <li>• Parameters passed to these exits are the same as those available to the ACS routines.</li> <li>• Do not code any of these exits unless you require special processing that is not available in an ACS routine. Dummy exits degrade performance. Special processing could include functions such as calling other programs, writing SMF records, or writing generalized trace facility trace records.</li> </ul>
IGDACST	SMS pre-ACS exit: <ul style="list-style-type: none"> <li>• Used by a tape management system</li> <li>• Can set the &amp;MSPOOL, &amp;MSPOLICY, &amp;MSDEST, and &amp;MSPARM read-only variables to the ACS routines</li> <li>• Parameters passed onto this exit are the same as those available to the ACS exits.</li> </ul>

End Programming Interface Information

## DFSMSHsm Installation Exits

### Programming Interface Information

Table 47 describes DFSMSHsm installation exits used for either system-managed or non-system-managed data.

Table 47. DFSMSHsm Installation Exits

Name	Usage
ARCADEXT	Data Set Deletion Exit: <ul style="list-style-type: none"> <li>• Non-system-managed data sets</li> <li>• Delete-by-age and delete-if-backed-up processing</li> </ul>
ARCBDEXT	Data Set Backup Exit: <ul style="list-style-type: none"> <li>• System-managed and non-system-managed data sets.</li> <li>• Volume and migrated data sets backup.</li> <li>• VTOC entry indicates the SMS status. Use this flag to bypass processing for system-managed data sets.</li> </ul>
ARCCDEXT	Data Set Reblock Exit: <ul style="list-style-type: none"> <li>• System-managed and non-system-managed data sets.</li> <li>• Recall or recovery of sequential, blocked data if the SETSYS CONVERSION is specified and the VTOC entry indicates that the data set is not system-reblockable.</li> <li>• Use to change the block size to zero. This causes the data set to be marked system-reblockable, and allows DFSMSDfp to determine the block size for the device to which the data set is being recalled or recovered. The block size is recalculated if the data set moves to a different device type in the future.</li> <li>• Exercise care with load modules; some data sets (for example, load libraries) are sensitive to reblocking.</li> </ul>
ARCMDEXT	Data Set Migration Exit: <ul style="list-style-type: none"> <li>• System-managed and non-system-managed data sets</li> <li>• Data sets selected for migration from a user volume</li> <li>• Bypass processing for system-managed data sets</li> <li>• A VTOC entry indicates the SMS status</li> </ul>
ARCMMEXT	Second Level Migration Exit: <ul style="list-style-type: none"> <li>• System-managed and non-system-managed data sets.</li> <li>• Data sets selected for migration from a user volume.</li> <li>• A flag in the DFSMSHsmmigration control data set (MCDS) record indicates the SMS status. Use this flag to bypass processing for system-managed data sets.</li> </ul>
ARCMVEXT	Volume Migration Exit: <ul style="list-style-type: none"> <li>• System-managed data sets.</li> <li>• Defragments volumes when their fragmentation index is sufficiently large</li> </ul>
ARCRDEXT	Recall Exit: <ul style="list-style-type: none"> <li>• Non-system-managed data sets.</li> <li>• Determine the recall target volume from a set of volumes selected by DFSMSHsm.</li> </ul>



Table 47. DFSMShsm Installation Exits (continued)

Name	Usage
ARCSAEXT	Space Management and Backup Exit: <ul style="list-style-type: none"> <li>• System-managed and non-system-managed data sets</li> <li>• Use during space management and backup</li> <li>• VTOC entry indicates the SMS status</li> <li>• Parameter list has been extended to contain SMS-related data</li> </ul>

End Programming Interface Information

## DFSMSDss Installation Exit

### Programming Interface Information

Table 48 describes the DFSMSDss installation exit.

Table 48. DFSMSDss Installation Exit

Name	Usage
ADRREBLK	Data Set Reblock Exit: <ul style="list-style-type: none"> <li>• System-managed and non-system-managed data sets.</li> <li>• Called if REBLOCK is specified for COPY or RESTORE. If the VTOC entry for the data set indicates that the data set is reblockable, the data set is always reblocked and the ADDREBLK exit is not taken.</li> <li>• Return from the exit with a return code of 12 to enable the data set to be marked as reblockable. Be aware, however, that some data sets (such as load libraries) are sensitive to reblocking.</li> </ul>

End Programming Interface Information

## MVS Installation Exits

### Programming Interface Information

Table 49 describes MVS installation exits used for either system-managed or non-system-managed data.

Table 49. MVS Installation Exits

Name	Usage
IEFDB401	MVS Dynamic Allocation Exit: <ul style="list-style-type: none"> <li>• Enforce allocation standards.</li> <li>• Called prior to ACS services execution; therefore, action taken in this exit must be designed considering your ACS routines' logic.</li> </ul>
IGGDASU2 and IGGDARU3	Installation-Replaceable Module: <ul style="list-style-type: none"> <li>• Called before processing the DADSM request for SCRATCH and RENAME.</li> <li>• If you have written your own IGGO3ODU or IGGO29ODU modules, update these modules for the DFSMS environment.</li> </ul>

End Programming Interface Information

---

## Appendix D. Accessibility

Accessible publications for this product are offered through the z/OS Information Center, which is available at [www.ibm.com/systems/z/os/zos/bkserv/](http://www.ibm.com/systems/z/os/zos/bkserv/).

If you experience difficulty with the accessibility of any z/OS information, please send a detailed message to [mhvrcfs@us.ibm.com](mailto:mhvrcfs@us.ibm.com) or to the following mailing address:

IBM Corporation  
Attention: MHVRCFS Reader Comments  
Department H6MA, Building 707  
2455 South Road  
Poughkeepsie, NY 12601-5400  
USA

---

### Accessibility features

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 Vol 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.

---

### Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the z/OS Information Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually

exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

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

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

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

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

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- ! means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1!

(KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

- \* means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the \* symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1\* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3\*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

**Note:**

1. If a dotted decimal number has an asterisk (\*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The \* symbol is equivalent to a loop-back line in a railroad syntax diagram.
- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the \* symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the \* symbol, is equivalent to a loop-back line in a railroad syntax diagram.



---

## Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not 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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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

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

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

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

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

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

Site Counsel  
IBM Corporation  
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.

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

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

#### COPYRIGHT LICENSE:

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

---

## Policy for unsupported hardware

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

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

---

## Minimum supported hardware

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

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

---

## Programming interface information

This publication primarily documents information that is NOT intended to be used as programming interfaces.

Programming interface information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

————— **Programming Interface Information** —————

————— **End Programming Interface Information** —————

---

## Trademarks

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





---

# Index

## Special characters

- &ACCT\_JOB variable 50
- &ACSENVIR variable 49
- &ALLVOL variable 47
- &ANYVOL variable 47
- &APPLIC variable 136
- &DATACLAS variable 46
- &DSN variable 48
- &DSTYPE variable 136
- &EXPDT variable 48
- &GROUP variable 152
- &LLQ variable 58
- &MAXSIZE variable 48
- &MGMTCLAS variable 46
- &NQUAL variable 48
- &NVOL variable 48
- &PGM variable 205
- &RECORG variable 131
- &RETPD variable 48
- &SIZE variable 48
- &SPECIAL\_USERS variable 152
- &STORCLAS variable 46
- &STORGRP variable 46
- &SYSNAME variable 50
- &SYSPLEX variable 50
- &USER variable 109
- &VALID\_STORAGE\_CLASS variable 152

## Numerics

- 32-system mode 86
- 3880 Model 23 storage control 25
- 3990 Model 3 Storage Control 9
- 3990 Model 6 Storage Control 9
- 8-name mode 86

## A

- ABACKUP command 39
- ABARS 5, 39
- absolute track data sets 73, 76
- access methods
  - basic sequential access method (BSAM) 25
  - improved performance for data sets 23, 25
  - indexed sequential access method (ISAM) 52, 76
  - object access method (OAM)
    - defining to z/OS 223
    - managing tape data 222
  - overflow sequential access method (OSAM) 148, 239
  - preallocating the VVDS 53
  - queued sequential access method (QSAM) 25
- accessibility 263
  - contact IBM 263
  - features 263

- accessing a dataserver from multiple SMS complexes 228
- ACDS
  - calculating the size 81
  - defining to z/OS 81
  - description 16
- ACL indexing 232
- ACS
  - data class routine
    - example 246
    - tape mount management 204
  - environment
    - RMMPOOL 47
    - RMMVRS 45, 47
  - language 46
  - management class routine
    - example 253
    - tape mount management 207
  - routine
    - creating test cases 113
    - data class example 246
    - description 14
    - determining classes for Distributed FileManager/MVS data sets 15
    - for minimal configuration 98
    - management class example 253
    - managing temporary data 113
    - running test cases 116
    - sample 243
    - storage class 113
    - storage class example 249
    - storage group example 256
    - tape mount management 204, 206, 207, 208
    - testing 113
    - translating 101, 113
  - selecting ISMF application 98, 113
  - storage class routine
    - example 249
    - managing temporary data 113
    - tape mount management 206
  - storage group routine
    - example 256
    - tape mount management 208
- ACS Application Selection panel 98, 113
- ACS routine
  - creating 112
  - writing 113
- ACS Test Case Define panels 114
- ACS Test Selection panel 113
- activating
  - minimal configuration 79
  - new SMS configuration 105
  - SMS subsystem 106
- active control data set (ACDS) 81
- Add'l Volume Amount (data class attribute) 32
- ADDVOL command 36
- aggregate
  - backup and recovery support (ABARS) 39

- aggregate (*continued*)
  - group
    - overview 14
    - using for backup and recovery 39
- allocation
  - data class 8
  - data set 42
  - DFSORT temporary data sets 110
  - improved control 3
  - simplified 3
  - storage class, specific volume 33
  - VSAM data sets 57
- analyzing
  - tape 182
- ARECOVER command 39
- assistive technologies 263
- attribute
  - Add'l Volume Amount, data class 32
  - Compaction (data class) 219
  - Guaranteed Backup Frequency, storage group 36
  - Media Type (data class) 219
  - Recording Technology (data class) 219
- authorizing access using SAF 68
- automated tape libraries 219
- automated type libraries
  - defining 227
- automatic cartridge loader 232
- automatic class selection (ACS) 98
- automatic tape library dataserver 219

## B

- backup and recovery support, aggregate (ABARS) 39
- backup-while-open function 5, 58, 147
- base configuration, SMS
  - ACS routine
    - storage class 99
    - storage group 98
    - translating 101
    - writing 98
  - classes and groups 87
  - default management class 95
  - defining 83
  - description 16
  - storage class
    - ACS routine 99
    - defining 87
  - storage group
    - ACS routine 98
    - defining 91
    - system group 86
- batch data
  - benefits, SMS 135
  - defining sequential data striping 142
  - implementing DFSMS
    - defining data classes 141
    - fallback strategy 137
    - general considerations 136

- batch data (*continued*)
  - implementing DFSMS (*continued*)
    - managing GDGs 139
    - managing generation data sets (GDGs) 137
    - managing non-GDGs 140
    - performance and availability 138
    - physical storage 141
  - migrating to DFSMS
    - DFSMSdss migration 142
  - using sequential data striping 137
- BLKSIZE parameter 57
- block size, system-determined 77, 237
- BSAM data sets 25

## C

- cache statistics 4
- cache-capable 3990 storage control 23
- caching, dynamic data set
  - database data 145
  - improving performance 23
  - using dynamic cache management 24
- calculating
  - free space percentage 196
  - the size of control data sets 81
  - tracks allocated 85
- cartridge entry exit 228, 233
- catalog
  - protecting using RACF 68
- CDS Application Selection panel 84, 102
- CICS data 149
- CICSVR 1, 2
- CISIZE parameter 57
- classes and groups, SMS
  - data class 200
  - management class
    - defining default 95
    - tape mount management 202
  - restricting user access using RACF 70
  - storage class
    - minimal configuration 87
    - tape mount management 202
  - storage group
    - minimal configuration 91
    - tape mount management 203
  - translating 101
- classifying tape data sets 181
- command
  - ABACKUP 39
  - ADDVOL 36
  - ARECOVER 39
  - DEFINE 8, 16
  - DEVSERV 106, 107
  - DISPLAY SMS 106
  - SET SMS 107
  - SETSMS 106
  - SETSYS 38
  - VARY SMS 106
- COMMDS
  - calculating the size 81
  - defining to z/OS 81
  - description 16
- communications data set (COMMDS) 81
- Compaction (data class attribute) 219

- compatibility mode 86
- compression 58
- concurrent copy
  - database data 146
  - description 30
  - planning for 30
  - using 31
- configuration, SMS
  - activating 79
  - allocating control data sets 80
  - contents of minimal configuration 83
  - defining 79
  - minimal 79
  - planning for 80
- control data set, SMS
  - ACDS 16
  - allocating 80
  - calculating sizes 81
  - COMMDS 16
  - defining to z/OS 81
  - overview 16
  - SCDS 16
- Copy Entry panel 87
- copy services
  - FlashCopy 29
  - PPRC 29
  - XRC suspend/resume
    - unplanned outages 29
- copying storage classes
  - Copy Entry panel 87
  - List panel 87, 89

## D

- DADSM 3
- DASD
  - volume selection 12
- DASD volume 19
- data class
  - ACS routine
    - example 246
    - tape mount management 204
  - Add'l Volume Amount attribute 32
  - allocating
    - HFS data 132
    - TSO data 128
    - VSAM extended addressability 59
- attributes 58
- defining for tape mount management 200
- description and summary 243
- NONTMM 201
- overriding 61
- overview 8
- simplifying data set allocation 58
- TAPACTV 201
- TAPBKUP 201
- TAPOSITE 201
- TAPTEMP 201
- TMMACTV 201
- TMMBKUP 201
- TMMTEMP 201
- Data Class Define panels 58
- data conversion in-place 51, 52, 54, 122, 126
- data conversion with movement 51, 52, 54, 122, 126

data set

- ACDS 16
  - assigning to a fault-tolerant device 29
- COMMDS 16
- data class allocation 8
- extended addressability 59
- extended format 59
- List panel 74
- multivolume
  - conversion eligibility 54
  - Improved data recording capability (IDRC) 197
  - preallocation, SMS support 23, 32
  - tape mount management
    - candidate 181
    - TMMBKUP data class 201
  - protecting with RACF 68
- SCDS 16
- Selection Entry panel 73
- separation 29, 84, 86
- stacking
  - description 209
  - volume reference 211
  - volume serial 212
- temporary 111
- Data Set List panel 74
- Data Set Selection Entry panel 73

data types

- absolute track data sets 73, 76
- ISAM data sets 73
- unmovable data sets 73

DATABASE 2 5

database data

- benefits, SMS 145
- implementing DFSMS
  - backup and recovery 162
  - CICS data 149
  - DB2 data 157
  - IMS data 153
  - major tasks 148
  - physical storage 166
  - using dynamic data set caching 145
  - using point-in-time copy 146

DATACLAS keyword 57

DB2 5
 

- partitioned table space 169

DB2 data 157

DCOLLECT function 58, 78

Default Device Geometry, track calculation 85

DEFINE command 8, 16

defining
 

- control data set 81
- default management class 95
- minimal SMS configuration 83
- pool storage group attributes 93
- sequential data striping 142
- SMS base configuration 83
- SMS system 86
- storage class 87
- storage group 91
- Storage Management Subsystem 103
- system group 86, 87

designing ACS routines 46

device pool 221

DEVSERV operator command 106, 107

- DFSMS 1, 19
  - DFSMSdfp 1
  - DFSMSdss 1
  - DFSMSHsm 1
  - DFSMSrmm 1
  - DFSMSStvs 1
  - Optimizer 2, 4
- DFSMS environment
  - benefits 2
  - managing data 6
  - protecting with RACF 67
- DFSMS Fast Implementation Techniques (FIT) 18, 19
- DFSMS implementation 237
- DFSMSdfp
  - description 1
- DFSMSdss
  - description 1
- DFSMSHsm 36
  - commands 37
  - description 2
  - tuning 218
- DFSMSrmm
  - description 2
  - managing tape data 222
  - updating tape configuration database 222, 223
- DFSORT 1
  - allocation considerations 110
  - description 2
  - ICEGENER facility 64
- directing tapes to a remote library 224
- disaster recovery
  - classifying tapes 181
  - description 5
  - directing tapes to a remote library 224
  - ejecting tape volumes 222
  - overview 14
  - using aggregate backup and recovery support (ABARS) 5
- DISPLAY SMS operator command 106, 107
- Distributed FileManager/MVS data set 15, 23
- DO/END statement 46
- documentation, project plan 237
- DS separation profile parameter 84, 86
- DSCB
  - replacing pattern DSCBs 137, 141
- dual copy
  - assigning a data set 29
  - benefits for batch data 135
  - DFSMS benefits for database data 146
- dynamic data set caching
  - benefits 23
  - database data 145
  - using 24
- Dynamic Volume Count parameter 59

## E

- Edit-Entry panel 98
- ejecting tape volumes for disaster recovery 222
- enabling the software base 67

- Enterprise Storage Server (ESS) 9
- ESS 9
  - copy services 29
  - multiple allegiance 23
  - parallel access volume 23
- estimate report 185
- EXCP data sets 73
- EXIT statement 46
- exits 262
  - installation 259
  - user 259
- expiration date, overriding 97
- extended format 59
- Extent Constraint 58

## F

- fault-tolerant device, assigning 29
- FILTLIST statement 46
- FIT 18, 19
- free space percentage, calculating 196
- FREESPACE parameter 57

## G

- GDG processing 64
- GRS
  - allocating control data sets 82
  - converting RESERVE/RELEASE requests 181
- Guaranteed Backup Frequency (storage group attribute) 36
- guaranteed space preallocation 32

## H

- Hardware Configuration Definition 40, 223
- HFS
  - allocating HFS data sets 132
  - data class example 243
- Hiperbatch 28
- Hiperspace 110

## I

- I/O
  - analysis, using DFSMS Optimizer feature 4
  - priority scheduling 158
  - statistics 158
- IBM NaviQuest for z/OS 18, 20
- IBM TotalStorage Enterprise Automated Tape Library (3495) 4, 63, 219
- ICEGENER facility 64
- ICKDSF (Device Support Facilities) 53
- identifying SMS benefits to users 57
- IDRC 197
- IEASYSy member 104
- IEBGENER processing 64
- IEFSSNxx member 103
- IEHLIST processing 64
- IEHMOVE processing 64
- IEHPROGM processing 64

- IF/THEN statement 46
- IGDSMSxx member 104
- IMBED parameter 57
- implementing
  - database data in DFSMS environment 148
  - permanent data to DFSMS environment 119
  - system-managed storage, using DFSMS FIT 18, 19
  - temporary data in DFSMS environment 110
- Improved data recording capability (IDRC) 197
- IMS data 153
- in-place conversion 51, 52, 54, 122, 126
- indexing functions, ACS 47
- indexing the ACL 232
- initializing volumes 21, 53
- installation exit 259
- integrity, data 3
- interactive storage management facility (ISMF) 84
- interval
  - migration 36, 181
  - synchronization time 104
  - threshold 36
- ISAM data sets 52, 76
- ISMF
  - ANALYZE line operator 77
  - identifying ineligible data sets 73
  - INIT line operator 77
  - managing storage devices 77
  - panels
    - ACS Application Selection 98, 113
    - ACS Test Case Define, Page 1 of 4 114
    - ACS Test Case Define, Page 2 of 4 113
    - ACS Test Selection 113
    - CDS Application Selection 84, 102
    - Copy Entry 87
    - Data Class Define, Page 1 58
    - Data Class Define, Page 2 58
    - Data Class Define, Page 3 58
    - Data Class Define, Page 5 58
    - Data Set List 74
    - Data Set Selection Entry 73
    - Edit-Entry 98
    - Management Class Application Selection 95
    - Management Class Define (3) 35
    - Management Class Define (page 1) 34
    - Management Class Define (page 2) 34
    - Management Class Define, Page 1 of 5 95
    - Management Class Define, Page 2 of 5 95
    - Management Class Define, Page 3 of 5 97
    - Pool Storage Group Define 40, 91
    - SCDS Base Define, Page 1 of 2 84
    - SCDS Base Define, Page 2 of 2 83
    - SMS Storage Group Status Define 91

ISMF (*continued*)  
 panels (*continued*)  
   SMS Volume Status Define 91  
   Storage Class Application Selection 88  
   Storage Class Define 25  
   Storage Class List 87, 89  
   Storage Group Application Selection 91  
   Storage Group Volume Selection 94  
   Tape Storage Group Define 224  
   Test ACS Routines 116  
   Translate ACS Routines 101  
   Validate ACS Routines or Entire SCDS 102  
 protecting functions 71  
 SETCACHE line operator 77  
 viewing the starter set 72

## J

JCL, changing 61  
 JES3 restrictions 50

## K

keyboard  
   navigation 263  
   PF keys 263  
   shortcut keys 263  
 KEYLEN parameter 57  
 KEYOFF parameter 57

## L

large data sets, separating 44  
 library  
   tape conversion activities 173  
 library manager 229  
 LIKE keyword 57  
 locate processing 62

## M

management class  
   ACS routine  
     example 253  
     tape mount management 207  
     translating 101  
   default for minimal configuration 95  
   Define panels 34, 95  
   description and summary 250  
   managing with DFSMSHsm 34  
   overview 10  
   report 193  
   selecting ISMF application 95  
   specifying attributes 34  
   tape mount management 202  
   TMMACTV 202  
   TMMBKUP 202  
 Management Class Application Selection panel 95

Management Class Define panel (page 1) 34  
 managing  
   storage devices 77  
   tape volumes 219, 220  
 manual tape libraries 219  
 configuration characteristics 231  
   connectivity 231  
   defining 229  
   device mixtures 231  
   HCD 230  
   indexing 232  
   mixtures 231  
   tape console 231  
   using 220  
 maximum gigabyte report 188  
 measuring data set accesses 158  
 Media Type (data class attribute) 219  
 migration  
   attributes 33  
   defining a management class 10  
   interval 181  
   permanent data 119  
   using DFSMSHsm 2  
 minimal configuration, SMS  
   allocating control data sets 80  
   contents 83  
   defining  
     classes and groups 87  
     default management class 95  
     storage class 87  
     storage group 91  
     translating ACS routines 101  
     writing ACS routines 98, 99  
 mixing devices in a storage group 43  
 multiple allegiance 23, 32  
 multivolume data sets  
   conversion eligibility 54  
   Improved data recording capability (IDRC) 197  
   preallocation, SMS support 23, 32  
   tape mount management 181  
   TMMBKUP data class 201  
 MVS  
   operator command  
     DEVSERV 106  
     DISPLAY SMS 106  
     SET SMS 106  
     VARY SMS 106

## N

navigation  
   keyboard 263  
 NaviQuest 18, 20  
 NetView File Transfer Program 222  
 NONTMM data class 201  
 Notices 267

## O

OAM  
   defining to z/OS 223  
   managing tape data 222  
 object  
   data class 9

object (*continued*)  
   storage class 7, 10  
 operator command, MVS  
   DEVSERV 106  
   DISPLAY SMS 106  
   SET SMS 106  
   SETSMS 106  
   VARY SMS 106  
 Optimizer 2, 4  
 optimizing  
   current tape environment 171  
 OSAM data sets 148, 239  
 overflow storage group 199, 215  
 overriding  
   data class 61  
   expiration date 97, 180  
 OVRD\_EXPDT keyword 97, 180

## P

panels, ISMF  
   ACS Application Selection 98, 113  
   ACS Test Case Define, Page 1 of 4 114  
   ACS Test Case Define, Page 2 of 4 113  
   ACS Test Selection 113  
   CDS Application Selection 84, 102  
   Copy Entry 87  
   Data Class Define, Page 1 58  
   Data Class Define, Page 2 58  
   Data Class Define, Page 3 58  
   Data Class Define, Page 5 58  
   Data Set List 74  
   Data Set Selection Entry 73  
   Edit-Entry 98  
   Management Class Application Selection 95  
   Management Class Define (3) 35  
   Management Class Define (page 1) 34  
   Management Class Define (page 2) 34  
   Management Class Define, Page 1 of 5 95  
   Management Class Define, Page 2 of 5 95  
   Management Class Define, Page 3 of 5 97  
   Pool Storage Group Define 40, 91  
   SCDS Base Define, Page 1 of 2 84  
   SCDS Base Define, Page 2 of 2 83  
   SMS Storage Group Status Define 91  
   SMS Volume Status Define 91  
   Storage Class Application Selection 88  
   Storage Class Define 25  
   Storage Class List 87, 89  
   Storage Group Application Selection 91  
   Storage Group Volume Selection 94  
   Tape Storage Group Define 224  
   Test ACS Routines 116  
   Translate ACS Routines 101  
   Validate ACS Routines or Entire SCDS 102  
 parallel access volume 23

parallel access volume (PAV) option 28  
 parallel access volumes 32  
 Parallel Sysplex  
   name 86, 93  
 PARMLIB, updating members  
   IEASYSyy 104  
   IEFSSNxx 103  
   IGDSMSxx 104  
 partial release 19  
 Partial Release (management class attribute) 97  
 partitioning a tape library  
   dataserver 228  
 pattern DSCBs 137, 141  
 PAV-alias 33  
 PAV-base 33  
 PDS 122  
 PDSE  
   advantages over PDSs 122  
   format 122  
   multisystem sharing of 122  
 permanent data  
   implementing DFSMS 119  
   migrating 119  
 point-in-time copy 30, 146  
 Pool Storage Group Define panel 40, 91  
 pre-ACS interface  
   DFSMSrmm support 204  
 project plans, sample 237  
 protecting, DFSMS environment  
   creating a security policy 67  
   functions and commands 68  
   ISMF functions 71  
   restricting access in RACF profile 69  
   SMS control data sets 68  
   system-managed data sets 68

## Q

QSAM data sets 25

## R

RACF 1, 67  
   default SMS classes 69  
   description 2  
   FIELD resource class 70  
   PROGRAM resource class 71  
   protecting DFSMS environment 67  
   protecting SMS facilities  
     access to system-managed data 68  
     use of ISMF functions 71  
   resource classes 70  
   restricting user access to SMS classes and groups 70  
 RAID architecture devices 10  
 RAMAC device  
   assigning a data set 29  
   meeting availability requirements 9  
   sequential data striping 25  
 RECALL command 37  
 recommendations  
   allocating control data sets 81  
   assigning data classes 9, 130

recommendations (*continued*)  
   avoiding use of the guaranteed space attribute 157  
   coding an IEFSSNxx member 103, 104  
   configuration characteristics for MTL 228, 231  
   converting unmovable data sets 73  
   defining a minimal configuration 79, 83  
   defining COMMDSs 16  
   defining SMS classes and groups 199  
   defining the default management class 95  
   designing for database data 162  
   designing for IMS data 154  
   managing DFSMSHsm volumes 234  
   managing temporary data sets 109  
   managing TSO data sets 123  
   migrating data types 119  
   migrating database data 166  
   migrating from prior DFSMS releases 81  
   migrating permanent data 119  
   minimal configuration 87  
   planning to manage permanent DASD data 21  
   requesting specific devices 13  
   setting up a VIO storage group 44  
   simplifying recovery 106  
   simulating the tape mount management environment 190  
   testing ACS routines 113  
   training users to understand SMS 56  
   using DASD calculation services exits 260  
   using data set separation 86  
   using global resource serialization (GRS) 82  
   using logical backup and restore techniques for striped data sets 26  
   using tape mount management 4  
   using the ISMF Mountable Tape Application 229  
   using VOL=REF vs. VOL=SER 212, 213  
   using volsers 235  
 Recording Technology (data class attribute) 219  
 RECOG parameter 57  
 RECOVER command 38  
 recovery  
   considerations for a system group 106  
   support, aggregate (ABARS) 39  
 REFDD keyword 57  
 releasing unused space 136  
 replacing pattern DSCBs 137, 141  
 REPLICATE parameter 57  
 reporting, using NaviQuest 20  
 restrictions  
   removing data from system management 53  
   testing ACS routines 113  
   using tape mount management for OAM 172  
   using VOL=REF 63, 112, 216

restrictions (*continued*)  
   using VSAM data striping 27  
 RLS CF Cache Value parameter 58

## S

SAF 68  
 SAM compression 58  
 sample  
   ACS routines 243  
   classes 243  
   groups 243  
 sample project plan 237  
 SCDS  
   calculating the size 81  
   defining to z/OS 81  
   description 16  
   validating 102  
 SCDS Base Define Panel  
   Page 1 of 2 84  
   Page 2 of 2 83  
 security 67  
 SELECT/WHEN statement 46  
 selecting  
   ACS variables 50  
   volumes 42  
 sending comments to IBM xiii  
 separating large data sets 44  
 sequential data striping  
   description 25  
   DFSMS benefits for batch data 135, 137  
   planning for 25  
   tape mount management data sets 199  
   using 26  
 SET SMS operator command 106, 107  
 SET statement 46  
 SETSMS operator command 106, 107  
 SETSYS command 38  
 Shareoptions parameter 58  
 SHAREOPTIONS parameter 57  
 shortcut keys 263  
 SMS 1  
   activating 105, 106  
   aggregate group 14  
   allocating DFSORT temporary data sets 110  
   base configuration  
   default management class 95  
   description 16, 83  
   storage class 87  
   storage group 91  
   system group 87  
   classes and groups  
   defining for minimal configuration 87  
   managing temporary data sets 111  
   restricting user access using RACF 70  
   configuration 79  
   control data sets  
   ACDS 16  
   allocating 80  
   calculating sizes 81  
   COMMDS 16



- SMS (*continued*)
    - control data sets (*continued*)
      - defining to z/OS 81
      - overview 16
      - SCDS 16
    - controlling processing 106
    - data class overview 8
    - management class
      - defining default 95
      - overview 10
    - managing data 6
    - minimal configuration
      - allocating control data sets 80
      - defining 83
      - defining default management class 95
      - defining storage class 87
      - defining storage group 91
      - translating ACS routines 101
      - writing ACS routines 98
    - new configuration 105
    - operator commands 106
    - pre-ACS interface 204
    - storage class
      - defining for minimal configuration 87
      - overview 9
    - storage group
      - defining for minimal configuration 91
      - overview 12
  - SMS Volume Status Define panel 91
  - SnapShot feature, RAMAC Virtual Array 30, 146
  - software base 67
  - source control data set (SCDS) 81
  - SPACE parameter 57
  - spill storage group 215
  - split cylinder data sets 73
  - stacking, data set
    - overview 209
    - using volume reference 211
    - with volume serial 212
  - STANDARD storage class 202
  - starter set 20
  - statistics
    - cache, in SMF records 4
    - data set-level, in SMF records 24
    - DFSMS I/O 158
    - RMF Device Activity Report 137
    - space, in the COMMDS 16
    - tape usage summary 184
    - volume, in the COMMDS 16, 81
  - storage class 87, 89
    - ACS routine
      - example 249
      - managing temporary data 113
      - minimal configuration 99
      - tape mount management 206
      - translating 101
    - copying 87, 89
    - defining
      - managing temporary data 111
      - minimal configuration 87
      - tape mount management 202
    - description and summary 247
    - managing temporary data 111
  - storage class (*continued*)
    - objects 10
    - overview 9
    - specifying attributes
      - dual copy 29
      - fault-tolerant device 29
    - STANDARD 202
    - volume allocation 33
  - Storage Class Application Selection panel 88
  - Storage Class Define panel 25
  - Storage Class List panel 87, 89
  - storage group
    - ACS routine
      - example 256
      - minimal configuration 100
      - tape mount management 208
      - translating 101
    - defining
      - managing temporary data 111
      - minimal configuration 91
      - non-existent 94
      - tape mount management 203
      - volume serial numbers 94
    - description and summary 254
    - managing temporary data 111
    - overflow 199, 215
    - overview 12
    - selecting ISMF application 91
    - size-based 13
    - volume selection 12
  - Storage Group Application Selection panel 91
  - Storage Group Status Define panel 91
  - Storage Group Volume Selection panel 94
  - Storage Management Subsystem (SMS) 83
  - striping, data set
    - improving performance 25
    - using 26
  - striping, VSAM data 27
  - Summary of changes xv
  - synchronization time interval 104
  - system group
    - defining 87
    - name 86
  - system-determined block size 77
  - system-managed
    - buffering 27
    - tape 219
  - system-managed storage
    - backup 1
    - batch data 135
    - benefits 2
    - data security 1
    - database data 145
    - definition 1
    - deletion 1
    - managing data 6
    - migration 1
    - permanent data 119
    - placement 1
    - planning 19
    - recall 1
    - recovery 1
    - software base 67
  - system-managed storage (*continued*)
    - temporary data 109
    - TSO data 121
  - system-managed tape
    - planning 171
- T**
- TAPACTV data class 201
  - TAPBKUP data class 201
  - tape
    - configuration database 229
    - conversion
      - preparing 174
    - converting
      - tape volumes to system-managed 173
    - current environment
      - analyzing 182
      - optimizing 171
    - data set
      - classifying 181
      - storage class 7
    - management system 179
    - mount management
      - classifying tape data sets 181
      - data set stacking 209
      - description 177
      - excluding data sets 192
      - implementation 179
      - managing tape data 22
      - optimizing tape usage 177
      - volume reference 209
      - volume reference chains 218
      - volume reference data sets 215
      - volume serial 212
    - optimizing tape usage 179
    - system-managed 19
      - benefits 219
      - planning 171
    - system-managed benefits 4
    - tape data sets
      - system-managed 179
    - volume mount analyzer
      - analyzing 172
    - volumes
      - managing 219, 220
    - tape mount management 219, 220
      - implementation plan 172
    - Tape Storage Group Define panel 224
    - tape volumes
      - managing 219, 220
  - TAPOSITE data class 201
  - TAPTEMP data class 201
  - temporary data
    - benefits, SMS 109
    - implementing DFSMS 110
    - managing
      - activating the configuration 118
      - creating ACS test cases 113
      - defining the storage classes 111
      - defining the storage groups 111
      - running ACS test cases 116
      - testing ACS routines 113
      - translating ACS routines 113
      - validating the configuration 117
      - writing the ACS routines 113

- Test ACS Routines panel 116
- testing, using NaviQuest 20
- Tivoli Storage Manager 1, 2
- TMMACTV data class 201
- TMMACTV management class 202
- TMMBKUP data class 201
- TMMBKUP management class 202
- TMMTEMP data class 201
- top report 189
- trademarks 269
- Translate ACS Routines panel 101
- TSO data
  - benefits, SMS 121
  - implementing DFSMS
    - automatic data allocation 128
    - backup and space management 124
  - CONVERTV command, DFSMSdss 126
  - description 122
  - fallback strategy 123
  - performance and availability 124
  - physical space requirements 126
  - using data movement, DFSMSdss 127
- managing 121

## U

- unit
  - affinity 213
- unit affinity 214
- UNIT usage 63
- unmovable data sets
  - absolute track allocation 73
  - absolute track data sets 76
  - EXCP 73
  - split cylinder allocation 73
- updating PARMLIB 103
- usage report 189
- user exit 259
- user interface
  - ISPF 263
  - TSO/E 263

## V

- Validate ACS Routines or Entire SCDS panel 102
- variable
  - &ACCT\_JOB 50
  - &ACSENVIR 49
  - &ALLVOL 47
  - &ANYVOL 47
  - &APPLIC 136
  - &DATACLAS 46
  - &DSN 48
  - &DSTYPE 136
  - &EXPDT 48
  - &GROUP 152
  - &LLQ 58
  - &MAXSIZE 48
  - &MGMTCLAS 46
  - &NQUAL 48
  - &NVOL 48
  - &PGM 205

- variable (*continued*)
  - &RECORG 131
  - &RETPD 48
  - &SIZE 48
  - &SPECIAL\_USERS 152
  - &STORCLAS 46
  - &STORGRP 46
  - &SYSNAME 50
  - &SYSPLEX 50
  - &USER 109
  - &VALID\_STORAGE\_CLASS 152
- VARY SMS command 106
- VARY SMS operator command 107
- viewing the starter set 72
- VIO storage group 44
- virtual concurrent copy 30, 146
- Virtual Tape Server (VTS) 22
- VOL=REF usage 62, 211
- VOL=SER usage 62, 212
- volume
  - DASD selection 12
  - defining storage group with no real volumes 94
- volume mount analyzer 172, 182
- VSAM
  - extended addressability data set
    - allocation 58
    - restrictions 59
  - KSDS data compression, IMS restriction 148
  - partial release 35
- VSAM data striping 27
- VTS 47, 179

## W

- WRITE statement 47









Product Number: 5650-ZOS

Printed in USA

SC23-6849-00

